



**INFORMATIKA VA AXBOROT
TEXNOLOGIYALARI FANINI
O'QITISHNING INNOVATSION
METODLARI
O'QUV-USLUBIY MAJMUA**

SAMDU_MM



2026

**O‘ZBEKISTON RESPUBLIKASI
OLIV TA‘LIM, FAN VA INNOVATSIYALAR VAZIRLIGI**

**OLIV TA‘LIM TIZIMI KADRLARINI QAYTA TAYYORLASH VA
MALAKASINI OSHIRISH INSTITUTI**

**SHAROF RASHIDOV NOMIDAGI SAMARQAND DAVLAT UNIVERSITETI
HUZURIDAGI PEDAGOG KADRLARNI QAYTA TAYYORLASH VA
ULARNING MALAKASINI OSHIRISH
MINTAQAVIY MARKAZI**

**“INFORMATIKA VA AXBOROT TEXNOLOGIYALARI FANINI
O‘QITISHNING INNOVATSION METODLARI”**

moduli bo‘yicha

O‘QUV-USLUBIY MAJMUA

Samarqand–2026

Modulning ishchi dasturi Oliy ta'lim, fan va innovatsiyalar vazirligining
2025-yil 30-dekabrda oliy ta'lim muassasalari pedagog kadrlarini qayta tayyorlash
va malakasini oshirish yo'nalishlari o'quv reja va dasturlariga muvofiq ishlab
chiqilgan

Tuzuvchilar: **O.R.Yusupov** – Sharof Rashidov nomidagi Samarqand davlat
universiteti, “Dasturiy injinering” kafedrasida dotsenti, PhD, dotsent.

Taqrizchi: **S.S.Qobilov** – Sharof Rashidov nomidagi Samarqand davlat
universiteti, “Dasturiy injinering” kafedrasida dotsenti, PhD,
dotsent.

*O'quv-uslubiy majmua Sharof Rashidov nomidagi Samarqand davlat
universiteti Kengashining qarori bilan nashrga tavsiya qilingan
(2025- yil “27” noyabrda 4-sonli bayonnoma).*

KIRISH

Ushbu ishchi dastur O‘zbekiston Respublikasining 2020-yil 23 sentyabrda tasdiqlangan “Ta’lim to‘g‘risida”gi Qonuni, O‘zbekiston Respublikasi Prezidentining 2020-yil 3 dekabrda “Iqtidorli yoshlarni saralab olish tizimi va akademik litseylar faoliyatini takomillashtirish chora-tadbirlari to‘g‘risida”gi PQ-4910-son hamda Vazirlar Mahkamasining 2022-yil 1 iyundagi “Akademik litseylar rahbar va pedagog xodimlarining uzduksiz malakasini oshirish tizimini joriy etish to‘g‘risida”gi 296-son Qarorlarida belgilangan ustuvor yo‘nalishlar hamda namunaviy dastur mazmunidan kelib chiqqan holda tuzilgan bo‘lib, u zamonaviy talablar asosida malaka oshirish jarayonlarining mazmunini takomillashtirish hamda akademik litseylar pedagog xodimlarining kasbiy kompetentligini muntazam oshirib borishni maqsad qiladi.

Dastur doirasida berilayotgan tayanch modullari mavzulari orqali akademik litseylarda faoliyat olib borayotgan pedagog xodimlarning informatika va axborot texnologiyalari fanini o‘qitishning innovatsion metodlarini ta’lim jarayoniga joriy etish darajasini oshirish hisobiga ularning pedagogik mahorat va kasbiy kompetentligini muntazam takomillashtirish bilan birgalikda pedagog xodimlarning ehtiyojlari asosida tanlab olingan tanlov modullari bo‘yicha bilim, ko‘nikma va malakalarga ega bo‘lishlari ta’minlanadi.

Modulning maqsad va vazifalari

Akademik litseylar pedagog xodimlarining malakasini oshirish kursining **maqsadi** pedagog xodimlarning o‘quv-tarbiyaviy jarayonlarini yuqori ilmiy-metodik darajada ta’minlashlari uchun zarur bo‘ladigan kasbiy bilim, ko‘nikma va malakalarini muntazam yangilash, kasbiy kompetentligi va pedagogik mahoratining uzluksiz rivojlanishini ta’minlashdan iborat.

Modulning vazifalariga quyidagilar kiradi:

- pedagog kadrlarning kasbiy bilim, ko‘nikma, malakalarini uzluksiz oshirish va rivojlantirish;
- pedagoglarning zamonaviy talablarga mos holda akademik litseylardagi o‘qitish sifati va samaradorligini ta’minlash uchun zarur bo‘lgan kasbiy mahorat darajasini oshirish;
- o‘qitishning innovatsion texnologiyalari va ilg‘or xorijiy tajribalarni o‘zlashtirish hamda ulardan o‘quv jarayonida samarali foydalanish ko‘nikmalarini shakllantirish;
- o‘quv jarayonini ilm-fan va ishlab chiqarish bilan samarali integratsiyasini ta’minlashga qaratilgan faoliyatni tashkil etish.

Modul yakunida tinglovchilarning bilim, ko‘nikma va malakalari hamda kompetentligiga qo‘yiladigan talablar:

“Informatika va axborot texnologiyalari fanini o‘qitishning innovatsion metodlari” modulini o‘zlashtirish jarayonida amalga oshiriladigan masalalar doirasida:

Tinglovchi:

- onlayn va gibrid ta‘lim muhitida o‘quv jarayonini tashkil etish obektga yo‘naltirilgan va hodisalarga asoslangan dasturlash prinsiplari hamda ularni bosqichma-bosqich o‘rgatish metodlarini;
- Pandas, NumPy, Flask, Django kabi kutubxona va freymvorklardan ta‘limiy vazifalarda foydalanish imkoniyatlarini;
- raqamli pedagogika konsepsiyalari, aralash va masofaviy ta‘limni samarali tashkil etish metodlarini;
- kiberxavfsizlik asoslari, shaxsiy ma‘lumotlarni himoyalash va xavf-xatarlarning oldini olish choralarini;
- STEAM yondashuvi asosida informatika fanini boshqa fanlar bilan integratsiyalash yo‘llarini;
- 3D-modellashtirish, robototexnika, simulyatsiya hamda VR/AR texnologiyalarini *bilishi* kerak.

Tinglovchi:

- raqamli muhitdagi muloqot madaniyatini shakllantirish;
- onlayn ta‘lim jarayonida etik me‘yorlarga rioya qilishni o‘rgatish;
- sun‘iy intellekdan halol foydalanish meyorlariga rioya etish;
- raqamli ta‘lim resurslaridan samarali foydalanish va ularni tanlash; innovatsion metodlar asosida informatika darslarini loyihalash;
- raqamli pedagogika konsepsiyalari, aralash va masofaviy ta‘limni samarali tashkil etish metodlaridan samarali foydalanish;
- 3D-modellashtirish, robototexnika, simulyatsiya hamda VR/AR texnologiyalarini amaliyotga joriy etish *ko‘nikma va malakalariga* ega bo‘lishi lozim.

Tinglovchi:

- onlayn va gibrid ta‘lim muhitida o‘quv jarayonini tashkil etish obektga yo‘naltirilgan va hodisalarga asoslangan dasturlash prinsiplari hamda ularni bosqichma-bosqich o‘rgatish metodlarini amaliyotga tadbiq etish;
- ta‘lim jarayonini raqamli texnologiyalar asosida optimallashtirish.

- kiberxavfsizlik asoslari, shaxsiy ma'lumotlarni himoyalash va xavf-xatarlarning oldini olish choralari ko'rish;
- STEAM yondashuvi asosida informatika fanini boshqa fanlar bilan integratsiyalash yo'llaridan foydalanish *kompetensiyalariga* ega bo'lishi lozim.

MODUL BO'YICHA SOATLAR TAQSIMOTI

№	Modul mavzulari	Auditoriya o'quv yuklamasi		
		Jami	Nazariy	Amaliy mashg'ulot
1.	Ta'lim jarayonini raqamli texnologiyalar asosida optimallashtirish. Pandas, NumPy, Flask, Django kabi kutubxonalar va freymvorklardan ta'limiy vazifalarda foydalanish	4	2	2
2.	Kiberxavfsizlik asoslari, shaxsiy ma'lumotlarni himoyalash va xavf-xatarlarning oldini olish choralari	2	2	
3.	STEAM yondashuvi asosida informatika fanini boshqa fanlar bilan integratsiyalash yollari.	2		2
4.	3D-modellashtirish, robototexnika, simulyatsiya hamda VR/AR texnologiyalari.	2		2
	Jami:	10	4	6

NAZARIY MASHG'ULOT MAZMUNI

1-MAVZU: Ta'lim jarayonini raqamli texnologiyalar asosida optimallashtirish. Pandas, NumPy, Flask, Django kabi kutubxonalar va freymvorklardan ta'limiy vazifalarda foydalanish. (2 SOAT)

Ta'lim jarayonini raqamli texnologiyalar asosida optimallashtirishning nazariy asoslari. NumPy kutubxonasidan ta'limiy vazifalarda foydalanish imkoniyatlari. Pandas kutubxonasi orqali ta'lim jarayonini tahlil qilish va optimallashtirish. Flask freymvorki asosida yengil ta'limiy veb-ilovalar yaratish. Django freymvorki asosida kompleks ta'lim platformalarini ishlab chiqish. Kutubxonalar va freymvorklarni integratsiyalashgan holda qo'llash. Ta'limda raqamli yechimlarni joriy etish muammolari va ularni hal etish yo'llari

2-MAVZU: Kiberxavfsizlik asoslari, shaxsiy ma'lumotlarni himoyalash va xavf-xatarlarning oldini olish choralari

Kiberxavfsizlikning asosiy tushunchalari va tamoyillari. Shaxsiy ma'lumotlar va ularni himoyalash asoslari. Kiberxavflar va tahdidlar turlari. Shaxsiy va korporativ ma'lumotlarni himoyalash usullari Xavf-xatarlarning oldini olish choralari. Ta'lim muassasalari va tashkilotlarda kiberxavfsizlik. Zamonaviy texnologiyalar va kiberxavfsizlik.

AMALIY MASHG'ULOT MAZMUNI

1-amaliy mashg'ulot: Ta'lim jarayonini raqamli texnologiyalar asosida optimallashtirish. Pandas, NumPy, Flask, Django kabi kutubxona va freymvorklardan ta'limiy vazifalarda foydalanish. (2 SOAT)

Ta'lim jarayonini raqamli texnologiyalar asosida optimallashtirishning nazariy asoslari. NumPy kutubxonasidan ta'limiy vazifalarda foydalanish imkoniyatlari. Pandas kutubxonasi orqali ta'lim jarayonini tahlil qilish va optimallashtirish. Flask freymvorki asosida yengil ta'limiy veb-ilovalar yaratish. Django freymvorki asosida kompleks ta'lim platformalarini ishlab chiqish. Kutubxona va freymvorklarni integratsiyalashgan holda qo'llash. Ta'limda raqamli yechimlarni joriy etish muammolari va ularni hal etish yo'llari

2-amaliy mashg'ulot: STEAM yondashuvi asosida informatika fanini boshqa fanlar bilan integratsiyalash yo'llari. (2 soat)

STEAM yondashuvining nazariy asoslari. Informatika fanining STEAM tizimidagi ro'li. Informatika fanini tabiiy fanlar bilan integratsiyalash. Informatika fanini muhandislik va texnologiya bilan integratsiyalash. Informatika va san'at (Arts) integratsiyasi. STEAM loyihalarini loyihalash va amalga oshirish. STEAM darslarida raqamli vositalar va platformalar. STEAM yondashuvini joriy etishdagi muammolar va yechimlar

3-amaliy mashg'ulot: 3D-modellashtirish, robototexnika, simulyatsiya hamda VR/AR texnologiyalari. (2 soat)

3D-modellashtirish texnologiyalari. Robototexnika asoslari. Simulyatsiya texnologiyalari. VR (Virtual Reality) texnologiyalari. AR (Augmented Reality) texnologiyalari. Texnologiyalarni integratsiyalashgan holda qo'llash. Ta'limda joriy etish muammolari va yechimlar

Modulni tashkil etish va o‘tkazish bo‘yicha tavsiyalar

“Informatika va axborot texnologiyalari fanini o‘qitishning innovatsion metodlari” moduli ma’ruza va amaliy mashg‘ulotlar shaklida olib boriladi.

Kursni o‘qitish jarayonida ta’limning zamonaviy metodlari, pedagogik texnologiyalar va masofaviy ta’limga asoslangan raqamli texnologiyalari qo‘llanilishi nazarda tutilgan:

- video ma’ruzada zamonaviy interaktiv texnologiyalar yordamida prezentatsiya va elektron-didaktik texnologiyalar;

- o‘tkaziladigan amaliy mashg‘ulotlarda bulutli texnologiyaga asoslangan dasturiy vositalar, ekspress-so‘rovlar, test so‘rovlari va boshqa interaktiv ta’lim usullarini qo‘llash nazarda tutiladi.

ADABIYOTLAR RO‘YXATI

I. Maxsus adabiyotlar

1. UNESCO. Education in a Post-COVID World: Nine Ideas for Public Action. – Paris: UNESCO, 2020.
2. OECD. Digital Education Outlook 2021: Pushing the Frontiers with AI, Blockchain and Robots. – Paris: OECD Publishing, 2021.
3. World Economic Forum. Schools of the Future: Defining New Models of Education for the Fourth Industrial Revolution. – Geneva, 2020.
4. Redecker, C. European Framework for the Digital Competence of Educators (DigCompEdu). – Luxembourg: Publications Office of the European Union, 2020.
5. Anderson, J., Rainie, L. The Future of Digital Learning. – Pew Research Center, 2021.
6. Hodges, C., Moore, S., Lockee, B., Trust, T., Bond, A. The Difference Between Emergency Remote Teaching and Online Learning. – EDUCAUSE Review, 2020.
7. Selwyn, N. Education and Technology: Key Issues and Debates. – 3rd ed. – London: Bloomsbury Academic, 2022.
8. Holmes, W., Bialik, M., Fadel, C. Artificial Intelligence in Education: Promises and Implications for Teaching and Learning. – Boston: Center for Curriculum Redesign, 2021.
9. Kergel, D., Heidkamp, B., Tellés, P. K., Rachwal, T., Nowakowski, S. Digital Transformation in Higher Education. – Wiesbaden: Springer, 2021.
10. Alimov, R. X., Xoliqulov, A. A. Raqamli ta’lim muhitida pedagogik texnologiyalar. – Toshkent: Fan va texnologiya, 2021.
11. Karimov, U. U. Informatika ta’limida innovatsion yondashuvlar. – T.: O’qituvchi, 2022.
12. Mirzayev, M. M. Ta’limda raqamli texnologiyalar va kompetensiyaviy yondashuv. – T.: Akademnashr, 2023.
13. Bond, M., Bedenlier, S., Marín, V. I., Händel, M. Emergency Remote Teaching in Higher Education: Mapping the First Global Online Semester. – International Journal of Educational Technology in Higher Education, 2021.
14. Zhu, Z. T., Yu, M. H., Riezebos, P. A Research Framework of Smart Education. – Smart Learning Environments, 2020.
15. Qodirov, B. S. Axborot-kommunikatsiya texnologiyalari asosida informatika fanini o’qitish metodikasi. – T., 2024.

IV. Elektron ta’lim resurslari

1. <http://natlib.uz>

2. <http://yedu.uz>
3. <http://lex.uz>
4. <http://lib.bimm.uz>
5. <http://ziyonet.uz>

II. MODULNI O‘QITISHDA FOYDALANILADIGAN INTERFAOL TA’LIM METODLARI

Hozirgi kunda ta’lim jarayonida sun’iy intellekt texnologiyalariga asoslangan zamonaviy o‘qitish metodlari tobora keng qo‘llanilmoqda. Ushbu metodlardan foydalanish ta’lim jarayonining samaradorligini oshirish, o‘quvchilarning bilimlarni chuqur va ongli o‘zlashtirishiga erishish imkonini beradi. Sun’iy intellektga asoslangan ta’lim metodlari o‘quvchilarning individual imkoniyatlari va ehtiyojlarini hisobga olgan holda ta’limni tashkil etish, shaxsiylashtirilgan va adaptiv o‘qitishni amalga oshirishga xizmat qiladi. Shu bois, ta’lim metodlarini tanlashda har bir mashg‘ulotning maqsadi va didaktik vazifalaridan kelib chiqish muhim ahamiyat kasb etadi.

An’anaviy dars shaklini saqlab qolgan holda, uni sun’iy intellekt texnologiyalariga asoslangan interfaol metodlar bilan boyitish ta’lim oluvchilarning faolligini oshiradi va ularning o‘zlashtirish darajasining sezilarli darajada yaxshilanishiga olib keladi. Bunda o‘quv mashg‘ulotlarini oqilona tashkil etish, ta’lim beruvchining sun’iy intellekt vositalari yordamida ta’lim oluvchilarning qiziqishlari, bilim darajasi va o‘rganilayotgan o‘quv materialining mazmuniga mos metod hamda vositalarni tanlashi yuqori samaradorlikni ta’minlaydi. Shuningdek, sun’iy intellektga asoslangan interfaol va interaktiv ta’lim metodlaridan foydalanish orqali ta’lim oluvchilarning mustaqil fikrlash, tahlil qilish, amaliy ko‘nikma va malakalarini rivojlantirish, hamda raqamli muhitda samarali faoliyat yuritish kompetensiyalarini shakllantirish mumkin bo‘ladi.

Interfaol metodlar deganda ta’lim oluvchilarning faolligini oshirish, ularni mustaqil va tanqidiy fikrlashga yo‘naltirish hamda ta’lim-tarbiya jarayonida ta’lim oluvchi–ta’lim oluvchi va ta’lim oluvchi–o‘qituvchi hamkorligi orqali yuqori samaradorlikka erishishni ta’minlaydigan o‘qitish usullari tushuniladi. Ushbu metodlardan foydalanilganda o‘qituvchi ta’lim jarayonini tashkil etuvchi va yo‘naltiruvchi sifatida ishtirok etib, ta’lim oluvchilarni faol ishtirok etishga undaydi. Natijada ta’lim oluvchilar o‘quv jarayonining barcha bosqichlarida faol qatnashadilar.

Interfaol ta’lim metodlari asosida tashkil etilgan mashg‘ulotlar bir qator ijobiy jihatlari bilan ajralib turadi. Jumladan, bunday mashg‘ulotlar o‘quv materialini chuqur va samarali o‘zlashtirishga xizmat qiladi hamda ta’lim oluvchilarning o‘qishga bo‘lgan qiziqishini va ichki rag‘batini oshiradi. Shu bilan birga, ilgari o‘zlashtirilgan bilimlar yangi bilimlarni egallash jarayonida hisobga olinadi va ular bilan uzviy bog‘lanadi. Interfaol metodlar ta’lim oluvchilarning tashabbuskorligi va mas’uliyatini qo‘llab-quvvatlab, bilimlarni amaliy faoliyat orqali mustahkamlash imkonini beradi. Shuningdek, o‘quv jarayonida ikki tomonlama fikr almashish, muhokama va tahlil qilish uchun qulay sharoit yaratiladi.

Quyida o‘quv mashg‘ulotlarida keng qo‘llaniladigan interfaol ta‘lim metodlarining mazmuni hamda ularni qo‘llash bosqichlari bilan batafsil tanishamiz.



“Aqliy hujum” metodi - biror muammo bo‘yicha ta‘lim oluvchilar tomonidan bildirilgan erkin fikr va mulohazalarni to‘plab, ular orqali ma‘lum bir yechimga kelinadigan metoddir. “Aqliy hujum” metodining yozma va og‘zaki shakllari mavjud. Og‘zaki shaklida ta‘lim beruvchi tomonidan berilgan savolga ta‘lim oluvchilarning har biri o‘z fikrini

og‘zaki bildiradi. Ta‘lim oluvchilar o‘z javoblarini aniq va qisqa tarzda bayon etadilar. Yozma shaklida esa berilgan savolga ta‘lim oluvchilar o‘z javoblarini qog‘oz kartochkalarga qisqa va barchaga ko‘rinarli tarzda yozadilar. Javoblar doskaga (magnitlar yordamida) yoki «pinbord» doskasiga (ignalar yordamida) mahkamlanadi. “Aqliy hujum” metodining yozma shaklida javoblarni ma‘lum belgilar bo‘yicha guruhlab chiqish imkoniyati mavjuddir. Ushbu metod to‘g‘ri va ijobiy qo‘llanilganda shaxsni erkin, ijodiy va nostandart fikrlashga o‘rgatadi.

“Aqliy hujum” metodidan foydalanilganda ta‘lim oluvchilarning barchasini jalb etish imkoniyati bo‘ladi, shu jumladan ta‘lim oluvchilarda muloqot qilish va munozara olib borish madaniyati shakllanadi. Ta‘lim oluvchilar o‘z fikrini faqat og‘zaki emas, balki yozma ravishda bayon etish mahorati, mantiqiy va tizimli fikr yuritish ko‘nikmasi rivojlanadi. Bildirilgan fikrlar baholanmasligi ta‘lim oluvchilarda turli g‘oyalar shakllanishiga olib keladi. Bu metod ta‘lim oluvchilarda ijodiy tafakkurni rivojlantirish uchun xizmat qiladi.

“Muammoli vaziyat” metodi - ta‘lim oluvchilarda muammoli vaziyatlarning sabab va oqibatlarini tahlil qilish hamda ularning yechimini topish bo‘yicha ko‘nikmalarini shakllantirishga qaratilgan metoddir.

“Muammoli vaziyat” metodi uchun tanlangan muammoning murakkabligi ta‘lim oluvchilarning bilim darajalariga mos kelishi kerak. Ular qo‘yilgan muammoning yechimini topishga qodir bo‘lishlari kerak, aks holda yechimni topa olmagach, ta‘lim oluvchilarning qiziqishlari so‘nishiga, o‘zlariga bo‘lgan ishonchlarining yo‘qolishiga olib keladi. «Muammoli vaziyat» metodi qo‘llanilganda ta‘lim oluvchilar mustaqil fikr yuritishni, muammoning sabab va oqibatlarini tahlil qilishni, uning yechimini topishni o‘rganadilar.

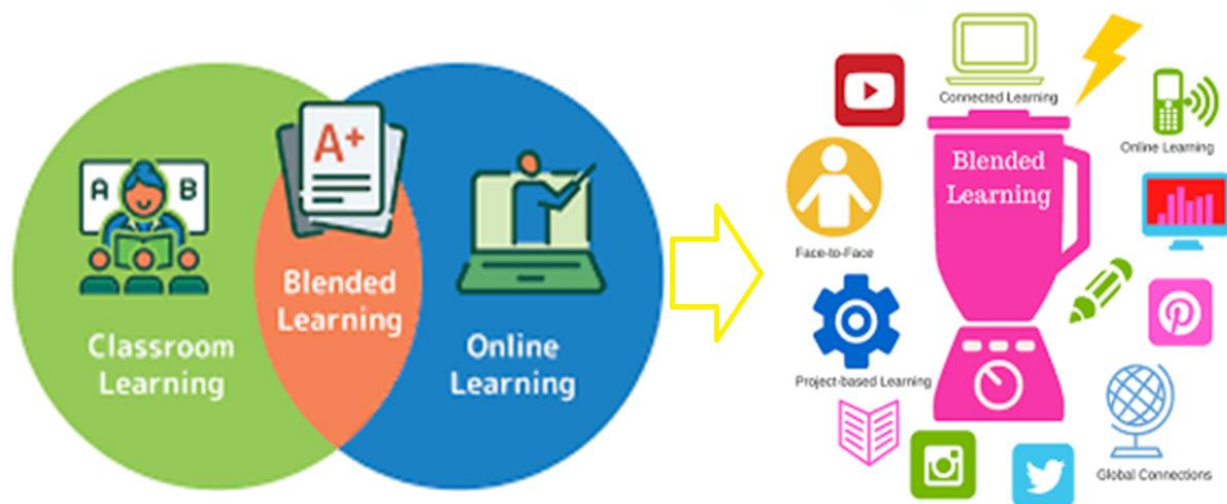
“Muammoli vaziyat” metodining afzalliklari:

- ta‘lim oluvchilarda mustaqil fikrlash qobiliyatlarini shakllantiradi;
- ta‘lim oluvchilar muammoning sabab, oqibat va yechimlarni topishni o‘rganadilar;

- ta'lim oluvchilarning bilim va qobiliyatlarini baholash uchun yaxshi imkoniyat yaratiladi;
- ta'lim oluvchilar fikr va natijalarni tahlil qilishni o'rganadilar.

Blended Learning metodi

Blended Learning



Flipped Learning texnologiyasi

Ўқитишнинг анъанавий шакли

Машғулот давомида



Назарий материални ўзлаштириш



Берилган вазифани бажариш

Машғулотдан ташқари вақтда



Берилган вазифани бажариш

Flipped learning shakli

Машғулотдан ташқари вақтда



Назарий материални ўзлаштириш



Берилган вазифани бажариш

Машғулот давомида



Билимни такомиллаштириш



Берилган вазифани бажариш

III.NAZARIY MATERIALLAR

1-MAVZU: Ta'lim jarayonini raqamli texnologiyalar asosida optimallashtirish. Pandas, NumPy, Flask, Django kabi kutubxona va freymvorklardan ta'limiy vazifalarda foydalanish. (2 SOAT)

Ta'lim jarayonini raqamli texnologiyalar asosida optimallashtirishning nazariy asoslari. NumPy kutubxonasi ta'limiy vazifalarda foydalanish imkoniyatlari. Pandas kutubxonasi orqali ta'lim jarayonini tahlil qilish va optimallashtirish. Flask freymvorki asosida yengil ta'limiy veb-ilovalar yaratish. Django freymvorki asosida kompleks ta'lim platformalarini ishlab chiqish. Kutubxona va freymvorklarni integratsiyalashgan holda qo'llash. Ta'limda raqamli yechimlarni joriy etish muammolari va ularni hal etish yo'llari.

Zamonaviy jamiyatda raqamli texnologiyalarning jadal rivojlanishi ta'lim tizimiga ham bevosita ta'sir ko'rsatmoqda. Axborot-kommunikatsiya texnologiyalari ta'lim jarayonini tashkil etish, boshqarish va baholash mexanizmlarini tubdan o'zgartirib, uning samaradorligini oshirish imkonini bermoqda. Shu jihatdan, ta'lim jarayonini raqamli texnologiyalar asosida optimallashtirish masalasi pedagogika, didaktika va ta'limni boshqarish nazariyalarining muhim yo'nalishlaridan biri hisoblanadi.

Ta'lim jarayonini optimallashtirish tushunchasi pedagogik jarayonning maqsad, mazmun, metod, vosita va natijalarini eng maqbul holatga keltirishni anglatadi. Raqamli texnologiyalar esa ushbu jarayonni tezkor, moslashuvchan va shaxsga yo'naltirilgan tarzda amalga oshirishga xizmat qiladi. Nazariy jihatdan optimallashtirish g'oyasi tizimli yondashuvga asoslanadi, ya'ni ta'lim jarayoni o'zaro bog'liq elementlardan tashkil topgan yaxlit tizim sifatida qaraladi. Raqamli texnologiyalar ana shu elementlar o'rtasidagi bog'liqlikni mustahkamlab, axborot oqimini samarali boshqarishga imkon beradi.

Ta'limni raqamlashtirishning nazariy asoslaridan biri konstruktivizm ta'lim nazariyasidir. Ushbu yondashuvga ko'ra, bilim tayyor holda berilmaydi, balki o'quvchi tomonidan faol faoliyat jarayonida shakllantiriladi. Raqamli ta'lim platformalari, interaktiv resurslar, virtual laboratoriyalar va onlayn muhitlar o'quvchining mustaqil izlanishini qo'llab-quvvatlab, bilimni ongli ravishda o'zlashtirishga yordam beradi. Bu esa ta'lim jarayonining sifatini oshirishda muhim nazariy asos bo'lib xizmat qiladi.

Shuningdek, ta'lim jarayonini raqamli texnologiyalar asosida optimallashtirishda kompetensiyaviy yondashuv muhim ahamiyatga ega. Ushbu yondashuv ta'lim natijasini faqat bilim bilan emas, balki amaliy ko'nikma, malaka va shaxsiy sifatlar bilan baholashni nazarda tutadi. Raqamli vositalar orqali o'quvchilarning faoliyati,

ishtiroki, topshiriqlarni bajarish darajasi va rivojlanish dinamikasini doimiy monitoring qilish mumkin. Bu esa kompetensiyalarni shakllantirish jarayonini aniq ma'lumotlar asosida boshqarishga imkon beradi.

Raqamli texnologiyalar asosida optimallashtirishning yana bir nazariy asosi individual va differensial ta'lim konsepsiyasidir. Har bir o'quvchining bilim darajasi, qobiliyati va o'rganish sur'ati turlicha bo'lgani sababli, an'anaviy yondashuvlar ko'pincha barcha uchun bir xil natija bera olmaydi. Raqamli ta'lim muhitlari esa adaptiv o'qitish imkoniyatlarini yaratib, o'quvchilarga individual ta'lim trayektoriyalarini shakllantirishga xizmat qiladi. Natijada ta'lim jarayoni samaraliroq va adolatiroq bo'ladi.

Bundan tashqari, ta'lim jarayonini raqamli texnologiyalar asosida optimallashtirish pedagogik jarayonda tahlil va baholashning ilmiy asoslarini mustahkamlaydi. Ta'limiy ma'lumotlarni yig'ish, saqlash va tahlil qilish imkoniyati o'qituvchiga o'z faoliyatini refleksiya qilish, metodlarni takomillashtirish va ta'lim sifatini oshirishga yordam beradi. Bu jarayon ta'limni boshqarish nazariyasi bilan uzviy bog'liq bo'lib, dalillarga asoslangan qarorlar qabul qilish imkonini yaratadi.

Xulosa qilib aytganda, ta'lim jarayonini raqamli texnologiyalar asosida optimallashtirishning nazariy asoslari tizimli, konstruktivistik, kompetensiyaviy hamda shaxsga yo'naltirilgan yondashuvlarga tayanadi. Ushbu yondashuvlar raqamli vositalar bilan uyg'unlashgan holda ta'lim samaradorligini oshirish, ta'lim sifati va natijalarini barqaror rivojlantirish uchun mustahkam ilmiy asos bo'lib xizmat qiladi.

Zamonaviy ta'lim tizimida raqamli texnologiyalar va dasturlash vositalarining o'rni tobora ortib bormoqda. Ayniqsa, ma'lumotlar bilan ishlash, matematik modellashtirish, statistik tahlil va sun'iy intellekt elementlarini o'qitishda Python dasturlash tili yetakchi o'rinlardan birini egallaydi. Pythonning eng muhim va asosiy kutubxonalaridan biri bo'lgan **NumPy (Numerical Python)** esa sonli hisoblashlar uchun mustahkam nazariy va amaliy asos yaratadi. NumPy kutubxonasidan ta'limiy vazifalarda foydalanish o'quvchilarning mantiqiy fikrlashini, algoritmik tafakkurini va amaliy ko'nikmalarini rivojlantirishda muhim ahamiyatga ega.

NumPy kutubxonasi haqida umumiy tushuncha

NumPy — bu Python tilida yozilgan, ko'p o'lchovli massivlar va matematik amallar bilan samarali ishlashga mo'ljallangan kutubxona hisoblanadi. U ilmiy hisoblashlar uchun mo'ljallangan bo'lib, tezkorligi va qulay sintaksisi bilan ajralib turadi. NumPy'ning asosiy obyekti — **ndarray** (n-o'lchovli massiv) bo'lib, u katta hajmdagi sonli ma'lumotlarni tez va samarali qayta ishlash imkonini beradi.

Ta'lim jarayonida NumPy'dan foydalanishning nazariy asosi shundan iboratki, u matematik tushunchalarni abstrakt shakldan amaliy va ko'rgazmali shaklga o'tkazishga yordam beradi. Masalan, vektor, matritsa, funksiya qiymatlari, statistik

ko'rsatkichlar kabi mavzularni faqat formulalar orqali emas, balki real hisoblashlar orqali tushuntirish mumkin.

NumPy'ning ta'limdagi ahamiyati

NumPy kutubxonasi quyidagi fan va yo'nalishlarda ta'limiy ahamiyatga ega:

1. matematika va chiziqli algebra;
2. ehtimollar nazariyasi va statistika;
3. sonli usullar;
4. fizik va muhandislik hisoblari;
5. ma'lumotlar tahlili (Data Analysis);
6. mashinali o'qitish va sun'iy intellekt asoslari.

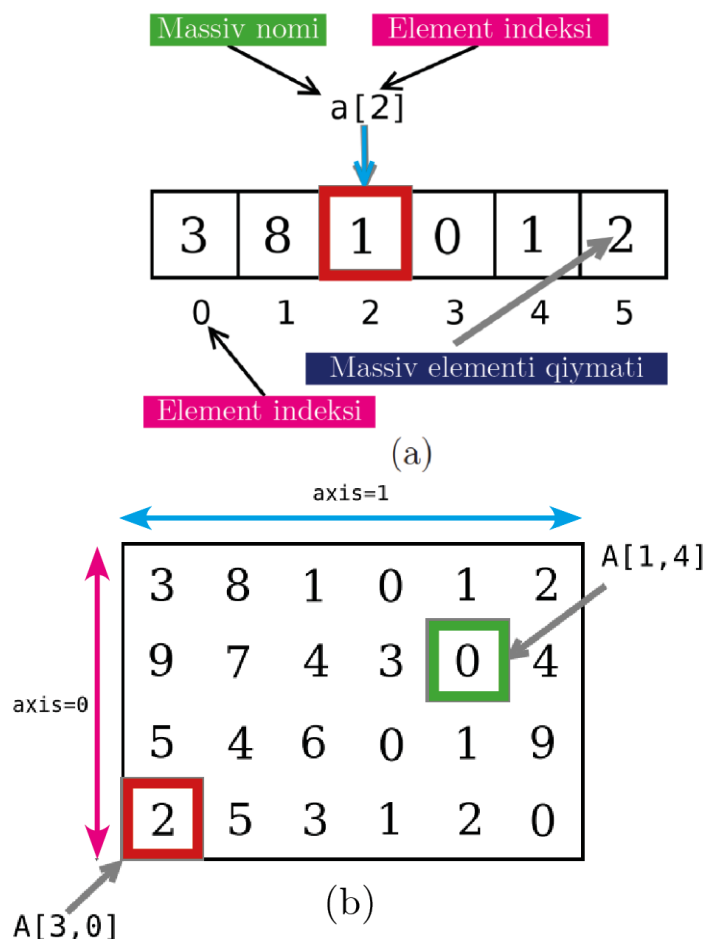
NumPy yordamida talabalar murakkab hisob-kitoblarni qo'lda bajarish o'rniga, algoritmik yondashuv asosida yechishga o'rganadilar. Bu esa nazariy bilimlarning mustahkamlanishiga xizmat qiladi.

NumPy'da massivlar yaratish va ular ustida amallar bajarish juda qulay. Masalan, oddiy ro'yxatlar bilan solishtirganda NumPy massivlari tezroq ishlaydi va kamroq xotira talab qiladi.

Massiv — xotirada ketma-ket joylashgan bir xil turdagi qiymatlarni saqlovchi tuzilmadir. Odatda, massivlar katta hajmdagi, ammo cheklangan miqdordagi va tartiblangan qiymatlarni qayta ishlash zarur bo'lganda ishlatiladi. Masalan, talabalar guruhining reyting ballari bilan ishlash zarurati paydo bo'lganda, o'rtacha reytingni aniqlash, reytinglarni kamayish bo'yicha tartiblash va muayyan talabalar reytinglarini qayta ishlash kabi masalalarni yechish uchun massivlar qo'llaniladi.

Massivning asosiy xususiyati shundaki, har bir qiymat o'zining o'rniga ega bo'ladi. Ya'ni, massivdagi har bir elementga indeks orqali murojaat qilinadi (birinchi talabaning reytingi massivda birinchi o'rinda, ikkinchi talabaniki ikkinchi o'rinda va hokazo).

Massivning har bir elementiga indeksatsiya orqali kirish amalga oshiriladi, ya'ni kerakli elementni topish uchun massivga indeks ko'rsatiladi. Buning sababi shundaki, barcha qiymatlar bir xil turga ega bo'lib, bir xil miqdordagi xotira baytlarini egallaydi; shuning uchun massivning manzili va element raqamini bilib, uning xotirada qayerda joylashganligini hisoblash mumkin. Massivning indekslari soni har xil bo'lishi mumkin, ya'ni bitta indeksli massivlar bir o'lchamli, ikkita indeksli massivlar esa ikki o'lchamli deb ataladi va hokazo. Bir o'lchamli massiv ("ustun", "qator") matematika fanidagi vektorga mos keladi (4.1(a) rasmda $a[4] = 56$, ya'ni a massivining to'rtinchi elementi 56 ga teng); ikki o'lchamli massiv esa matritsaga mos keladi (4.1(b) rasmda $A[1][6] = 22$ deb yozish mumkin, yoki $A[1, 6] = 22$). Ko'pincha bir yoki ikki indeksli massivlardan foydalaniladi; uch indeksli massivlar kamroq qo'llaniladi; undan ko'p indeksli massivlar esa juda kam uchraydi.



4.1-rasm. Rasm 5.1. Bir o'lchovli – (a) va ikki o'lchovli – (b) massivlar.

Odatda, dasturlashda massivlar statik va dinamik turlarga bo'linadi. Statik massiv – bu dastur kompilyatsiya qilingan paytda o'lchami aniqlanadigan massivdir. Dinamik tipizatsiyaga ega tillarda, masalan, Python tilida ular qo'llanilmaydi. Dinamik massiv – bu dastur ishlayotgan paytda o'lchami belgilangan massivdir. Ya'ni, dastur ishga tushirilganda bu massiv kompyuter xotirasida umuman joy egallamaydi (murojaatni saqlash uchun zarur bo'lgan xotiradan tashqari). Dinamik massivlar dastur ishlash jarayonida o'lchamini o'zgartirishni qo'llab-quvvatlashi yoki qo'llab-quvvatlamasligi mumkin. Python tilidagi massivlar o'lchamini aniq o'zgartirishni qo'llab-quvvatlamaydi: ular ro'yxatlardan farqli o'laroq, elementlarni qo'shish uchun **append** va **extend** metodlari, hamda ularni o'chirish uchun **pop** va **remove** metodlariga ega emas. Agar massivning o'lchamini o'zgartirish kerak bo'lsa, buni massivni ifodalovchi o'zgaruvchi nomiga yangi qiymatni, ya'ni eski qiymatga nisbatan katta yoki kichikroq bo'lgan yangi xotira maydoniga mos keladigan qiymatni qayta tayinlash orqali amalga oshirish mumkin.

Massiv yaratishning asosiy operatori **array** deb ataladi. Uning yordamida noldan yangi massiv yaratish yoki mavjud ro'yxatni massivga aylantirish mumkin.

Python tilini matematik paketga aylantirish uchun asosiy modullar numpy va matplotlib hisoblanadi.

numpy — bu Python tilidagi modul (aslida, modullar to‘plami) bo‘lib, u katta ko‘p o‘lchamli massivlar va matritsalar bilan turli amallarni bajarish uchun mo‘ljallangan. Shuningdek, bu modul massivlar bilan amallar bajarish uchun yuqori darajadagi (va juda tezkor) matematik funksiyalar to‘plamini taqdim etadi.

matplotlib — bu Python dasturlash tilida ikki o‘lchovli (2D) grafikani vizuallashtirish uchun modul (aslida, modullar to‘plami)dir. **matplotlib** 3D grafika bilan ham ishlay oladi.

NumPy modulidan foydalanmasdan Python tilida massivlarni e‘lon qilish uchun standart ro‘yxatlar (*lists*) dan foydalanish mumkin. Quyidagi misollar yordamida massivlarni e‘lon qilishni ko‘rib chiqamiz:

1. Bir o‘lchamli massiv (vektor) yaratish:

```
massiv = [1, 2, 3, 4, 5]
# Elementga murojaat qilish
print(massiv[2]) # Natija: 3
# Element qo'shish
massiv.append(6)
print(massiv) # Natija: [1, 2, 3, 4, 5, 6]
# Elementni olib tashlash
massiv.remove(3)
print(massiv) # Natija: [1, 2, 4, 5, 6]
```

2. Ikki o‘lchamli massiv (matritsa) yaratish

```
matritsa = [
    [1, 2, 3],
    [4, 5, 6],
    [7, 8, 9]
]
# Elementga murojaat qilish
print(matritsa[1][2]) # Natija: 6
# Qator qo'shish
matritsa.append([10, 11, 12])
print(matritsa) # Natija: [[1, 2, 3], [4, 5, 6], [7, 8, 9], [10, 11, 12]]
# Elementni o'zgartirish
matritsa[0][0] = 0
print(matritsa) # Natija: [[0, 2, 3], [4, 5, 6], [7, 8, 9], [10, 11, 12]]
```

3. Uch o‘lchamli massiv yaratish

```
massiv_3d = [
    [ [1, 2],
      [3, 4] ],
    [ [5, 6],
```

```

    [7, 8] ]
]
# Elementga murojaat qilish
print(massiv_3d[1][0][1]) # Natija: 6
# Element qo'shish
massiv_3d[0].append([9, 10])
print(massiv_3d) # Natija: [[[1, 2], [3, 4], [9, 10]], [[5, 6], [7, 8]]]

```

Python tilida ro'yxatlarning o'lchamni o'zgartirish juda oson, chunki ular dinamik tuzilmalar hisoblanadi. Misol uchun:

```

# Bir o'lchamli massivni kattalashtirish
massiv = [1, 2, 3]
massiv = massiv + [4, 5, 6]
print(massiv) # Natija: [1, 2, 3, 4, 5, 6]
# Ikki o'lchamli massivni kattalashtirish
matritsa = [
    [1, 2],
    [3, 4] ]
matritsa = matritsa + [[5, 6], [7, 8]]

```

```

print(matritsa) # Natija: [[1, 2], [3, 4], [5, 6], [7, 8]]

```

NumPy moduli yordamida Python tilida massivlarni yaratish va ulardan foydalanish juda qulay hisoblanadi. NumPy moduli Python standart kutubxonasiga kirmaydi, shuning uchun uni o'rnatish zarur bo'ladi. Ma'lumki, o'natilgan kutubxonalar(yoki modullar)dan foydalanish uchun ularni import qilish talab etiladi. Quyida Numpy yordamida massivlarni e'lon qilishni ko'rib o'tamiz:

1. Bir o'lchamli massiv (vektor) yaratish.

```

import numpy as np # numpy kutubxonasini import qilish
# Bir o'lchamli massiv yaratish
massiv = np.array([1, 2, 3, 4, 5])
# Elementga murojaat qilish
print(massiv[2]) # Natija: 3
# Massivning o'lchamini ko'rish
print(massiv.shape) # Natija: (5,)

```

2. Ikki o'lchamli massiv (matritsa) yaratish

```

# Ikki o'lchamli massiv yaratish
matritsa = np.array([
    [1, 2, 3],
    [4, 5, 6],
    [7, 8, 9] ])

```

```

# Elementga murojaat qilish
print(matritsa[1, 2]) # Natija: 6
# Massivning o'lchamini ko'rish
print(matritsa.shape) # Natija: (3, 3)
3. Uch o'lchamli massiv yaratish
import numpy as np
# Uch o'lchamli massiv yaratish
massiv_3d = np.array([
    [ [1, 2],
      [3, 4] ],
    [ [5, 6],
      [7, 8] ] ])
# Elementga murojaat qilish
print(massiv_3d[1, 0, 1]) # Natija: 6
# Massivning o'lchamini ko'rish
print(massiv_3d.shape) # Natija: (2, 2, 2)

```

NumPy moduli yordamida massiv yaratishning boshqa qulay usullari ham mavjud bo'lib, bu usullar massivlarni tezda yaratish imkonini beradi:

1. Zeros - barcha elementlari nolga teng massiv yaratish

```

import numpy as np
# Barcha elementlari nolga teng bo'lgan bir o'lchamli massiv
zeros_massiv = np.zeros(5)
print(zeros_massiv) # Natija: [0. 0. 0. 0. 0.]
# Barcha elementlari nolga teng bo'lgan ikki o'lchamli massiv
zeros_matritsa = np.zeros((3, 4))
print(zeros_matritsa)
# Natija:
# [[0. 0. 0. 0.]
#  [0. 0. 0. 0.]
#  [0. 0. 0. 0.]]

```

2. Ones - barcha elementlari birga teng massiv yaratish

```

import numpy as np
# Barcha elementlari birga teng bo'lgan bir o'lchamli massiv
ones_massiv = np.ones(5)
print(ones_massiv) # Natija: [1. 1. 1. 1. 1.]
# Barcha elementlari birga teng bo'lgan ikki o'lchamli massiv
ones_matritsa = np.ones((3, 4))
print(ones_matritsa)

```

```
# Natija:
```

```
# [[1. 1. 1. 1.]
```

```
# [1. 1. 1. 1.]
```

```
# [1. 1. 1. 1.]]
```

3. Arange - ketma-ket raqamlar massivini yaratish

```
# 0 dan 9 gacha bo'lgan ketma-ket raqamlar massivi
```

```
arange_massiv = np.arange(10)
```

```
print(arange_massiv) # Natija: [0 1 2 3 4 5 6 7 8 9]
```

```
# 0 dan 9 gacha bo'lgan ketma-ket raqamlar massivi, 2 qadam bilan
```

```
arange_qadam = np.arange(0, 10, 2)
```

```
print(arange_qadam) # Natija: [0 2 4 6 8]
```

4. Linspace - berilgan intervaldagi teng masofadagi raqamlar massivini yaratish

```
# 0 dan 1 gacha bo'lgan 5 ta teng masofadagi raqamlar massivi
```

```
linspace_massiv = np.linspace(0, 1, 5)
```

```
print(linspace_massiv) # Natija: [0. 0.25 0.5 0.75 1. ]
```

`arange` va `linspace` funksiyalaridan foydalanib ko'p o'lchamli massivlarni yaratish uchun massivni kerakli o'lchamda qayta shakllantirish (reshape) funksiyasidan foydalanish mumkin.

Arange va reshape yordamida ko'p o'lchamli massiv yaratish quyidagicha amalga oshirish mumkin.

```
import numpy as np
```

```
#Ikki o'lchamli massiv yaratish
```

```
#0 dan 11 gacha bo'lgan ketma-ket raqamlar massivi yaratish
```

```
arange_massiv = np.arange(12)
```

```
#Massivni 3x4 o'lchamli matritsaga aylantirish
```

```
ikkinchi_massiv = arange_massiv.reshape(3, 4)
```

```
print(ikkinchi_massiv)
```

```
# Natija:
```

```
# [[ 0  1  2  3]
```

```
# [ 4  5  6  7]
```

```
# [ 8  9 10 11]]
```

```
#Uch o'lchamli massiv yaratish
```

```
# 0 dan 23 gacha bo'lgan ketma-ket raqamlar massivi yaratish
```

```
arange_massiv = np.arange(24)
```

```
# Massivni 2x3x4 o'lchamli uch o'lchamli massivga aylantirish
```

```
uchinchi_massiv = arange_massiv.reshape(2, 3, 4)
```

```
print(uchinchi_massiv)
```

```
# Natija:
```

```
# [[[ 0  1  2  3]
```

```
# [ 4 5 6 7]
# [ 8 9 10 11]]
#
# [[12 13 14 15]
# [16 17 18 19]
# [20 21 22 23]]]
```

Linspace va reshape yordamida ham ko'p o'lchamli massiv yaratish imkoni mavjud bo'lib, ushbu amallar quyidagicha bajariladi:

```
#Ikki o'lchamli massiv yaratish
# 0 dan 1 gacha bo'lgan 12 ta teng masofadagi raqamlar massivi yaratish
linspace_massiv = np.linspace(0, 1, 12)
# Massivni 3x4 o'lchamli matritsaga aylantirish
ikkinchi_massiv = linspace_massiv.reshape(3, 4)
print(ikkinchi_massiv)
# Natija:
# [[0.      0.09090909 0.18181818 0.27272727]
# [0.36363636 0.45454545 0.54545455 0.63636364]
# [0.72727273 0.81818182 0.90909091 1.      ]]
```

```
#Uch o'lchamli massiv yaratish
# 0 dan 1 gacha bo'lgan 24 ta teng masofadagi raqamlar massivi yaratish
linspace_massiv = np.linspace(0, 1, 24)
# Massivni 2x3x4 o'lchamli uch o'lchamli massivga aylantirish
uchinchi_massiv = linspace_massiv.reshape(2, 3, 4)
print(uchinchi_massiv)
# Natija:
# [[[0.      0.04347826 0.08695652 0.13043478]
# [0.17391304 0.2173913  0.26086957 0.30434783]
# [0.34782609 0.39130435 0.43478261 0.47826087]]
#
# [[0.52173913 0.56521739 0.60869565 0.65217391]
# [0.69565217 0.73913043 0.7826087  0.82608696]
# [0.86956522 0.91304348 0.95652174 1.      ]]]]
```

NumPy modulida tasodifiy sonlar yordamida massiv yaratish uchun `numpy.random` submodulidan foydalanish mumkin.

```
#Bir o'lchamli massiv
# 10 ta tasodifiy sonlardan iborat bir o'lchamli massiv yaratish
bir_olchamli = np.random.rand(10)
print(bir_olchamli)
```

```
# Natija:
# [0.5488135 0.71518937 0.60276338 0.54488318 0.4236548 0.64589411
# 0.43758721 0.891773 0.96366276 0.38344152]
```

```
#Ikki o'lchamli massiv
# 3x4 o'lchamli tasodifiy sonlardan iborat ikki o'lchamli massiv yaratish
ikki_olchamli = np.random.rand(3, 4)
```

```
print(ikki_olchamli)
```

```
# Natija:
# [[0.56804456 0.92559664 0.07103606 0.0871293 ]
# [0.0202184 0.83261985 0.77815675 0.87001215]
# [0.97861834 0.79915856 0.46147936 0.78052918]]
```

```
#Uch o'lchamli massiv
```

```
# 2x3x4 o'lchamli tasodifiy sonlardan iborat uch o'lchamli massiv yaratish
uch_olchamli = np.random.rand(2, 3, 4)
```

```
print(uch_olchamli)
```

```
# Natija:
# [[[0.11827443 0.63992102 0.14335329 0.94466892]
# [0.52184832 0.41466194 0.26455561 0.77423369]
# [0.45615033 0.56843395 0.0187898 0.6176355 ]]
#
# [[0.61209572 0.616934 0.94374808 0.6818203 ]
# [0.3595079 0.43703195 0.6976312 0.06022547]
# [0.66676672 0.67063787 0.21038256 0.1289263 ]]]
```

Agar butun sonlardan iborat tasodifiy massiv yaratish kerak bo'lsa, ``numpy.random.randint`` funksiyasidan foydalanish mumkin:

```
# Bir o'lchamli massiv
```

```
# 0 dan 10 gacha bo'lgan 10 ta tasodifiy butun sonlardan iborat bir o'lchamli
massiv yaratish
```

```
bir_olchamli = np.random.randint(0, 10, size=10)
```

```
print(bir_olchamli)
```

```
# Natija: [5 0 3 3 7 9 3 5 2 4]
```

```
#Ikki o'lchamli massiv
```

```
# 0 dan 10 gacha bo'lgan 3x4 o'lchamli tasodifiy butun sonlardan iborat ikki
o'lchamli massiv yaratish
```

```
ikki_olchamli = np.random.randint(0, 10, size=(3, 4))
```

```
print(ikki_olchamli)
```

```

# Natija:
# [[3 7 9 3]
# [7 2 0 0]
# [4 5 5 6]]
#Uch o'lchamli massiv
# 0 dan 10 gacha bo'lgan 2x3x4 o'lchamli tasodifiy butun sonlardan iborat uch
o'lchamli massiv yaratish
uch_olchamli = np.random.randint(0, 10, size=(2, 3, 4))
print(uch_olchamli)
# Natija:
# [[[9 3 5 7]
# [3 2 7 2]
# [0 9 9 1]]
#
# [[0 6 1 0]
# [1 2 5 6]
# [7 7 8 1]]]

```

NumPy modulida massiv elementlarining turini yaratish vaqtida `dtype` nomli argument yordamida belgilash mumkin. NumPy quyidagi ma'lumot turlaridan birini tanlash imkoniyatini beradi: `bool` (mantiqiy qiymat), `character` (belgi), `int8`, `int16`, `int32`, `int64` (ishorali butun sonlar, o'lchamlari mos ravishda 8, 16, 32 va 64 bit), `uint8`, `uint16`, `uint32`, `uint64` (ishorasiz butun sonlar, o'lchamlari mos ravishda 8, 16, 32 va 64 bit), `float32` va `float64` (haqiqiy sonlar, bir va ikki barobar aniqlikda), `complex64` va `complex128` (kompleks sonlar, bir va ikki barobar aniqlikda), shuningdek, o'ziga xos ma'lumot turlarini, jumladan murakkab turlarni aniqlash imkoniyatini beradi.

```

#Mantiqiy tipli (bool) massiv
# Mantiqiy (bool) qiymatlardan iborat massiv yaratish
bool_massiv = np.array([True, False, True], dtype=bool)
print(bool_massiv)
# Natija: [ True False  True]

```

```

#Butun (int) tipli massiv
# 8 bitli ishorali butun sonlardan iborat massiv yaratish
int_massiv = np.array([1, 2, 3, 4], dtype=np.int8)
print(int_massiv)
# Natija: [1 2 3 4]

```

```
#Ishorasiz butun (uint) tipli massiv
# 16 bitli ishorasiz butun sonlardan iborat massiv yaratish
uint_massiv = np.array([1, 2, 3, 4], dtype=np.uint16)
print(uint_massiv)
# Natija: [1 2 3 4]
```

```
#Haqiqiy (float) tipli massiv
# 32 bitli birlik aniqlikdagi haqiqiy sonlardan iborat massiv yaratish
float_massiv = np.array([1.0, 2.0, 3.0, 4.0], dtype=np.float32)
print(float_massiv)
# Natija: [1. 2. 3. 4.]
```

```
#Kompleks son (complex) massiv
# 64 bitli yagona aniqlikdagi kompleks sonlardan iborat massiv yaratish
complex_massiv = np.array([1+2j, 3+4j, 5+6j], dtype=np.complex64)
print(complex_massiv)
# Natija: [1.+2.j 3.+4.j 5.+6.j]
```

Yuqorida berilgan ma'lumot turlaridan tashqari, foydalanuvchi tomonidan murakkab ma'lumot turlarini aniqlash va foydalanish mumkin:

```
# Murakkab ma'lumot turini yaratish
dt = np.dtype([('name', 'S10'), ('age', 'i4'), ('height', 'f4')])
```

```
# Ushbu ma'lumot turidan foydalanib massiv yaratish
complex_dtype_massiv = np.array([('Alice', 25, 5.5), ('Bob', 30, 6.0)], dtype=dt)
print(complex_dtype_massiv)
# Natija: [(b' Alice', 25, 5.5) (b' Bob', 30, 6. )]
```

NumPy massivlari ro'yxatlar asosida qurilganligi sababli ularning ba'zi xususiyatlarini saqlab qolgan. Xususan, massiv elementlariga yuqorida aytib o'tilgan usullarning istalgan birida tsikl yordamida murojaat qilish mumkin, bundan tashqari `len` funksiyasidan massivning uzunligini aniqlashda foydalanish mumkin. Massivlarning muhim xususiyati shundaki, ular ustida oddiy sonlar bilan bajariladigan amallarni bajarish imkoniyati mavjud, bunda massivning har bir elementga nisbatan muayyan amal bajariladi. Masalan, sinusni hisoblash amali qo'llanilsa, bu amal massivning har bir elementga nisbatan qo'llaniladi.

NumPy moduli `math` modulining funksiyalarini qo'llagan holda uning imkoniyatlarini kengaytiradi, u barcha asosiy trigonometrik, logarifmik va boshqa funksiyalar, shuningdek, konstantalarni o'z ichiga oladi. Shuning uchun, NumPy bilan ishlaganda `math` moduliga odatda ehtiyoj qolmaydi.

Massivlarning muhim xususiyatlaridan biri — qism massiv yaratish imkoniyati (qarang rasm 4.2) hisoblanadi. Qism massiv – bu boshlang'ich massivning ba'zi elementlaridan iborat massiv bo'lib, massivning ma'lum bir elementlarini qirqib olinadi.

NumPy yordamida massivlarni yaratish va ularga ishlov berishning bir nechta asosiy usullari mavjud. Quyida massivlarni yaratish va ular bilan turli amallarni bajarish misollari keltirilgan.

```
import numpy as np
#Matritsa ustida amallar bajarish

# Matritsalarini qo'shish
matritsa1 = np.array([[1, 2, 3], [4, 5, 6], [7, 8, 9]])
matritsa2 = np.array([[9, 8, 7], [6, 5, 4], [3, 2, 1]])
natija = matritsa1 + matritsa2
print(natija)
# Natija:
# [[10 10 10]
#  [10 10 10]
#  [10 10 10]]

# Matritsani transponirlash (qatorlarni ustunlarga o'zgartirish)
transpon= matritsa1.T
print(transpon)
# Natija:
# [[1 4 7]
#  [2 5 8]
#  [3 6 9]]
```

NumPy massivlari bilan ishlaganda, qism massiv (slice) yaratish asosiy massiv bilan xotirani bo'lishadi. Bu shuni anglatadiki, elementlar yangi massivga nusxa ko'chirilmaydi, balki murakkab manzil yaratish orqali ishlaydi. Shuning uchun, asosiy massiv elementlarini o'zgartirish kesim massiv elementlarini ham o'zgartiradi va aksincha. Elementlarni nusxalash uchun `copy` funksiyasidan foydalanish kerak.

```
import numpy as np
# Qism massiv va xotirani bo'lishish
# asosiy massiv
massiv = np.array([1, 2, 3, 4, 5])

# qism massiv yaratish
kesim = massiv[1:4]
```

```

print("Qism massiv:", kesim)
# Natija: Qism massiv: [2 3 4]

# qism massivni o'zgartirish
kesim[0] = 20
print("O'zgartirilgan kesim:", kesim)
# Natija: O'zgartirilgan kesim: [20 3 4]

# asosiy massivni tekshirish
print("O'zgartirilgan massiv:", massiv)
# Natija: O'zgartirilgan massiv: [ 1 20 3 4 5]

#### Massivni nusxalash
# asosiy massiv
massiv = np.array([1, 2, 3, 4, 5])

# qism massiv yaratish
kesim = massiv[1:4].copy()
print("Qism massiv:", kesim)
# Natija: Qism massiv: [2 3 4]

# qism massivni o'zgartirish
kesim[0] = 20
print("O'zgartirilgan qism massiv:", kesim)
# Natija: O'zgartirilgan qism massiv: [20 3 4]

# asosiy massivni tekshirish
print("Asosiy massiv:", massiv)
# Natija: Asosiy massiv: [1 2 3 4 5]

```

Yuqoridagi misollarda ko‘rib turganingizdek, qism massiv asosiy massiv bilan xotirani bo‘lishadi, shuning uchun biridagi o‘zgarish ikkinchisiga ham ta'sir qiladi. Agar massiv elementlarini nusxalash kerak bo‘lsa, `copy` funksiyasidan foydalanish kerak. Bu usul elementlarni yangi xotira maydoniga nusxalaydi va ular orasidagi bog‘lanishni yo‘q qiladi.

			A[0, 3:]			
	7	4	-6	7	4	2
A[:, 0]	6	-1	-2	-3	-7	0
	9	0	-8	3	-5	-9
	2	8	-9	6	2	7
			A[2:, 2:5]			

4.2-rasm. Qism massivlar: ko'k rang bilan belgilanganlar — natijada ikki o'lchamli massivdan bir o'lchamli massivlar hosil bo'ladigan kesimlar (boshlang'ich massivning ustuni va qatori mos ravishda); yashil rang bilan belgilangan — natijada kichikroq o'lchamli ikki o'lchamli massiv hosil bo'ladigan kesim.

Yuqoridagi rasmda NumPy massivining turli qim massivlari ko'rsatilgan. Keling, ushbu kesimlarni Python kodida qanday amalga oshirish tartibini ko'rib chiqamiz.

```
import numpy as np
```

```
# Dastlabki massivni yaratish
```

```
A = np.array([
    [7, 4, -6, 7, 4, 2],
    [6, -1, -2, -3, -7, 0],
    [9, 0, -8, 3, -5, -9],
    [2, 8, -9, 6, 2, 7]
])
```

```
# A[:, 0] - birinchi ustunni olish
```

```
kesim1 = A[:, 0]
print("A[:, 0]:")
print(kesim1)
```

```
# A[0, 3:] - birinchi qatorning to'rtinchi elementidan boshlab olamiz
```

```
kesim2 = A[0, 3:]
print("A[0, 3]:")
print(kesim2)
```

```
# A[2:, 2:5] - uchinchi qatordan boshlab va uchinchi ustundan beshinchi ustungacha olamiz
```

```
kesim3 = A[2:, 2:5]
print("A[2:, 2:5]:")
print(kesim3)
```

#Kod natijalari

#A[:, 0] - Birinchi ustunni olish

A[:, 0]:

[7 6 9 2]

A[0, 3:] - Birinchi qatorning to'rtinchi elementidan boshlab

A[0, 3]:

[7 4 2]

#A[2:, 2:5] - Uchinchi qatordan boshlab va uchinchi ustundan beshinchi ustungacha

A[2:, 2:5]:

[[-8 3 -5]

[-9 6 2]]

Ushbu kod qism massivlarni qanday amalga oshirish mumkinligini namoyish qiladi.

Python – obyektga yo'naltirilgan dasturlash tili bo'lganligi uchun har bir Python o'zgaruvchisi – obyekt sanaladi, lekin bu dasturlash jarayonida ko'pincha sezilmaydi. Shuning uchun ko'plab Python o'zgaruvchilari o'ziga xos funksiyalari va xususiyatlariga ega bo'lgan metodlarga ega. Masalan, massivlar juda oddiy, lekin foydali metodlarga ega bo'lib, murakkab dasturlar yozishda dasturlash va hisoblash jarayonini sezilarli darajada osonlashtiradi. Shuning uchun, murakkab dasturlarda massivlardan foydalanish ro'yxatlarga qaraganda qulayroq. Massivlar ustida bajariladigan eng ko'p ishlatiladigan metodlar quyida keltirilgan. Aytaylik, bizda quyidagi massiv bor deb faraz qilamiz: `a = numpy.array([0.1, 0.25, -1.75, 0.4, -0.9])`.

$$\sum_{i=0}^{\text{len}(a)} a_i = 0.1 + 0.25 - 1.75 + 0.4 - 0.9 = -1.9$$

#Summa (sum`metodi) – Massiv elementlarining yig'indisini hisoblash.

summa = a.sum()

print("Summasi:", summa)

Natija: Summasi: -1.9

$$\bar{a} = \frac{1}{\text{len}(a)} \sum_{i=0}^{\text{len}(a)} a_i = (0.1 + 0.25 - 1.75 + 0.4 - 0.9)/5 = -0.38$$

O'rtacha qiymat (mean`metodi) – Massiv elementlarining o'rtacha qiymatini hisoblash.

```
ortacha = a.mean()
print("O'rtacha qiymat:", ortacha)
# Natija: O'rtacha qiymat: -0.38
```

```
#Minimum (`min` metodi) – Massivdagi eng kichik qiymatni aniqlash.
min_qiymat = a.min()
print("Eng kichik qiymat:", min_qiymat)
# Natija: Eng kichik qiymat: -1.75
```

```
# Maximum (`max` metodi) – Massivdagi eng katta qiymatni aniqlash.
max_qiymat = a.max()
print("Eng katta qiymat:", max_qiymat)
# Natija: Eng katta qiymat: 0.4
```

```
# Saralash (`sort` metodi) – Massiv elementlarini o'sish tartibida saralash.
a.sort()
print("Saralangan massiv:", a)
# Natija: Saralangan massiv: [-1.75 -0.9  0.1  0.25  0.4 ]
#Elementlarga murojaat qilish – Massivdagi elementlarga indekslar orqali
murojaat qilish mumkin.
birinchi_element = a[0]
oxirgi_element = a[-1]
print("Birinchi element:", birinchi_element)
print("Oxirgi element:", oxirgi_element)
# Natija:
# Birinchi element: -1.75
# Oxirgi element: 0.4
```

```
#Elementlarni o'zgartirish
Massivdagi elementlarni o'zgartirish.
a[0] = 1.0
print("O'zgartirilgan massiv:", a)
# Natija: O'zgartirilgan massiv: [ 1. -0.9  0.1  0.25  0.4 ]
```

Bu metodlar NumPy massivlarining ko'p ishlatiladigan asosiy metodlari bo'lib, ular yordamida murakkab matematik va statistik amallarni bajarish osonlashadi. NumPy moduli kuchli va keng qamrovli vosita bo'lib, dasturlash jarayonida katta qulaylik yaratadi.

Python dasturlash tilida, har bir funksiyaning nomidan keyin keltirilgan qavslar ularning chaqirilishini va bajarilishini bildiradi. Agar qavslar unutib qoldirilsa, hech qanday hisoblash amalga oshirilmaydi va natijalar o‘rniga tegishli funksiyalarning hujjatlari (dokumantatsiya) chop etiladi. Masalan, `a.min()` bu hisoblash natijasi, ya'ni massiv `a` ning eng kichik elementi, lekin `a.min` bu hisoblash usulining o‘zini bildiradi.

Quyida bu farqni tushuntiruvchi misollar keltirib o‘tamiz:

```
# Funksiyani chaqirish (hisoblash amalga oshiriladi)
```

```
min_element = a.min()
```

```
print("Min element:", min_element)
```

```
# Natija: Min element: -1.75
```

```
# Funksiyani chaqirmaslik (faqat metodning o'zi)
```

```
min_metod = a.min
```

```
print("Min metod:", min_metod)
```

```
# Natija: Min metod: <built-in method min of numpy.ndarray object at 0x...>
```

Yuqoridagi misolda, dastur `a.min()` funksiyani chaqiradi va massivdagi eng kichik elementni qaytaradi, `a.min` esa `min` metodining o‘zini qaytaradi.

Boshqa metodlarda ham xuddi shunday farq mavjud.

Ushbu farqni bilish dasturlash jarayonida juda muhimdir, chunki funksiyalarning noto‘g‘ri ishlatilishi kutilgan natijalarni qaytarmasligi va xatoliklarga olib kelishi mumkin. Funksiya metodlarini to‘g‘ri chaqirish dastur kodining to‘g‘ri ishlashi uchun zarurdir.

Murakkab hisoblashlar ko‘pincha ikki o‘lchamli massivlarsiz, ya'ni matritsalarsiz amalga oshirilmaydi. NumPy moduli ixtiyoriy o‘lchamdagi massivlar bilan ishlash imkoniyatini beradi: bir o‘lchamli, ikki o‘lchamli, uch o‘lchamli va hokazo. Ixtiyoriy o‘lchamdagi massivlar bilan ishlash uchun "shape" (shakl) tushunchasidan foydalanish qulaydir. Shakl — bu massivning har bir o‘lchami bo‘yicha uzunliklar to‘plamidir. Odatda shakl bir nechta sonlardan iborat korset (tuple) ko‘rinishida beriladi: ikki o‘lchamli massiv uchun ikkita, uch o‘lchamli massiv uchun uchta va hokazo. Shakl — bu massivning xususiyati va shape bilan belgilanadi. Massivning shaklini uni yaratishda belgilash mumkin. Agar massiv NumPy ning biror o‘rnatilgan funksiyasi yordamida yaratilsa, uning shakli bu funksiya tomonidan belgilanadi. Bir o‘lchamli massivlar uchun shakl — bu faqat uning uzunligidir.

NumPy yordamida ikki o‘lchamli massivlarga ham `sum`, `mean`, `max` va boshqa amallarni qo‘llash mumkin. Bu amallar, agar `axis` parametri ko‘rsatilmagan bo‘lsa, butun massivga nisbatan qo‘llaniladi va massivning shaklidan qat'i nazar

barcha elementlarni ko‘rib chiqadi. Biroq, agar `axis` parametri ko‘rsatilsa, amal ko‘rsatilgan o‘q (axis) bo‘yicha qo‘llaniladi.

```
import numpy as np
# Ikki o'lchamli massiv yaratish
A = np.array([
    [1, 2, 3],
    [4, 5, 6],
    [7, 8, 9]
])
print("Berilgan massiv:")
print(A)
# Natija:
# [[1 2 3]
#  [4 5 6]
#  [7 8 9]]
#Summa (`sum` metodi) - Butun massiv bo'yicha
total_sum = A.sum()
print("Butun massivning summasi:", total_sum)
# Natija: Butun massivning summasi: 45

#Qatorlar bo'yicha (`axis=1`)
row_sum = A.sum(axis=1)
print("Qatorlar bo'yicha summa:")
print(row_sum)
# Natija: Qatorlar bo'yicha summa: [ 6 15 24]

#Ustunlar bo'yicha (`axis=0`)
col_sum = A.sum(axis=0)
print("Ustunlar bo'yicha summa:")
print(col_sum)
# Natija: Ustunlar bo'yicha summa: [12 15 18]

#O'rtacha qiymat (`mean` metodi) – Butun massiv bo'yicha
total_mean = A.mean()
print("Butun massivning o'rtacha qiymati:", total_mean)
# Natija: Butun massivning o'rtacha qiymati: 5.0

#Qatorlar bo'yicha (`axis=1`)
row_mean = A.mean(axis=1)
```

```

print("Qatorlar bo'yicha o'rtacha qiymat:")
print(row_mean)
# Natija: Qatorlar bo'yicha o'rtacha qiymat: [2. 5. 8.]
# Ustunlar bo'yicha (`axis=0`)
col_mean = A.mean(axis=0)
print("Ustunlar bo'yicha o'rtacha qiymat:")
print(col_mean)
# Natija: Ustunlar bo'yicha o'rtacha qiymat: [4. 5. 6.]

# Maksimal qiymat (`max` metodi) –Butun massiv bo'yicha
total_max = A.max()
print("Butun massivning maksimal qiymati:", total_max)
# Natija: Butun massivning maksimal qiymati: 9

# Qatorlar bo'yicha (`axis=1`)
row_max = A.max(axis=1)
print("Qatorlar bo'yicha maksimal qiymat:")
print(row_max)
# Natija: Qatorlar bo'yicha maksimal qiymat: [3 6 9]

# Ustunlar bo'yicha (`axis=0`)
col_max = A.max(axis=0)
print("Ustunlar bo'yicha maksimal qiymat:")
print(col_max)
# Natija: Ustunlar bo'yicha maksimal qiymat: [7 8 9]

# Minimal qiymat (`min` metodi) –Butun massiv bo'yicha
total_min = A.min()
print("Butun massivning minimal qiymati:", total_min)
# Natija: Butun massivning minimal qiymati: 1

# Qatorlar bo'yicha (`axis=1`)
row_min = A.min(axis=1)
print("Qatorlar bo'yicha minimal qiymat:")
print(row_min)
# Natija: Qatorlar bo'yicha minimal qiymat: [1 4 7]

# Ustunlar bo'yicha (`axis=0`)
col_min = A.min(axis=0)

```

```
print("Ustunlar bo'yicha minimal qiymat:")
print(col_min)
# Natija: Ustunlar bo'yicha minimal qiymat: [1 2 3]
```

Yuqoridagi misollar ko'rish mumkinki, `sum`, `mean`, `max`, `min` va boshqa amallarni `axis` parametri yordamida ko'rsatilgan o'q bo'yicha qo'llash mumkin. Bu usul ikki o'lchamli massivlar bilan matematik amallarni bajarishni juda qulay va samarali qiladi.

NumPy yordamida bir nechta massivlarni turli o'qlar bo'ylab birlashtirish uchun `hstack` va `vstack` funksiyalaridan foydalanish mumkin. `hstack()` massivlarni boshlang'ich indeks bo'yicha, `vstack()` esa oxirgi indeks bo'yicha birlashtiradi.

```
from numpy import *
# Ikki massiv yaratish
a = array([[1, 2, 3], [4, 5, 6]])
b = array([[5, 6, 7], [8, 9, 10]])
# Vertikal birlashtirish (vstack)
vstacked = vstack((a, b))
print("Vertikal birlashtirilgan massiv:")
print(vstacked)
# Natija:
# [[ 1  2  3]
# [ 4  5  6]
# [ 5  6  7]
# [ 8  9 10]]
# Gorizental birlashtirish (hstack)
hstacked = hstack((a, b))
print("Gorizental birlashtirilgan massiv:")
print(hstacked)
# Natija:
# [[ 1  2  3  5  6  7]
# [ 4  5  6  8  9 10]]
```

Yuqoridagi kod quyidagi amallarni bajaradi:

1. Ikki massiv (`a` va `b`) yaratiladi.
2. `vstack` funksiyasi yordamida bu massivlar vertikal ravishda birlashtiriladi. Natijada yangi massivda qatorlar bo'yicha birlashtirilgan massiv hosil bo'ladi.
3. `hstack` funksiyasi yordamida bu massivlar gorizental ravishda birlashtiriladi. Natijada yangi massivda ustunlar bo'yicha birlashtirilgan massiv hosil bo'ladi.

NumPy moduli ko'p o'lchovli massivlar bilan ishlash uchun ko'plab turli funksiyalarni taqdim etadi, ulardan biri dot funksiyasi hisoblanadi. Bu funksiya matritsalar va bir o'lchamli massivlar (vektorlar) uchun matritsali ko'paytirishni bajaradi.

Numpy-da foydalanish uchun juda oson va juda foydali ikkita xususiyat mavjud bo'lib, ular sizga faqat o'quv vazifalarini bajaribgina qolmay amaliy vazifalarni bajarishni osonlashtiradi. Bu NumPyning `loadtxt` va `savetxt` funksiyalari bo'lib, ular yordamida fayllardan ma'lumotlarni o'qish va fayllarga yozish mumkin.

loadtxt funksiyasi yordamida fayldan ma'lumotlarni yuklash

`loadtxt` funksiyasi matnli fayldan ma'lumotlarni o'qib, massivga yuklaydi. Faylda ma'lumotlar bir qator yoki ustun bo'lsa, massiv bir o'lchamli bo'ladi, aks holda ikki o'lchamli massiv hosil bo'ladi.

Misol: Fayldan ma'lumotlarni yuklash

```
import numpy as np
# Fayldan ma'lumotlarni yuklash
data = np.loadtxt('data.txt')
print("Yuklangan ma'lumotlar:")
print(data)
```

savetxt funksiyasi yordamida ma'lumotlarni faylga yozish

`savetxt` funksiyasi massivni matnli faylga yozadi.

Misol: Massivni faylga yozish

```
import numpy as np
# Ikki o'lchamli massiv yaratish
data = np.array([
    [1.1, 2.2, 3.3],
    [4.4, 5.5, 6.6],
    [7.7, 8.8, 9.9]
])
# Massivni faylga yozish
np.savetxt('output.txt', data, fmt='%.2f')
print("Ma'lumotlar faylga yozildi.")
```

Keling, ma'lumotlarni fayldan yuklash va qayta faylga yozishni ko'rsatadigan murakkabroq misolni ko'rib chiqamiz:

1. `data.txt` faylidagi ma'lumotlar:

```
1.1 2.2 3.3
4.4 5.5 6.6
7.7 8.8 9.9
```

2. Python kod yordamida ma'lumotlarni yuklash, o'zgartirish va qayta yozish:

```

import numpy as np
# Fayldan ma'lumotlarni yuklash
data = np.loadtxt('data.txt')
print("Yuklangan ma'lumotlar:")
print(data)
# Ma'lumotlarni qayta ishlash (har bir elementga 10 sonini qo'shish)
modified_data = data + 10
print("O'zgartirilgan ma'lumotlar:")
print(modified_data)
# O'zgartirilgan ma'lumotlarni faylga yozish
np.savetxt('modified_data.txt', modified_data, fmt='%.2f')
print("O'zgartirilgan ma'lumotlar faylga yozildi.")
3. `modified_data.txt` faylidagi ma'lumotlar:
11.10 12.20 13.30
14.40 15.50 16.60
17.70 18.80 19.90

```

Faraz qilaylik, bir sinfda 10 nafar o‘quvchi matematika fanidan test topshirdi. O‘qituvchi quyidagilarni bilmoqchi:

- o‘rtacha test bali;
- eng yuqori va eng past natija;
- sinfdagi baholarning barqarorligi (standart og‘ish).

Bu vazifa **ta’lim jarayonini tahlil qilish**, ya’ni o‘quvchilar o‘zlashtirish darajasini aniqlash uchun juda muhimdir.

```

import numpy as np
# O‘quvchilarning test ballari
ballar = np.array([56, 62, 70, 75, 80, 85, 90, 68, 73,
78])
# Asosiy statistik ko‘rsatkichlar
ortalama = np.mean(ballar)
eng_katta = np.max(ballar)
eng_kichik = np.min(ballar)
standart_ogish = np.std(ballar)
print("O‘rtacha ball:", ortalama)
print("Eng yuqori ball:", eng_katta)
print("Eng past ball:", eng_kichik)
print("Standart og‘ish:", standart_ogish)

```

Ushbu misolda avvalo, `import numpy as np` buyrug'i orqali NumPy kutubxonasi chaqiriladi. NumPy sonli ma'lumotlar bilan tez va samarali ishlash imkonini beradi. Keyingi qadamda o'quvchilarning test natijalari **bir o'lchovli massiv** (`array`) ko'rinishida kiritiladi. Bu massivda har bir element bitta o'quvchining olgan ballini ifodalaydi.

`np.mean()` funksiyasi yordamida o'rtacha ball hisoblanadi. Bu ko'rsatkich sinfning umumiy o'zlashtirish darajasini baholash uchun xizmat qiladi. Agar o'rtacha ball past bo'lsa, o'qituvchi mavzuni qayta tushuntirish yoki qo'shimcha mashg'ulotlar tashkil etish zarurligini anglaydi.

`np.max()` va `np.min()` funksiyalari eng yuqori va eng past natijalarni aniqlaydi. Eng past ball olgan o'quvchilar bilan individual ishlash, eng yuqori ball olganlarni esa rag'batlantirish mumkin. Bu differensial yondashuvni amalga oshirishda muhim ahamiyatga ega.

`np.std()` funksiyasi orqali hisoblangan standart og'ish baholarning tarqalishini ko'rsatadi. Agar standart og'ish kichik bo'lsa, demak o'quvchilarning bilim darajasi bir-biriga yaqin. Agar u katta bo'lsa, sinfda bilim darajasi notekis ekanini bildiradi. Bu esa o'qituvchiga ta'lim strategiyasini qayta ko'rib chiqishga yordam beradi.

Xulosa qilib aytganda, NumPy moduli Python dasturlash tilida matematik va statistik amallarni bajarish uchun qulay va samarali vositadir. NumPy yordamida massivlarni yaratish va ularga ishlov berish, o'zining boy funksiyalar to'plami tufayli, murakkab hisoblashlar va ma'lumotlar tahlilini osonlashtiradi.

Ma'lumotlarni tahlil qilish zamonaviy jamiyatda axborot texnologiyalarining ajralmas qismi bo'lib, biznes, ilmiy tadqiqotlar, davlat boshqaruvi va boshqa sohalarda keng qo'llaniladi. Ma'lumotlarni tahlil qilish orqali foydalanuvchilar katta hajmdagi ma'lumotlardan foydali ma'lumotlar olish, qaror qabul qilish jarayonlarini optimallashtirish va natijada samaradorlikni oshirish imkoniyatiga ega bo'lishadi.

Ma'lumotlarni tahlil qilish bir necha bosqichlardan iborat murakkab jarayon bo'lib, har bir bosqich o'ziga xos vazifalarni bajaradi va oxirgi natijaga ta'sir ko'rsatadi. Ma'lumotlarni yig'ish jarayoni tahlil qilish uchun zarur bo'lgan ma'lumotlarni to'plashni o'z ichiga oladi. Bu bosqichda ma'lumotlar turli manbalardan, masalan, anketa, sensorlar, veb-saytlar, ijtimoiy tarmoqlar va boshqalardan yig'iladi. Ma'lumotlarni yig'ish jarayoni to'plangan ma'lumotlarning sifatiga va ishonchligiga katta ta'sir ko'rsatadi.



Ma'lumotlar yig'ilgandan so'ng, ular ko'pincha tozalashni talab qiladi. Bu bosqichda noto'g'ri, to'liq bo'lmagan yoki qayta ishlanmagan ma'lumotlar aniqlanadi va tozalanadi. Ma'lumotlarni tozalash va tayyorlash jarayoni tahlil qilishning ishonchliligini oshiradi va natijalarni to'g'ri interpretatsiya qilish imkonini beradi. Bu bosqichda ma'lumotlarni normalizatsiya qilish, to'ldirish, filtratsiya va boshqa metodlar qo'llaniladi.

Ma'lumotlarni tahlil qilish bosqichida ma'lumotlar tahlil qilinadi va foydali ma'lumotlar olinadi. Ma'lumotlarni tahlil qilish jarayoni statistik metodlar, mashinani o'rganish algoritmlari va boshqa tahlil usullarini o'z ichiga oladi. Bu bosqichda ma'lumotlarning xususiyatlari aniqlanadi, o'zgarishlar va tendensiyalar tahlil qilinadi, hamda bashoratlar va tavsiyalar ishlab chiqiladi. Ma'lumotlarni vizuallashtirish tahlil natijalarini ko'rgazmali va tushunarli tarzda taqdim etish imkonini beradi. Bu bosqichda grafiklar, diagrammalar, jadvallar va boshqa vizual vositalar yordamida ma'lumotlar taqdim etiladi. Ma'lumotlarni vizuallashtirish qaror qabul qilish jarayonini osonlashtiradi va tahlil natijalarini kengroq auditoriyaga yetkazish imkonini beradi.

Oxirgi bosqichda tahlil natijalari talqin qilinadi va asoslangan qarorlar qabul qilinadi. Bu bosqichda tahlil natijalari asosida muammolar aniqlanadi, imkoniyatlar baholanadi va navbatdagi vazifalar uchun strategiyalar ishlab chiqiladi. Ma'lumotlarni tahlil qilish natijalari samarali qaror qabul qilish va biznes jarayonlarini optimallashtirishga yordam beradi.

Ma'lumotlarni tahlil qilishda turli xil vositalar va metodlar qo'llaniladi. Ushbu vositalar va metodlar tahlil jarayonini samarali va aniq bajarishga yordam beradi. Python dasturlash tili ma'lumotlarni tahlil qilishda keng qo'llaniladi. Bu tilning qulayligi, boy kutubxonalari va jamoatchilik tomonidan qo'llab-quvvatlanishi uni ma'lumotlarni tahlil qilishda ommaviy vositaga aylantirdi. Pandas, NumPy, SciPy, Matplotlib, Seaborn va Scikit-Learn kabi kutubxonalar ma'lumotlarni yig'ish, tozalash, tahlil qilish va vizuallashtirish uchun qulay vosita hisoblanadi.

Pandas kutubxonasi strukturalangan ma'lumotlar bilan ishlash uchun mo'ljallangan bo'lib, bu kutubxona DataFrame va Series kabi ma'lumot tuzilmalari bilan ishlaydi, ushbu tuzilmalar ma'lumotlarni oson boshqarish va tahlil qilish

imkonini beradi. Pandas yordamida ma'lumotlarni yuklash, tozalash, guruhlash, filtratsiya qilish va boshqa amallarni bajarish mumkin. NumPy kutubxonasi ham ma'lum bir darajada ma'lumotlarni qayta ishlash jarayoni uchun qo'llash mumkin.

Ma'lumotlarni tahlil qilishda uning grafik iterepretatsiyasi muhim rol o'ynaydi. Python dasturlash tilida Matplotlib va Seaborn kutubxonalari mana shu maqsad uchun mo'ljallangan. Matplotlib yordamida grafikalar, diagrammalar va jadvallar yaratish mumkin. Seaborn esa Matplotlib ustiga qurilgan va statistik grafikalarini osonroq yaratish imkonini beradi. Bu kutubxonalar yordamida tahlil natijalarini ko'rgazmali va tushunarli tarzda taqdim etish mumkin. Bundan tashqari Scikit-Learn kutubxonasi mavjud bo'lib, bu kutubxona mashinali o'qitish algoritmlari bilan ishlash uchun mo'ljallangan. Ushbu kutubxona regressiya, klassifikatsiya, klasterlash va boshqa mashinali o'qitish amallarini bajarish uchun keng imkoniyatlar taqdim etadi. Scikit-Learn yordamida ma'lumotlarni tahlil qilish, model yaratish va basoratlar qilish mumkin.

Ma'lumotlarni tahlil qilish iqtisodiyot tarmoqlarining turli sohalarida keng qo'llanilmoqda va iqtisodiy samaradorlikni oshirishga yordam bermoqda. Yirik kompaniyalar iqtisodiy ma'lumotlarini tahlil qilish orqali o'z faoliyatini optimallashtirish, xarajatlarni kamaytirish va daromadni oshirish imkoniga ega bo'lishmoqda. Ma'lumotlarni tahlil qilish yordamida mijozlar ehtiyojlarini aniqlash, marketing strategiyalarini takomillashtirish va raqobatchilarni kuzatish mumkin. Ilmiy tadqiqotlarda ma'lumotlarni tahlil qilish yangi kashfiyotlar va innovatsiyalar uchun asos bo'lib xizmat qiladi. Ma'lumotlarni tahlil qilish orqali ilmiy eksperimentlar natijalarini tahlil qilish, statistik modellarning aniqligini baholash va yangi gipotezalarni tekshirish va shu asosda xulosalar chiqarish imkonini beradi. Davlat boshqaruvida ma'lumotlarni tahlil qilish siyosiy qarorlarni qabul qilish jarayonini optimallashtiradi. Ma'lumotlarni tahlil qilish natijasida iqtisodiy ko'rsatkichlarni kuzatish, ijtimoiy so'rovlarni tahlil qilish va aholining ehtiyojlarini aniqlash mumkin. Sog'liqni saqlash sohasida ma'lumotlarni tahlil qilish bemorlarning sog'lig'ini yaxshilash va tibbiy xizmatlarni optimallashtirish imkonini beradi. Ma'lumotlarni tahlil qilish bilan epidemiyalarni kuzatish, kasalliklarni erta aniqlash va davolash usullarini takomillashtirish mumkin. Moliyaviy tahlilda ma'lumotlarni tahlil qilish investitsion qarorlarni qabul qilish jarayonini optimallashtiradi. Ma'lumotlarni tahlil qilish orqali moliyaviy ko'rsatkichlarni baholash, investitsiya strategiyalarini ishlab chiqish va risklarni boshqarish mumkin.

Umuman olganda, ma'lumotlarni tahlil qilish zamonaviy jamiyatda muhim rol o'ynaydi. Ma'lumotlarni yig'ish, tozalash, tahlil qilish, vizuallashtirish va talqin qilish jarayonlari orqali foydalanuvchilar katta hajmdagi ma'lumotlardan foydali ma'lumotlar olish imkoniga ega bo'lishadi. Pandas, NumPy, Matplotlib, Seaborn va Scikit-Learn kabi vositalar ma'lumotlarni tahlil qilishda katta yordam beradi.

Ma'lumotlarni tahlil qilish turli sohalarda keng qo'llanilib, samaradorlikni oshirishga, qaror qabul qilish jarayonini optimallashtirishga va yangi imkoniyatlarni aniqlashga yordam beradi. Ma'lumotlarni tahlil qilish kelajakda ham muhim ahamiyatga ega bo'lib qoladi va yangi texnologiyalar yordamida yanada rivojlanadi.

Ma'lumotlarni yig'ish

Ma'lumotlarni yig'ish jarayoni tahlil qilish uchun zarur bo'lgan ma'lumotlarni to'plashni o'z ichiga oladi. Bu bosqich ma'lumotlar tahlilining eng muhim va asosiy qismidir, chunki yig'ilgan ma'lumotlarning sifati va ishonchliligi tahlil natijalarining aniqligiga katta ta'sir ko'rsatadi. Ma'lumotlarni yig'ish jarayoni turli manbalardan, usullardan va vositalardan foydalanishni talab qiladi. Quyida ma'lumotlarni yig'ishning asosiy usullari va ularning xususiyatlari keltirilgan.

1. Anketalar va so'rovnomalar

Anketalar va so'rovnomalar orqali ma'lumotlarni yig'ish keng tarqalgan usullardan biridir. Bu usul respondentlarning fikrlarini, xulq-atvorini va ehtiyojlarini aniqlash uchun foydalaniladi. Anketalar onlayn yoki oflayn tarzda o'tkazilishi mumkin. Onlayn anketalar uchun Google Forms, SurveyMonkey va boshqa vositalar qo'llaniladi. Anketalarni o'tkazishda quyidagi omillarga e'tibor berish kerak:

- Savollar aniq va tushunarli bo'lishi kerak.
- Savollar ketma-ketligi va mantiqiyliqi saqlanishi kerak.
- Respondentlarning shaxsiy ma'lumotlari himoyalaniishi kerak.

2. Sensorlar va IoT qurilmalari

Sensorlar va Internet of Things (IoT) qurilmalari yordamida ma'lumotlarni yig'ish zamonaviy texnologiyalarning yutuqlaridan biridir. Bu usul real vaqt rejimida katta hajmdagi ma'lumotlarni yig'ish imkonini beradi. Sensorlar turli xil fizik parametrlarni, masalan, harorat, namlik, bosim, harakat va boshqalarni o'lchaydi va ma'lumotlarni markaziy tizimga uzatadi. IoT qurilmalari yordamida yig'ilgan ma'lumotlar turli sohalarda, jumladan, sanoat, qishloq xo'jaligi, sog'liqni saqlash va transport sohasida qo'llaniladi.

3. Veb-saytlar va ijtimoiy tarmoqlar

Veb-saytlar va ijtimoiy tarmoqlardan ma'lumotlarni yig'ish ko'plab tadqiqotlar va biznes tahlillari uchun muhim manbadir. Bu usul veb-sayt foydalanuvchilari haqidagi ma'lumotlarni, jumladan, ularning xulq-atvori, qiziqishlari va fikrlarini o'rganish imkonini beradi. Web scraping texnikasi yordamida veb-saytlardan avtomatik ravishda ma'lumotlar yig'iladi. BeautifulSoup, Scrapy va Selenium kabi kutubxonalar va vositalar web scraping uchun keng qo'llaniladi. Ijtimoiy tarmoqlardan ma'lumotlarni yig'ish esa API'lar orqali amalga oshiriladi. Masalan,

Twitter API, Facebook Graph API va Instagram API yordamida ijtimoiy tarmoqlardan ma'lumotlarni olish mumkin.

4. Ma'lumotlar bazalari va omborlar

Ma'lumotlar bazalari va omborlar ma'lumotlarni yig'ish va saqlashning asosiy manbalaridan biridir. Bu usul ko'pincha korxonalarda va tashkilotlarda qo'llaniladi, chunki ular katta hajmdagi ma'lumotlarni samarali saqlash va boshqarish imkonini beradi. Ma'lumotlar bazalaridan SQL (Structured Query Language) yordamida ma'lumotlarni olish mumkin. MySQL, PostgreSQL va SQLite kabi ma'lumotlar bazalari tahlilchilar orasida keng tarqalgan. Ma'lumotlar omborlari esa katta hajmdagi ma'lumotlarni saqlash va qayta ishlash uchun mo'ljallangan. Apache Hadoop va Apache Spark kabi vositalar ma'lumotlar omborlari bilan ishlashda qo'llaniladi.

5. Ijtimoiy ma'lumotlar

Jamoatchilik ma'lumotlari ochiq manbalar orqali taqdim etilgan ma'lumotlardan tashkil topadi. Bu ma'lumotlar ko'pincha hukumat tashkilotlari, xalqaro tashkilotlar va tadqiqot institutlari tomonidan taqdim etiladi. Jamoatchilik ma'lumotlari ko'plab sohalarda tahlil qilish uchun muhim manba bo'lib xizmat qiladi. Ochiq ma'lumotlar portallari, masalan, data.gov va World Bank Open Data, tahlilchilar uchun katta hajmdagi ma'lumotlarni taqdim etadi.

Ma'lumotlarni yig'ishda bir nechta muhim omillarni e'tiborga olish zarur, ya'ni:

- Ma'lumotlarning sifati: Yig'ilayotgan ma'lumotlar aniq, to'liq va ishonchli bo'lishi kerak. Noto'g'ri yoki to'liq bo'lmagan ma'lumotlar tahlil natijalarining aniqligiga ta'sir ko'rsatishi mumkin.

- Maxfiylik va xavfsizlik: Ma'lumotlarni yig'ishda shaxsiy ma'lumotlarni himoya qilish va xavfsizlik choralarini ko'rish zarur. Ma'lumotlar maxfiyligini ta'minlash ma'lumotlar xavfsizligini saqlashda muhim ahamiyatga ega.

- Ma'lumotlarning xilma-xilligi: Turli manbalardan ma'lumotlarni yig'ish tahlil natijalarini boyitadi va har tomonlama yondashuvni ta'minlaydi. Bu turli xil ma'lumotlar manbalari o'rtasidagi bog'lanishlarni o'rganishga imkon beradi.

- Avtomatlashtirish: Katta hajmdagi ma'lumotlarni yig'ishda avtomatlashtirilgan vositalar va texnologiyalardan foydalanish samaradorlikni oshiradi va inson xatolarini kamaytiradi. Avtomatlashtirish vositalari ma'lumotlarni yig'ish jarayonini tezlashtiradi va natijalarni aniqroq qiladi.

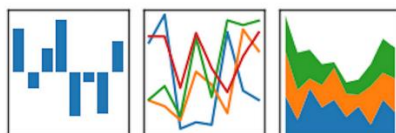
Ma'lumotlarni yig'ish jarayoni ma'lumotlarni tahlil qilishning eng muhim bosqichlaridan biri bo'lib, uning sifati va ishonchligi tahlil natijalarining aniqligiga katta ta'sir ko'rsatadi. Ma'lumotlarni yig'ishning turli usullari va vositalari mavjud bo'lib, har biri o'ziga xos xususiyatlarga ega. Anketalar va so'rovnomalar, sensorlar va IoT qurilmalari, veb-saytlar va ijtimoiy tarmoqlar, ma'lumotlar bazalari va omborlar, ijtimoiy ma'lumotlar — bularning barchasi ma'lumotlarni yig'ishda muhim manbalar

hisoblanadi. Ma'lumotlarni yig'ishda ma'lumotlarning sifati, maxfiylik va xavfsizlik, ma'lumotlarning xilma-xilligi va avtomatlashtirish omillariga e'tibor qaratish zarur. Bu omillar yig'ilgan ma'lumotlarning aniq va ishonchli bo'lishini ta'minlaydi va tahlil natijalarining aniqligiga hissa qo'shadi. Ma'lumotlarni tahlil qilish jarayonining samaradorligi to'g'ri va sifatli ma'lumotlarni yig'ishga bog'liq. Shu sababli, ma'lumotlarni yig'ish jarayoniga katta e'tibor qaratish va bu jarayonni samarali tashkil etish muhimdir.

Pandas kutubxonasi - bu Python dasturlash tilida yozilgan va ma'lumotlarni tahlil qilish uchun keng qo'llaniladigan kutubxona. Pandas, ayniqsa, strukturalangan ma'lumotlar (jadval ma'lumotlari) bilan ishlashda juda qulay va samarali vositadir. Pandas NumPy kutubxonasi asosida qurilgan va u yuqori darajadagi ma'lumotlar tuzilmalarini taqdim etadi.

pandas

$$y_{it} = \beta' x_{it} + \mu_i + \epsilon_{it}$$



Pandas kutubxonasi dastlab AQR Capital Management kompaniyasida Wes McKinney tomonidan ishlab chiqilgan. Pandas loyihasi 2008-yilda boshlangan. Wes McKinney AQR Capital Management kompaniyasida ishlayotgan vaqtda kompaniyada mavjud bo'lgan ma'lumotlar tahlili vositalarining yetarli emasligini his qildi. Shu sababli, u Python tilida kuchli, moslashuvchan va foydalanish oson bo'lgan ma'lumotlar tahlili kutubxonasini yaratishni maqsad qildi. Pandasning dastlabki versiyalari ichki foydalanish uchun ishlab chiqilgan bo'lib, ular AQR kompaniyasida foydalanilgan. 2009-yilda Wes McKinney Pandas kutubxonasini ochiq manbali (open-source) loyiha sifatida ommaga taqdim etdi. Bu bilan boshqa dasturchilar ham loyihaga hissa qo'sha boshladi.

2010-yilda Pandas kutubxonasining birinchi barqaror versiyasi (0.1) ishlab chiqildi. Bu versiyada ma'lumotlar tahlili va manipulyatsiyasi uchun asosiy funksiyalar mavjud edi. Pandas kutubxonasi tez orada mashhurlikka erishdi va katta hamjamiyat tomonidan qo'llab-quvvatlandi. 2012-yilda Wes McKinney Pandas kutubxonasi haqida "Python for Data Analysis" kitobini nashr etdi. Bu kitob Pandas kutubxonasini o'rganmoqchi bo'lgan dasturchilar uchun muhim manba bo'ldi. 2015-yilda Pandas kutubxonasining 0.16 versiyasi ommaga taqdim etildi. Bu versiyada ko'plab yangi funksiyalar va yaxshilanishlar qo'shildi. Pandas kutubxonasi ma'lumotlar tahlili va manipulyatsiyasi uchun asosiy vositalardan biri sifatida mustahkam o'rin egalladi. 2020-yilda Pandas kutubxonasining 1.0 versiyasi

chiqarildi. Bu katta yutuq bo'lib, kutubxonaning barqaror va foydalanishga tayyorligini ko'rsatdi.

Pandas kutubxonasi hozirgi kunda ma'lumotlar tahlili, ilmiy tadqiqotlar va mashinali o'qitish loyihalarida keng qo'llaniladi. Kutubxona qulay interfeysga ega bo'lib, bu dasturchilarga ma'lumotlar bilan ishlashni osonlashtiradi. Pandas ma'lumotlar tahlili va manipulyatsiyasi uchun keng ko'lamlı funksiyalarni taqdim etadi, jumladan ma'lumotlarni yuklash, tozalash, o'zgartirish va vizuallashtirish imkoniyatiga ega. Pandas katta va faol hamjamiyatga ega bo'lib, foydalanuvchilar kutubxona haqida savollar berishlari va yordam olishlari mumkin.

Pandas kutubxonasi qisqa vaqt ichida ma'lumotlar tahlili va manipulyatsiyasi uchun muhim vositalardan biriga aylandi. Wes McKinney tomonidan yaratilgan va ochiq manbali loyiha sifatida ommaga taqdim etilgan Pandas kutubxonasi bugungi kunda ko'plab dasturchilar tomonidan qo'llanilmoqda va rivojlanmoqda. Pandas yordamida ma'lumotlar bilan ishlash ancha oson va samarali bo'lib, bu kutubxona ma'lumotlar tahlilining ajralmas qismiga aylandi.

Pandas kutubxonasidan foydalanish uchun uni import qilish zarur bo'ladi. Shundan so'ng Pandas funksiyalarini va metodlarini dasturda ishlatishga imkon beradi. Pandas kutubxonasini import qilish uchun quyidagi koddan foydalaniladi:

```
import pandas as pd
```

Bu yerda:

- `import` kalit so'zi Python'ga kutubxonani import qilishni buyuradi.
- `pandas` bu import qilinayotgan kutubxonaning nomi.
- `as pd` qisqartmasi Pandas kutubxonasini `pd` nomi bilan qisqartirib ishlatishga imkon beradi. Bu dastur kodini yanada qisqa va o'qilishini osonlashtiradi.

Python dasturlash tilida kutubxonalarning import qilinishi ularning funksiyalari va amallarini dasturga kiritish imkonini beradi. Pandas kutubxonasi ma'lumotlar bilan ishlash, tahlil qilish va manipulyatsiya qilish uchun ko'plab qulay funksiyalarni taqdim etadi.

Pandas kutubxonasi ikkita asosiy ma'lumot tuzilmasiga ega: `Series` va `DataFrame`. Bu tuzilmalar ma'lumotlarni tahlil qilish va manipulyatsiya qilish uchun qulay vositalarni taqdim etadi.

Pandas `Series`

`Series` - bu indekslangan bir o'lchovli massiv tuzilmasiga ega. Bu massivning har bir elementiga indeks bilan murojaat qilinadi. `Series` obyektini yaratish uchun `pandas.Series` funksiyasidan foydalaniladi.

```
import pandas as pd
```

```
# Series yaratish
```

```
data = [1, 2, 3, 4, 5]
```

```
series = pd.Series(data)
```

```
print(series)
```

Yuqoridagi kodda `Series` obyektini yaratilib, ekranga chiqariladi. Har bir elementga indeks bilan murojaat qilinadi. `Series` indeksleri avtomatik ravishda 0 dan boshlanadi. Shu bilan birga, foydalanuvchi tomonidan ham indekslarni belgilash mumkin:

```
import pandas as pd
```

```
# Indeksni belgilash
```

```
series = pd.Series(data, index=['a', 'b', 'c', 'd', 'e'])
```

```
print(series)
```

Bu holatda, `Series` obyektini belgilangan indekslar bilan yaratiladi.

Pandas `DataFrame`

`DataFrame` - bu indekslangan ikki o'lchovli massiv tuzilmasiga ega. Bu massiv jadval ko'rinishida bo'lib, qatorlar va ustunlardan iborat bo'ladi. `DataFrame` obyektini yaratish uchun `pandas.DataFrame` funksiyasidan foydalaniladi.

```
import pandas as pd
```

```
# DataFrame yaratish
```

```
data = { 'Ism': ['Ali', 'Vali', 'Hasan'],
```

```
        'Yosh': [23, 35, 45],
```

```
        'Shahar': ['Toshkent', 'Samarqand', 'Buxoro'] }
```

```
df = pd.DataFrame(data)
```

```
print(df)
```

Yuqoridagi kodda `DataFrame` obyektini yaratilib, ma'lumotlar kiritiladi va ekranga chiqariladi. DataFrame yordamida ma'lumotlar qatorlar va ustunlar ko'rinishida taqdim etiladi.

`DataFrame` ustunlari bilan ishlash:

Ustunlar bilan ishlashda, biz `DataFrame` ustunlarini osongina tanlashimiz, qo'shishimiz yoki o'chirishimiz mumkin.

```
import pandas as pd
```

```
# Ma'lumotlar to'plamini yaratish
```

```
data = { 'Ism': ['Ali', 'Vali', 'Hasan'],
```

```
        'Yosh': [23, 35, 45],
```

```
        'Shahar': ['Toshkent', 'Samarqand', 'Buxoro'] }
```

```

df = pd.DataFrame(data)

# Ustunni tanlash
print("Ism ustuni:")
print(df['Ism'])

# Yangi ustun qo'shish
df['Maosh'] = [5000, 6000, 7000]

print("\nYangi Maosh ustuni qo'shilgan DataFrame:")
print(df)

# Ustunni o'chirish
df.drop('Shahar', axis=1, inplace=True)

print("\nShahar ustuni o'chirilgan DataFrame:")
print(df)

```

```

Ism ustuni:
0    Ali
1    Vali
2    Hasan
Name: Ism, dtype: object

Yangi Maosh ustuni qo'shilgan DataFrame:
   Ism  Yosh  Shahar  Maosh
0  Ali   23  Toshkent  5000
1  Vali   35  Samarqand  6000
2  Hasan   45   Buxoro  7000

Shahar ustuni o'chirilgan DataFrame:
   Ism  Yosh  Maosh
0  Ali   23  5000
1  Vali   35  6000
2  Hasan   45  7000

```

Yuqoridagi kodda `Ism` ustuni tanlanadi, yangi `Maosh` ustuni qo'shiladi va `Shahar` ustuni o'chiriladi.

Pandas ma'lumot tuzilmalarida indekslar va etiketlar muhim rol o'ynaydi. Indekslar va etiketlar yordamida qator va ustunlarni osongina tanlash va manipulyatsiya qilish mumkin.

Pandas kutubxonasida etiketlar (labels) ma'lumotlarni tanlash va ularga murojaat qilishda muhim rol o'ynaydi. Etiketlar yordamida qator va ustunlarni osongina tanlash va manipulyatsiya qilish mumkin.

`Series` va `DataFrame` tuzilmalarining umumiy imkoniyatlari:

1. Filtrlash va tanlash: `Series` va `DataFrame` ob'ektlarida qator va ustunlarni tanlash va filtrlash imkoniyatlari mavjud.

2. Matematik operatsiyalar: Ikkala tuzilma ham matematik operatsiyalarni qo'llab-quvvatlaydi. Masalan, ustunlar yoki qatorlar bo'yicha arifmetik amallar bajarish mumkin.

3. Statistik hisob-kitoblar: Pandas turli statistik funksiyalarni taqdim etadi, masalan, o'rtacha qiymat, median, dispersiya va boshqalar.

Pandas kutubxonasining `Series` va `DataFrame` tuzilmalari ma'lumotlar tahlili va manipulyatsiyasi uchun juda qulay vositalar hisoblanadi. `Series` bir o'lchovli indekslangan massiv, `DataFrame` esa ikki o'lchovli indekslangan massiv bo'lib, ular ma'lumotlarni jadval ko'rinishida ko'rsatishga imkon beradi. Ushbu tuzilmalar yordamida ma'lumotlarni tahlil qilish, tozalash va manipulyatsiya qilish ancha oson va samarali bo'ladi.

Pandas kutubxonasida `Series` va `DataFrame` ob'ektlarining ko'plab qulay metodlari mavjud. Quyida `Series` va `DataFrame` ob'ektlarining asosiy metodlari va ularning ishlatilishi haqida batafsil ma'lumot keltirilgan.

`Series` ning asosiy metodlari

1. `head()` metodi `Series` ning dastlabki n ta qatorini qaytaradi. n ning standart qiymati qiymati 5 ga teng.

```
import pandas as pd
```

```
data = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
```

```
series = pd.Series(data)
```

```
print(series.head(3)) # Dastlabki 3 qatorni chiqarish
```

```
0    1
1    2
2    3
dtype: int64
```

2. `tail()` metodi `Series` ning oxirgi n ta qatorini qaytaradi. n ning standart qiymati qiymati 5.

```
print(series.tail(3)) # Oxirgi 3 qatorni chiqarish
```

```
7    8
8    9
9   10
dtype: int64
```

3. `describe()` metodi `Series` ning statistik tahlili natijasini qaytaradi (o'rtacha, minimal, maksimal, kvartillar va hokazo).

```
print(series.describe())
```

```

count    10.00000
mean     5.50000
std      3.02765
min      1.00000
25%     3.25000
50%     5.50000
75%     7.75000
max     10.00000
dtype: float64

```

4. `sum()` metodi `Series` ning barcha elementlarining yig'indisini qaytaradi.

```
print(series.sum()) #Natija: 55
```

5. `mean()` metodi `Series` ning o'rtacha qiymatini qaytaradi.

```
print(series.mean()) #Natija: 5.5
```

6. `value_counts()` metodi `Series` ning har bir qiymati nechta marta takrorlanganligini hisoblab chiqadi.

```
import pandas as pd
```

```
data = ['a', 'b', 'a', 'c', 'b', 'a']
```

```
series = pd.Series(data)
```

```
print(series.value_counts())
```

7. `unique()` metodi `Series` ning noyob qiymatlarini qaytaradi.

```
print(series.unique())
```

DataFrame` ning asosiy metodlari

1. `head()` metodi `DataFrame` ning dastlabki n ta qatorini qaytaradi. n ning standart qiymati qiymati 5 ga teng.

```
import pandas as pd
```

```
data = { 'Ism': ['Ali', 'Vali', 'Hasan', 'Husan', 'Olim'],
```

```
        'Yosh': [23, 35, 45, 25, 32],
```

```
        'Shahar': ['Toshkent', 'Samarqand', 'Buxoro', 'Andijon', 'Nukus'] }
```

```
df = pd.DataFrame(data)
```

```
print(df.head(3)) # Dastlabki 3 qatorni chiqarish
```

2. `tail()` metodi `DataFrame` ning oxirgi n qatorini qaytaradi.

```
print(df.tail(3)) # Oxirgi 3 qatorni chiqarish
```

3. `describe()` metodi `DataFrame` ning statistik tahlilini qaytaradi.

```
print(df.describe())
```

```

      Yosh
count  5.000000
mean  32.000000
std    8.774964
min   23.000000
25%   25.000000
50%   32.000000
75%   35.000000
max   45.000000

```

4. `info()` metodi `DataFrame` ning umumiy ma'lumotlarini (ustun nomlari, turli turlari, to'ldirilgan qatorlar soni va hokazo) qaytaradi.

```
print(df.info())
```

```

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 5 entries, 0 to 4
Data columns (total 3 columns):
#   Column  Non-Null Count  Dtype
---  ---
0   Ism      5 non-null      object
1   Yosh     5 non-null      int64
2   Shahar   5 non-null      object
dtypes: int64(1), object(2)
memory usage: 248.0+ bytes
None

```

5. `sum()` metodi `DataFrame` ning belgilangan ustunidagi qiymatlarning yig'indisini qaytaradi (agar ustunning barcha elementlari sonli qiymatlar bo'lsa).

```
print(df['Yosh'].sum()) # Natija: 160
```

6. `mean()` metodi `DataFrame` ning belgilangan ustunidagi qiymatlarning o'rtacha qiymatini qaytaradi (agar ustunning barcha elementlari sonli qiymatlar bo'lsa).

```
import pandas as pd
```

```
print(df['Yosh'].mean()) # Natija: 32.2
```

7. `groupby()` metodi `DataFrame` ni bir yoki bir nechta ustun bo'yicha guruhlash va agregatsiya qilish imkoniyatini beradi.

```
import pandas as pd
```

```
# Ma'lumotlar to'plamini yaratish
```

```
data = { 'Ism': ['Ali', 'Vali', 'Hasan', 'Husan', 'Olim', 'Salim'],
```

```
        'Yosh': [23, 35, 45, 25, 32, 40],
```

```
        'Shahar': ['Toshkent', 'Samarqand', 'Buxoro', 'Toshkent', 'Samarqand',
                  'Buxoro'],
```

```
        'Maosh': [5000, 6000, 7000, 5500, 6500, 7500] }
```

```
df = pd.DataFrame(data)
```

```
print("Dastlabki DataFrame:\n", df)
```

DataFrame ni 'Shahar' ustuni bo'yicha guruhlash va o'rtacha qiymatni hisoblash

```
grouped_df = df.groupby('Shahar')
```

```
print("\n'groupby' va 'mean' orqali guruhlangan DataFrame:\n", grouped_df)
```

DataFrame ni bir nechta ustun bo'yicha guruhlash

```
grouped_multiple_df = df.groupby(['Shahar', 'Ism']).mean()
```

```
print("\nBir nechta ustun bo'yicha guruhlangan DataFrame:\n", grouped_multiple_df)
```

```
Dastlabki DataFrame:
   Ism  Yosh  Shahar  Maosh
0  Ali   23  Toshkent  5000
1  Vali  35  Samarqand  6000
2  Hasan 45  Buxoro    7000
3  Husan 25  Toshkent  5500
4  Olim  32  Samarqand  6500
5  Salim 40  Buxoro    7500

'groupby' va 'mean' orqali guruhlangan DataFrame:
<pandas.core.groupby.generic.DataFrameGroupBy object at 0x7819e62abaf0>

Bir nechta ustun bo'yicha guruhlangan DataFrame:
      Shahar  Ism  Yosh  Maosh
Buxoro  Hasan  45.0  7000.0
        Salim  40.0  7500.0
Samarqand Olim  32.0  6500.0
         Vali  35.0  6000.0
Toshkent Ali   23.0  5000.0
         Husan 25.0  5500.0
```

8. `apply()` metodi `DataFrame` yoki `Series` ga foydalanuvchi tomonidan aniqlangan funktsiyani qo'llash imkonini beradi.

```
def double(x):
```

```
    return x * 2
```

```
print(df['Yosh'].apply(double))
```

9. `drop()` metodi `DataFrame` dan qator yoki ustunni o'chiradi.

```
df = df.drop('Shahar', axis=1) # Ustunni o'chirish
```

```
print(df)
```

```
   Ism  Yosh  Maosh
0  Ali   23  5000
1  Vali  35  6000
2  Hasan 45  7000
3  Husan 25  5500
4  Olim  32  6500
5  Salim 40  7500
```

10. `fillna()` metodi `NaN` qiymatlarni berilgan qiymat bilan to'ldiradi.

```
data = {
```

```
    'Ism': ['Ali', 'Vali', 'Hasan'],
```

```
'Yosh': [23, None, 45],
'Shahar': ['Toshkent', 'Samarqand', None] }
```

```
df = pd.DataFrame(data)
```

```
df = df.fillna({'Yosh': df['Yosh'].mean(), 'Shahar': 'Noma'lum'})
```

```
print(df)
```

	Ism	Yosh	Shahar
0	Ali	23.0	Toshkent
1	Vali	34.0	Samarqand
2	Hasan	45.0	Noma'lum

11. `read_csv()` – CSV faylidan ma'lumotlarni yuklash.

```
import pandas as pd
```

```
df = pd.read_csv('data.csv')
```

```
print(df.head())
```

	Xususiyat1	Xususiyat2	Maqsad
0	3.745401	0.314292	8.946329
1	9.507143	6.364104	43.324677
2	7.319939	3.143560	28.316673
3	5.986585	5.085707	28.794880
4	1.560186	9.075665	34.859009

12. `to_csv()` – DataFrame ni CSV fayliga saqlash.

```
df.to_csv('data.csv', index=False)
```

13. `value_counts()` – ma'lumotlar massivida ustun yoki qatorlarda bir qiymatli elementlar yoki elementlar qatorinining necha marta takrorlanganligini hisoblab chiqadi.

```
print(df.value_counts()) #Satr bo'ylab barcha ustundagi elementlarni solishtirish
```

Xususiyat1	Xususiyat2	Maqsad	
0.055221	5.120931	14.556981	1
6.011150	2.287982	19.942457	1
7.290072	1.743664	21.300683	1
7.132448	8.972158	46.479103	1
7.080726	0.769799	21.585381	1
			..
2.713490	6.323058	27.055683	1
2.587800	2.848405	13.246999	1
2.123391	8.081204	30.377651	1
1.996738	8.180148	29.002974	1
9.868869	5.908929	38.664431	1

Name: count, Length: 100, dtype: int64

14. `loc[]` – Etiketlar (indekslar va ustun nomlari) yordamida qator va ustunlarni tanlash.

```
print(df.loc[0]) # 1-qatorni tanlash
```

```
print(df.loc[:, 'column_name']) # Ustunni tanlash
```

```
print(df.loc[0, 'column_name']) # 1-qator va ustunni tanlash
```

15. `iloc[]` – Raqamli indekslar yordamida qator va ustunlarni tanlash.

```
print(df.iloc[0]) # 1-qatorni tanlash
```

```
print(df.iloc[:, 1]) # 1-ustunni tanlash
```

```
print(df.iloc[0, 1]) # 1-qator va 1-ustunni tanlash
```

16. `filter()` – Ma'lum bir shart bo'yicha qator va ustunlarni tanlash.

```
filtered_df = df[df['column_name'] > value]
```

```
print(filtered_df)
```

17. `drop()` – DataFrame dan qator yoki ustunni o'chirish.

```
df = df.drop('column_name', axis=1) # Ustunni o'chirish
```

```
df = df.drop([0, 1], axis=0) # Qatorlarni o'chirish
```

18. `rename()` – Ustun va qator nomlarini o'zgartirish.

```
df = df.rename(columns={'old_name': 'new_name'})
```

19. `replace()` – Ma'lum qiymatlarni boshqasi bilan almashtirish.

```
df['column_name'] = df['column_name'].replace('old_value', 'new_value')
```

20. `astype()` – Ustunning ma'lumotlar turini o'zgartirish.

```
df['column_name'] = df['column_name'].astype('int')
```

21. `agg()` – Bir nechta agregatsion funksiyalarni qo'llash.

Ma'lumotlar to'plamini yaratish

```
data = { 'Ism': ['Ali', 'Vali', 'Hasan', 'Husan', 'Olim', 'Salim'],
```

```
        'Yosh': [23, 35, 45, 25, 32, 40],
```

```
        'Shahar': ['Toshkent', 'Samarqand', 'Buxoro', 'Toshkent', 'Samarqand',
```

```
        'Buxoro'],
```

```
        'Maosh': [5000, 6000, 7000, 5500, 6500, 7500] }
```

```
df = pd.DataFrame(data)
```

```
print("Dastlabki DataFrame:\n", df)
```

DataFrame ni 'Shahar' ustuni bo'yicha guruhlash

```
grouped_df = df.groupby('Shahar')
```

```
print("\n'groupby' va 'mean' orqali guruhlangan DataFrame:\n", grouped_df)
```

DataFrame ni bir nechta ustun bo'yicha guruhlash

```
grouped_multiple_df = df.groupby(['Shahar', 'Ism']).mean()
```

```

print("\nBir nechta ustun bo'yicha guruhlangan DataFrame:\n",
grouped_multiple_df)

# DataFrame ni 'Shahar' ustuni bo'yicha guruhlash va bir nechta agregatsiya
funksiyalarini qo'llash

agg_df = df.groupby('Shahar').agg({'Yosh': 'mean', 'Maosh': 'sum'})

print("\nBir nechta agregatsiya funksiyalari qo'llangan DataFrame:\n", agg_df)

```

```

Dastlabki DataFrame:
   Ism  Yosh  Shahar  Maosh
0  Ali   23  Toshkent  5000
1  Vali  35  Samarqand  6000
2  Hasan 45   Buxoro  7000
3  Husan 25  Toshkent  5500
4  Olim  32  Samarqand  6500
5  Salim 40   Buxoro  7500

'groupby' va 'mean' orqali guruhlangan DataFrame:
<pandas.core.groupby.generic.DataFrameGroupBy object at 0x7819e7f7f2b0>

Bir nechta ustun bo'yicha guruhlangan DataFrame:
   Shahar  Ism  Yosh  Maosh
Buxoro   Hasan  45.0  7000.0
         Salim  40.0  7500.0
Samarqand Olim  32.0  6500.0
         Vali  35.0  6000.0
Toshkent  Ali   23.0  5000.0
         Husan 25.0  5500.0

Bir nechta agregatsiya funksiyalari qo'llangan DataFrame:
   Shahar  Yosh  Maosh
Buxoro   42.5  14500
Samarqand 33.5  12500
Toshkent  24.0  10500

```

21. ``merge()`` – Ikki yoki undan ortiq DataFrame larni birlashtirish (qo'shish).

```
merged_df = pd.merge(df1, df2, on='common_column')
```

```
print(merged_df)
```

22. ``concat()`` – DataFrame larning qator yoki ustunlarini birlashtirish.

```
concat_df = pd.concat([df1, df2], axis=0) # Qator bo'yicha birlashtirish
```

```
concat_df = pd.concat([df1, df2], axis=1) # Ustun bo'yicha birlashtirish
```

```
print(concat_df)
```

23. ``pivot_table()`` – DataFrame ning pivot jadvalini yaratish.

```
import pandas as pd
```

```
# Ma'lumotlar to'plamini yaratish
```

```

data = {
    'Ism': ['Ali', 'Vali', 'Hasan', 'Husan', 'Olim', 'Salim'],
    'Yosh': [23, 35, 45, 25, 32, 40],
    'Shahar': ['Toshkent', 'Samarqand', 'Buxoro', 'Toshkent', 'Samarqand',
'Buxoro'],
    'Maosh': [5000, 6000, 7000, 5500, 6500, 7500]
}

df = pd.DataFrame(data)

print("Dastlabki DataFrame:\n", df)

# Pivot jadval yaratish

pivot_df = df.pivot_table(values='Maosh', index='Shahar', columns='Ism',
aggfunc='mean')

print("\nPivot jadval:\n", pivot_df)

```

```

Dastlabki DataFrame:
   Ism  Yosh  Shahar  Maosh
0  Ali   23  Toshkent  5000
1  Vali   35  Samarqand  6000
2  Hasan  45   Buxoro  7000
3  Husan  25  Toshkent  5500
4  Olim   32  Samarqand  6500
5  Salim  40   Buxoro  7500

Pivot jadval:
Ism      Ali  Hasan  Husan  Olim  Salim  Vali
Shahar
Buxoro    NaN  7000.0   NaN   NaN  7500.0   NaN
Samarqand  NaN   NaN   NaN  6500.0   NaN  6000.0
Toshkent  5000.0   NaN  5500.0   NaN   NaN   NaN

```

Pandas kutubxonasi ma'lumotlarni tahlil qilish, tozalash, o'zgartirish va vizuallashtirish uchun keng ko'lamli metodlarni taqdim etadi. Ushbu metodlar yordamida ma'lumotlar bilan ishlash ancha oson va samarali bo'ladi. Pandas kutubxonasi ilmiy tadqiqotlar, ma'lumotlar tahlili va mashinani o'qitish loyihalarida keng qo'llaniladi va uning kuchli imkoniyatlari tahlilchilarga va dasturchilarga o'z ishlarini samarali bajarishga yordam beradi.

Flask — Python tilidagi yengil (micro) web-freymvork bo'lib, kichik va o'rta hajmdagi ta'limiy veb-ilovalarni tez yaratish uchun juda qulay. U “keragicha minimal” tamoyiliga asoslanadi: ya'ni dasturchi faqat zarur funksiyalarni qo'shadi, ortiqcha murakkablik bo'lmaydi. Shu sababli Flask ta'lim jarayonida ham o'qituvchilar, ham talaba-o'quvchilar uchun amaliy loyihalar yaratishda yaxshi tanlov

hisoblanadi. Quyida Flask asosida yengil ta'limiy veb-illovalar yaratishning mohiyati, nazariy asoslari, amaliy qo'llanish yo'nalishlari va misollar bilan yoritiladi.

Ta'limda veb-illovalardan foydalanishning asosiy maqsadi — o'quv jarayonini raqamlashtirish, o'quvchilar faolligini oshirish, topshiriqlarni qulay tarqatish va natijalarni tez tahlil qilishdir. Masalan, oddiy test platforma, davomatni belgilash sahifasi, uy vazifasi topshirish shakli, o'quv materiallarini joylashtirish, savol-javob (FAQ) yoki mini-LMS kabi tizimlar aynan yengil veb-ilovalar formatida juda tez ishga tushadi. Flask bu kabi loyihalarda server qismi (backend) vazifasini bajaradi: foydalanuvchi brauzer orqali so'rov yuboradi, Flask uni qabul qiladi, kerakli ishlov beradi va javob qaytaradi.

Flaskning nazariy asosini HTTP protokoli, “request-response” (so'rov-javob) modeli va marshrutlash (routing) tushunchalari tashkil etadi. Veb-ilovada har bir manzil (URL) ma'lum bir funksiyaga bog'lanadi. Masalan, / bosh sahifa, /quiz test sahifasi, /result natijalar sahifasi bo'lishi mumkin. Flaskda “route decorator” orqali URL va funksiyalar bog'lanadi. Shu yondashuv o'quvchilar uchun ham juda tushunarli: ular veb-dasturlashning amaliy mexanizmini tez anglab oladilar.

Flask ta'limda ayniqsa “tez prototiplash” (rapid prototyping) uchun qulay. O'qituvchi yoki talaba murakkab tizimni boshidan qurib o'tirmaydi, avval minimal ishlaydigan versiyani (MVP) yaratadi: masalan, 10 ta savolli test, natijani chiqarish, oddiy baholash. Keyin esa bosqichma-bosqich imkoniyatlarni kengaytiradi: foydalanuvchi ro'yxatdan o'tishi, natijalar bazaga yozilishi, statistik tahlil, admin sahifasi va hokazo. Bu uslub ta'lim metodikasiga ham mos: o'quvchi “kichikdan kattaga” tamoyili asosida loyiha yaratishni o'rganadi.

Yengil ta'limiy veb-illovalarning eng keng tarqalgan turlaridan biri — mini-test tizimidir. Bunda o'quvchi savollarga javob beradi, tizim esa avtomatik tekshiradi va natijani ko'rsatadi. Flask yordamida buni juda oddiy amalga oshirish mumkin. Masalan, savollarni Python lug'ati yoki JSON ko'rinishida saqlab, forma orqali javoblarni qabul qilish va baholash mumkin. Bu usul o'quv jarayonida baholashni tezlashtiradi, o'qituvchi vaqtini tejaydi, o'quvchi esa natijani darhol ko'rib, xatolarini tahlil qiladi.

Flaskning yana bir kuchli jihati — Jinja2 shablon tizimi (template engine). Bu orqali HTML sahifalarni dinamik yaratish mumkin: masalan, har bir o'quvchi uchun alohida natijalar, turli mavzu bo'yicha testlar, topshiriqlar ro'yxati, ball jadvali. Template yondashuvi o'quvchilarga frontend va backend aloqasini tushuntirishda juda qulay: Python tomonda ma'lumot tayyorlanadi, HTML tomonda esa chiroyli ko'rinishda chiqariladi. Masalan, o'quvchining ismi, to'plagan balli, to'g'ri va noto'g'ri javoblar soni sahifada avtomatik ko'rsatiladi.

Ta'limiy ilovalarda muhim masalalardan biri — ma'lumotlarni saqlash. Flaskning o'zi “micro” bo'lgani uchun, bazani o'zingiz tanlaysiz: oddiy loyihada

SQLite, kengaytirilgan loyihada PostgreSQL yoki MySQL. Odatda, Flask + SQLAlchemy kombinatsiyasi ishlatiladi. Bu orqali o'quvchilar "ma'lumotlar bazasi + web" integratsiyasini amalda o'rganadilar: o'quvchi login qiladi, test topshiradi, natija bazaga yoziladi, o'qituvchi esa admin sahifada ko'radi. Bu real hayotdagi LMS va ERP tizimlariga o'xshash tajriba beradi.

Flask orqali ta'lim jarayonini optimallashtirishning yana bir yo'li — davomat va monitoring tizimlari yaratishdir. Masalan, har bir darsda o'quvchilar telefon yoki kompyuter orqali QR kod orqali sahifaga kirib, o'zini belgilaydi. Tizim vaqtni qayd etadi va Excel ko'rinishida hisobot chiqaradi. Bu o'qituvchining qog'ozbozligini kamaytiradi va aniq statistika beradi. Bunday loyihalarda Flask foydalanuvchi formasidan ma'lumot olib, bazaga yozish va natijani ko'rsatish vazifasini bajaradi.

Flask asosida yaratiladigan yana bir yengil, lekin foydali ta'limiy loyiha — "uy vazifasi topshirish" sahifasidir. O'quvchi topshiriq matnini yoki faylini yuklaydi, tizim esa ularni tartib bilan saqlaydi. O'qituvchi esa ro'yxatdan ko'rib, baho qo'yadi. Agar bunga oddiy autentifikatsiya (login/parol) qo'shilsa, tizim ancha tartibli bo'ladi. Bu turdagi ilovalar ayniqsa kichik kurslar, to'garaklar yoki bitta fan doirasida juda qulay.

Ta'limiy veb-ilovalar yaratishda xavfsizlik va etik masalalar ham nazariy jihatdan muhim. Masalan, o'quvchilar ma'lumotlarini himoyalash, parolni ochiq saqlamaslik, formalarni CSRF hujumlaridan himoya qilish, ruxsatsiz kirishni cheklash, test savollarini oshkor bo'lishini oldini olish. Flask ekotizimida bu masalalar uchun tayyor yechimlar bor: Flask-Login, Flask-WTF kabi kengaytmalar yordamida autentifikatsiya va forma xavfsizligini kuchaytirish mumkin. Ta'lim jarayonida bunday masalalarni ko'tarish o'quvchilarda raqamli madaniyat va kiberxavfsizlik tushunchalarini ham shakllantiradi.

Flaskning ta'limdagi yana bir katta afzalligi — u API (masalan, JSON qaytaruvchi servis) yaratish uchun ham juda qulay. Bu degani, siz Flaskda "backend" yozib, mobil ilova yoki Telegram bot bilan bog'lashingiz mumkin. Masalan, o'quvchi botga sinfini yuboradi, bot Flask API'dan jadvalni olib qaytaradi. Bu siz ilgari qilgan "jadval Excel'dan olinadi" kabi g'oyalar bilan juda mos. Shunday integratsiya ta'lim jarayonini yanada avtomatlashtiradi va o'quvchilarni zamonaviy IT yechimlarga yaqinlashtiradi.

Quyida **Flask asosida bitta yengil ta'limiy loyihaga** misol keltiramiz: **"Mini Quiz (Test) ilovasi"**.

Loyiha imkoniyatlari

- / — bosh sahifa (testga kirish)
- /quiz — savollar sahifasi (radio button)
- /result — natijani chiqarish (ball, to'g'ri/noto'g'ri)
- Savollar vaqtincha Python ichida saqlanadi (keyin DB/JSONga oson ko'chadi)

Papka tuzilishi

```
miniquiz/  
  app.py  
  templates/  
    index.html  
    quiz.html  
    result.html
```

1) app.py

```
from flask import Flask, render_template, request  
app = Flask(__name__)  
# Savollar bazasi (oddiy ko'rinishda)  
QUESTIONS = [  
    {  
        "id": 1,  
        "question": "Python tilida massivga o'xshash  
tuzilma qaysi?",  
        "options": ["list", "int", "float", "bool"],  
        "answer": "list"  
    },  
    {  
        "id": 2,  
        "question": "NumPy kutubxonasining asosiy obyekt  
turi nima?",  
        "options": ["ndarray", "DataFrame", "Series",  
"dict"],  
        "answer": "ndarray"  
    },  
    {  
        "id": 3,  
        "question": "Flask odatda qaysi turdagi  
framework?",  
        "options": ["micro", "desktop", "game",  
"mobile"],  
        "answer": "micro"  
    }  
]  
  
@app.route("/")  
def index():  
    return render_template("index.html")  
  
@app.route("/quiz", methods=["GET", "POST"])  
def quiz():  
    if request.method == "GET":  
        return render_template("quiz.html",  
questions=QUESTIONS)
```

```

# POST: javoblarni tekshiramiz
correct = 0
total = len(QUESTIONS)
details = []

for q in QUESTIONS:
    user_answer = request.form.get(f"q{q['id']}")
    is_correct = (user_answer == q["answer"])
    if is_correct:
        correct += 1

    details.append({
        "question": q["question"],
        "your_answer": user_answer,
        "correct_answer": q["answer"],
        "is_correct": is_correct
    })

score_percent = round((correct / total) * 100, 2)

return render_template(
    "result.html",
    correct=correct,
    total=total,
    score_percent=score_percent,
    details=details
)

```

```

if __name__ == "__main__":
    app.run(debug=True)

```

2) templates/index.html

```

<!DOCTYPE html>
<html lang="uz">
<head>
  <meta charset="UTF-8" />
  <title>MiniQuiz</title>
</head>
<body>
  <h2>MiniQuiz – Flask asosida test</h2>
  <p>Testni boshlash uchun quyidagi tugmani bosing.</p>
  <a href="/quiz">✔ Testni boshlash</a>
</body>
</html>

```

3) templates/quiz.html

```

<!DOCTYPE html>
<html lang="uz">
<head>
  <meta charset="UTF-8" />
  <title>Test</title>
</head>
<body>
  <h2>Test savollari</h2>

  <form method="POST">
    {% for q in questions %}
      <div style="margin-bottom: 16px;">
        <p><b>{{ loop.index }}. {{ q.question }}</b></p>

        {% for opt in q.options %}
          <label>
            <input type="radio" name="q{{ q.id }}"
value="{{ opt }}" required>
            {{ opt }}
          </label><br>
        {% endfor %}
      </div>
      <hr>
    {% endfor %}

    <button type="submit">Natijani ko`rish</button>
  </form>
</body>
</html>

```

4) templates/result.html

```

<!DOCTYPE html>
<html lang="uz">
<head>
  <meta charset="UTF-8" />
  <title>Natija</title>
</head>
<body>
  <h2>Natija</h2>

  <p><b>To`g`ri:</b> {{ correct }} / {{ total }}</p>
  <p><b>Foiz:</b> {{ score_percent }}%</p>

  <h3>Tafsilotlar</h3>
  {% for d in details %}
    <div style="margin-bottom: 14px;">

```

```

<p><b>Savol:</b> {{ d.question }}</p>
<p><b>Sizning javob:</b> {{ d.your_answer }}</p>
<p><b>To'g'ri javob:</b> {{ d.correct_answer }}</p>
<p>
  <b>Holat:</b>
  {% if d.is_correct %}
    ✓ To'g'ri
  {% else %}
    ✗ Noto'g'ri
  {% endif %}
</p>
</div>
<hr>
{% endfor %}

<a href="/quiz">🔄 Qayta ishlash</a>
</body>
</html>

```

Ishga tushirish tartibi

1. Papkani yarating: miniquiz/
2. Ichiga app.py va templates/ papkasini qo'ying
3. Terminalda:

```

pip install flask
python app.py

```
4. Brauzerda quyidagi havolani oching: <http://127.0.0.1:5000/>

Zamonaviy ta'lim tizimida raqamli texnologiyalarning keng joriy etilishi ta'lim jarayonini boshqarish, tashkil etish va baholash usullarini tubdan o'zgartirmoqda. An'anaviy auditoriya va qog'ozga asoslangan ta'lim modellaridan farqli ravishda, raqamli platformalar o'quvchilar faolligini oshirish, o'qituvchi faoliyatini optimallashtirish hamda ta'lim sifatini oshirish imkonini beradi. Shu jarayonda **kompleks ta'lim platformalari (Learning Management System – LMS)** alohida ahamiyat kasb etadi.

Python dasturlash tilining kuchli va ommabop freymvorklaridan biri bo'lgan **Django** murakkab, kengaytiriladigan va xavfsiz ta'lim platformalarini ishlab chiqish uchun samarali vosita hisoblanadi. Ushbu bobda Django freymvorki asosida kompleks ta'lim platformalarini yaratishning nazariy va amaliy asoslari yoritiladi.

Kompleks ta'lim platformasi — bu o'qituvchi, talaba va ta'lim muassasasi ehtiyojlarini qamrab oluvchi, o'quv jarayonining barcha bosqichlarini raqamli muhitda boshqarishga xizmat qiluvchi axborot tizimidir. Bunday platformalar quyidagi funksiyalarni o'z ichiga oladi:

foydalanuvchilarni boshqarish;

kurslar va fanlarni tashkil etish;
o‘quv materiallarini joylashtirish;
topshiriqlar va testlar orqali baholash;
natijalarni monitoring qilish va tahlil qilish;
aloqa va bildirishnomalarni ta’minlash.

Django asosida yaratilgan platformalar yuqoridagi vazifalarni yagona tizim doirasida amalga oshirish imkonini beradi.

Django — bu “batteries included” tamoyiliga asoslangan to‘liq funksional web-freymvork bo‘lib, u murakkab tizimlarni tez va sifatli ishlab chiqish imkonini beradi. Ta’lim platformalari uchun Django quyidagi afzalliklarga ega:

Birinchidan, Django’da foydalanuvchilarni autentifikatsiya va avtorizatsiya qilish tizimi tayyor holatda mavjud. Bu ta’lim platformalarida rollar (administrator, o‘qituvchi, talaba) asosida kirish huquqlarini belgilashni osonlashtiradi.

Ikkinchidan, Django’ning obyektga yo‘naltirilgan ma’lumotlar modeli (ORM) ma’lumotlar bazasi bilan ishlashni soddalashtiradi. Kurslar, darslar, topshiriqlar, natijalar kabi ma’lumotlar tuzilmali va izchil tarzda saqlanadi.

Uchinchidan, Django admin paneli o‘qituvchi va administratorlar uchun qulay boshqaruv muhiti yaratadi. O‘quv materiallarini qo‘shish, talabalarni ro‘yxatdan o‘tkazish, baholarni tahrirlash kabi jarayonlar kod yozmasdan amalga oshiriladi.

Kompleks ta’lim platformasini ishlab chiqishda modulga asoslangan arxitektura muhim ahamiyatga ega. Django loyihasi odatda bir nechta mustaqil ilovalarga (apps) bo‘linadi. Har bir ilova platformaning ma’lum funksional qismini bajaradi.

Asosiy modullar quyidagilardan iborat bo‘lishi mumkin:

- foydalanuvchilar va rollar moduli;
- kurslar va fanlar moduli;
- o‘quv kontenti moduli;
- topshiriqlar va baholash moduli;
- test va quiz moduli;
- monitoring va analitika moduli;
- bildirishnoma va aloqa moduli.

Bunday yondashuv platformani keyinchalik kengaytirish, texnik xizmat ko‘rsatish va jamoaviy ishlab chiqishni yengillashtiradi.

Ta’lim platformasining asosiy subyektlari — bu foydalanuvchilardir. Django asosida foydalanuvchilar bilan ishlashda rollarga asoslangan yondashuv qo‘llaniladi. Administrator tizimni umumiy boshqaradi, o‘qituvchi kurslar va baholashni yuritadi, talaba esa o‘quv materiallari bilan ishlaydi va topshiriqlarni bajaradi.

Bu model pedagogik jarayonda mas'uliyat va vakolatlarni aniq ajratishga yordam beradi. Talaba faqat o'z kurslari va natijalarini ko'ra oladi, o'qituvchi esa faqat o'z fanlari doirasida boshqaruvni amalga oshiradi.

Django asosidagi ta'lim platformalarida kurslar ierarxik tuzilishda tashkil etiladi. Har bir kurs bo'limlarga, bo'limlar esa alohida darslarga bo'linadi. Dars tarkibida matnli tushuntirish, video materiallar, fayllar va tashqi havolalar bo'lishi mumkin.

Bunday tuzilma didaktik jihatdan qulay bo'lib, o'quvchilarga materialni ketma-ket va tizimli o'zlashtirish imkonini beradi. O'qituvchi esa kursni istalgan vaqtda yangilashi yoki kengaytirishi mumkin.

Kompleks ta'lim platformalarining muhim qismi — bu baholash mexanizmlaridir. Django asosida topshiriqlarni yaratish, ularni belgilangan muddatda qabul qilish va baholash avtomatlashtiriladi. Talabalar o'z ishlanmalarini matn yoki fayl ko'rinishida topshiradilar, o'qituvchi esa baho va izoh qoldiradi.

Test va quiz modullari esa bilimlarni tezkor va obyektiv baholash imkonini beradi. Savollar banki, vaqt cheklovi, bir nechta urinishlar kabi mexanizmlar ta'lim sifatini oshiradi va akademik halollikni ta'minlashga xizmat qiladi.

Django asosidagi platformalarda barcha ta'limiy jarayonlar raqamli iz qoldiradi. Bu esa o'quvchilar faolligi, o'zlashtirish darajasi va rivojlanish dinamikasini tahlil qilish imkonini beradi. Analitik modul orqali o'qituvchi qaysi mavzular qiyinligini, qaysi talabalar qo'shimcha yordamga muhtojligini aniqlay oladi.

Bunday yondashuv ta'lim jarayonini ma'lumotlarga asoslangan holda boshqarish imkonini yaratadi va ta'limni optimallashtirishga xizmat qiladi.

Ta'lim platformalarida axborot xavfsizligi va akademik halollik masalalari alohida e'tibor talab qiladi. Django freymvorki CSRF va XSS hujumlariga qarshi himoya mexanizmlariga ega bo'lib, foydalanuvchi ma'lumotlarini ishonchli saqlashni ta'minlaydi.

Shuningdek, test savollarini tasodifiylashtirish, vaqt cheklovlari va urinishlar sonini belgilash orqali baholash jarayonining adolatliligi oshiriladi.

Mini LMS (Django) loyihasini ishlab chiqishni ko'rib chiqamiz:

Loyiha g'oyasi quyidagicha

O'qituvchi kurs yaratadi, dars qo'shadi, topshiriq beradi. Talaba topshiriqni matn ko'rinishida topshiradi. Tizim topshirilgan ishlarni ro'yxatda ko'rsatadi.

Funksiyalar (minimal)

- Kurslar ro'yxati
- Kurs ichida darslar ro'yxati
- Darsga topshiriq biriktirish
- Talaba topshiriqni topshirishi (submission)
- O'qituvchi submissions ro'yxatini ko'rishi

1) Loyihani yaratish

Terminal:

```
pip install django
django-admin startproject eduplatform
cd eduplatform
python manage.py startapp lms
eduplatform/settings.py ichida:
```

```
INSTALLED_APPS = [
    # ...
    'lms',
]
```

2) Model (lms/models.py)

```
from django.db import models
```

```
class Course(models.Model):
```

```
    title = models.CharField(max_length=200)
```

```
    description = models.TextField(blank=True)
```

```
    def __str__(self):
```

```
        return self.title
```

```
class Lesson(models.Model):
```

```
    course = models.ForeignKey(Course,
on_delete=models.CASCADE, related_name="lessons")
```

```
    title = models.CharField(max_length=200)
```

```
    content = models.TextField()
```

```
    def __str__(self):
```

```
        return f"{self.course.title} - {self.title}"
```

```
class Assignment(models.Model):
```

```
    lesson = models.OneToOneField(Lesson,
on_delete=models.CASCADE, related_name="assignment")
```

```
    title = models.CharField(max_length=200)
```

```
    description = models.TextField()
```

```
    deadline = models.DateField(null=True,
blank=True)
```

```
    def __str__(self):
```

```
return self.title
```

```
class Submission(models.Model):  
    assignment = models.ForeignKey(Assignment,  
on_delete=models.CASCADE, related_name="submissions")  
    student_name = models.CharField(max_length=120)  
    answer_text = models.TextField()  
    submitted_at =  
models.DateTimeField(auto_now_add=True)
```

```
def __str__(self):  
    return f"{self.student_name} -  
{self.assignment.title}"
```

Migratsiya:

```
python manage.py makemigrations  
python manage.py migrate
```

3) Admin panel (lms/admin.py)

```
from django.contrib import admin  
from .models import Course, Lesson, Assignment,  
Submission
```

```
admin.site.register(Course)  
admin.site.register(Lesson)  
admin.site.register(Assignment)  
admin.site.register(Submission)
```

Admin yaratish:

```
python manage.py createsuperuser
```

4) View'lar (lms/views.py)

```
from django.shortcuts import render,  
get_object_or_404, redirect  
from .models import Course, Lesson, Assignment,  
Submission
```

```
def course_list(request):  
    courses = Course.objects.all()  
    return render(request, "lms/course_list.html",  
{"courses": courses})
```

```

def lesson_list(request, course_id):
    course = get_object_or_404(Course, id=course_id)
    return render(request, "lms/lesson_list.html",
{"course": course})

def lesson_detail(request, lesson_id):
    lesson = get_object_or_404(Lesson, id=lesson_id)
    assignment = getattr(lesson, "assignment", None)
    return render(request, "lms/lesson_detail.html",
{"lesson": lesson, "assignment": assignment})

def submit_assignment(request, assignment_id):
    assignment = get_object_or_404(Assignment,
id=assignment_id)

    if request.method == "POST":
        student_name =
request.POST.get("student_name")
        answer_text = request.POST.get("answer_text")

        Submission.objects.create(
            assignment=assignment,
            student_name=student_name,
            answer_text=answer_text
        )
        return redirect("submissions",
assignment_id=assignment.id)

    return render(request, "lms/submit.html",
{"assignment": assignment})

def submissions(request, assignment_id):
    assignment = get_object_or_404(Assignment,
id=assignment_id)
    subs = assignment.submissions.order_by("-
submitted_at")
    return render(request, "lms/submissions.html",
{"assignment": assignment, "subs": subs})

```

5) URL'lar

eduplatform/urls.py:

```
from django.contrib import admin
from django.urls import path, include
```

```
urlpatterns = [
    path('admin/', admin.site.urls),
    path('', include('lms.urls')),
]
```

lms/urls.py:

```
from django.urls import path
from . import views
```

```
urlpatterns = [
    path('', views.course_list, name="course_list"),
    path('course/<int:course_id>/',
views.lesson_list, name="lesson_list"),
    path('lesson/<int:lesson_id>/',
views.lesson_detail, name="lesson_detail"),
    path('assignment/<int:assignment_id>/submit/',
views.submit_assignment, name="submit_assignment"),

path('assignment/<int:assignment_id>/submissions/',
views.submissions, name="submissions"),
]
```

6) Template'lar (minimal)

lms/templates/lms/course_list.html

```
<h2>Kurslar</h2>
```

```
<ul>
```

```
    {% for c in courses %}
```

```
        <li><a href="/course/{{ c.id }}/">{{ c.title
}}</a></li>
```

```
    {% endfor %}
```

```
</ul>
```

lms/templates/lms/lesson_list.html

```
<h2>{{ course.title }} - Darslar</h2>
```

```
<ul>
```

```

        {% for l in course.lessons.all %}
            <li><a href="/lesson/{{ l.id }}/">{{ l.title
}}</a></li>
        {% endfor %}
    </ul>

    <a href="/">← Kurslarga qaytish</a>
    lms/templates/lms/lesson_detail.html
    <h2>{{ lesson.title }}</h2>
    <p>{{ lesson.content }}</p>

    {% if assignment %}
        <h3>Topshiriq: {{ assignment.title }}</h3>
        <p>{{ assignment.description }}</p>
        <a href="/assignment/{{ assignment.id
}}/submit/">✓ Topshirish</a> |
        <a href="/assignment/{{ assignment.id
}}/submissions/">■ Topshirilganlar</a>
    {% else %}
        <p><i>Bu dars uchun topshiriq
biriktirilmagan.</i></p>
    {% endif %}
    lms/templates/lms/submit.html
    <h2>Topshiriqni topshirish: {{ assignment.title
}}</h2>

    <form method="POST">
        {% csrf_token %}
        <label>Ism:</label><br>
        <input type="text" name="student_name"
required><br><br>

        <label>Javob:</label><br>
        <textarea name="answer_text" rows="6"
required></textarea><br><br>

        <button type="submit">Yuborish</button>
    </form>
    lms/templates/lms/submissions.html
    <h2>Topshirilganlar: {{ assignment.title }}</h2>

```

```

<ul>
  {% for s in subs %}
    <li>
      <b>{{ s.student_name }}</b> ({{ s.submitted_at
}})<br>
      {{ s.answer_text }}
    </li>
  <hr>
  {% empty %}
    <li>Hali topshirilgan ish yo`q.</li>
  {% endfor %}
</ul>

<a href="/lesson/{{ assignment.lesson.id }}/">←
Darsga qaytish</a>

```

7) Ishga tushirish

```
python manage.py runserver
```

Brauzer: <http://127.0.0.1:8000/>

Admin: <http://127.0.0.1:8000/admin/> (kurs, dars, topshiriqni admin'dan qo'shasiz)

Ushbu kichik loyiha Django freymvorkining ta'limiy platformalar yaratishdagi asosiy imkoniyatlarini ko'rsatadi: ma'lumotlar modeli (Course-Lesson-Assignment-Submission), admin panel orqali kontent boshqarish, URL-View-Template bog'lanishi hamda foydalanuvchi bilan interaktiv ishlash. Loyiha keyingi bosqichlarda autentifikatsiya (talaba login), fayl yuklash, baholash, test moduli va analitik hisobotlar bilan kengaytirilishi mumkin.

Raqamli texnologiyalar XXI asrda jamiyatning barcha sohalariga, jumladan ta'lim tizimiga ham chuqur kirib bormoqda. Onlayn platformalar, elektron darsliklar, masofaviy ta'lim, sun'iy intellekt, ta'limiy analitika kabi raqamli yechimlar ta'lim jarayonini samarali, moslashuvchan va shaffof qilish imkonini beradi. Biroq raqamli yechimlarni ta'lim jarayoniga joriy etish faqat texnologik masala emas, balki pedagogik, tashkiliy, psixologik va ijtimoiy muammolar bilan chambarchas bog'liq murakkab jarayondir. Ushbu jarayonda yuzaga keladigan muammolarni chuqur tahlil qilish va ularni hal etish yo'llarini ishlab chiqish ta'lim sifatini oshirishda muhim ahamiyat kasb etadi.

Ta'limda raqamli yechimlarni joriy etishdagi asosiy muammolardan biri — texnik infratuzilmaning yetarli darajada rivojlanmaganligidir. Ko'plab ta'lim muassasalarida kompyuterlar sonining kamligi, internet tezligining pastligi yoki

umuman mavjud emasligi raqamli ta'limni samarali tashkil etishga to'sqinlik qiladi. Ayniqsa, qishloq hududlarida bu muammo yanada yaqqol namoyon bo'ladi.

Mazkur muammoni hal etish uchun davlat va xususiy sektor hamkorligida ta'lim muassasalarini zamonaviy texnika bilan ta'minlash, keng polosali internet tarmoqlarini joriy etish zarur. Shuningdek, mobil texnologiyalar va oflayn ishlaydigan raqamli resurslardan foydalanish ham muhim yechimlardan biri hisoblanadi.

Raqamli texnologiyalarni ta'limga joriy etishda o'qituvchining roli hal qiluvchi ahamiyatga ega. Ammo ko'plab pedagoglar raqamli vositalardan samarali foydalanish bo'yicha yetarli bilim va ko'nikmaga ega emas. Natijada raqamli texnologiyalar faqat prezentatsiya ko'rsatish yoki tayyor elektron darsliklardan foydalanish bilan cheklanib qoladi.

Bu muammoni hal etish uchun pedagoglar uchun muntazam malaka oshirish kurslari, seminarlar va treninglar tashkil etilishi lozim. Ayniqsa, DigCompEdu kabi xalqaro raqamli kompetensiya modellariga asoslangan tayyorlov dasturlarini joriy etish samarali natija beradi. O'qituvchi raqamli vositalarni faqat texnik jihatdan emas, balki metodik va pedagogik maqsadlarda qo'llay olishi kerak.

Raqamli texnologiyalarni an'anaviy ta'lim metodlariga moslashtirmasdan joriy etish ko'pincha kutilgan natijani bermaydi. Agar raqamli vositalar faqat eski metodlarni elektron shaklga o'tkazish bilan cheklansa, ta'lim sifati oshmaydi. Bu holat pedagogik yondashuvlarning eskirganligi bilan bog'liq muammolarni yuzaga keltiradi.

Ushbu muammoni hal etish uchun ta'lim jarayonida konstruktivistik, shaxsga yo'naltirilgan va kompetensiyaviy yondashuvlarni qo'llash zarur. Raqamli texnologiyalar o'quvchini passiv tinglovchidan faol ishtirokchiga aylantirishi lozim. Masalan, loyiha asosida o'qitish, kollaborativ o'rganish va adaptiv ta'lim modellari raqamli muhitda samarali ishlaydi.

Ta'limda raqamli yechimlarni joriy etishda yana bir muhim muammo — raqamli tengsizlikdir. Barcha talabalar bir xil texnik imkoniyatlarga ega emas: ayrimlarida shaxsiy kompyuter yoki planshet bo'lsa, boshqalarida bunday imkoniyat mavjud emas. Bu holat ta'lim jarayonida adolatsizlikni keltirib chiqarishi mumkin.

Mazkur muammoni bartaraf etish uchun ta'lim muassasalari tomonidan umumiy foydalanish uchun kompyuter sinflarini rivojlantirish, ijtimoiy himoyaga muhtoj talabalarni texnik vositalar bilan ta'minlash muhimdir. Bundan tashqari, mobil qurilmalarga moslashtirilgan platformalar yaratish ham raqamli tengsizlikni kamaytiradi.

Raqamli ta'lim muhitida o'quvchilarning o'z-o'zini nazorat qilishi va motivatsiyasi muhim rol o'ynaydi. Biroq masofaviy yoki raqamli ta'limda ayrim

talabalar o'qishga bo'lgan qiziqishini yo'qotishi, diqqatni jamlay olmasligi mumkin. Bu esa ta'lim samaradorligini pasaytiradi.

Ushbu muammoni hal etish uchun gamifikatsiya, interaktiv topshiriqlar, vizual va multimediya resurslardan foydalanish tavsiya etiladi. Shuningdek, o'qituvchi tomonidan muntazam fikr-mulohaza (feedback) berilishi va talabalar bilan muloqotning kuchaytirilishi motivatsiyani oshiradi.

Raqamli ta'lim muhitida talabalar va pedagoglarning shaxsiy ma'lumotlari bilan ishlash masalasi alohida e'tibor talab qiladi. Ma'lumotlarning ruxsatsiz tarqalishi, kiberhujumlar va akademik halollikning buzilishi jiddiy muammolarni keltirib chiqarishi mumkin.

Bu muammoni hal etish uchun xavfsiz platformalardan foydalanish, parollarni himoyalash, autentifikatsiya va avtorizatsiya mexanizmlarini joriy etish zarur. Shuningdek, talabalar va pedagoglarda raqamli etika va kiberxavfsizlik bo'yicha bilimlarni shakllantirish muhim hisoblanadi.

Raqamli ta'limda baholash jarayonini to'g'ri tashkil etmaslik o'quvchilarning real bilim darajasini aniqlashda xatolarga olib kelishi mumkin. Avtomatik testlar bilimni yuzaki baholashi, plagiat va ko'chirish holatlari esa natijalarning ishonchliligini pasaytiradi.

Bu muammoni hal etish uchun baholashning turli shakllarini birlashtirish lozim: testlar, loyiha ishlari, portfoliolar, og'zaki suhbatlar va reflektiv topshiriqlar. Raqamli analitika vositalari yordamida o'quvchilar faoliyatini kompleks tahlil qilish mumkin.

Ta'limda raqamli yechimlarni joriy etishda me'yoriy-huquqiy baza va tashkiliy mexanizmlarning yetarli darajada shakllanmaganligi ham muammolarga sabab bo'ladi. Raqamli ta'lim natijalarini tan olish, masofaviy ta'limni akkreditatsiya qilish kabi masalalar aniq tartibga solinishi lozim.

Bu yo'nalishda davlat siyosati darajasida aniq strategiyalar ishlab chiqish, raqamli ta'limni qo'llab-quvvatlovchi normativ hujjatlarni takomillashtirish muhimdir.

Qisqa qilib aytganda, ta'limda raqamli yechimlarni joriy etish ko'plab imkoniyatlar bilan bir qatorda murakkab muammolarni ham yuzaga keltiradi. Texnik infratuzilma, pedagogik kompetensiya, raqamli tengsizlik, motivatsiya, xavfsizlik va baholash bilan bog'liq muammolarni tizimli va kompleks yondashuv orqali hal etish mumkin. Raqamli texnologiyalarni pedagogik maqsadlarga mos holda joriy etish, o'qituvchi va o'quvchilarni tayyorlash hamda ta'lim jarayonini ma'lumotlarga asoslangan holda boshqarish raqamli ta'limning barqaror rivojlanishini ta'minlaydi. Shu yo'l bilan ta'lim sifati oshadi va zamonaviy jamiyat talablariga mos, raqobatbardosh kadrlar tayyorlanadi.

2-MAVZU: Kiberxavfsizlik asoslari, shaxsiy ma'lumotlarni himoyalash va xavf-xatarlarning oldini olish choralari

Kiberxavfsizlikning asosiy tushunchalari va tamoyillari. Shaxsiy ma'lumotlar va ularni himoyalash asoslari. Kiberxavflar va tahdidlar turlari. Shaxsiy va korporativ ma'lumotlarni himoyalash usullari Xavf-xatarlarning oldini olish choralari. Ta'lim muassasalari va tashkilotlarda kiberxavfsizlik. Zamonaviy texnologiyalar va kiberxavfsizlik.

Axborot texnologiyalarining jadal rivojlanishi natijasida jamiyatning barcha sohalari, jumladan ta'lim, iqtisodiyot, davlat boshqaruvi va kundalik hayot raqamli muhit bilan chambarchas bog'lanib bormoqda. Shu bilan birga, axborot resurslari va axborot tizimlariga bo'lgan tahdidlar soni ham ortib bormoqda. Ana shunday sharoitda kiberxavfsizlik tushunchasi muhim ilmiy va amaliy ahamiyat kasb etadi.

Kiberxavfsizlik — bu axborot tizimlari, kompyuter tarmoqlari, dasturiy ta'minot va foydalanuvchi ma'lumotlarini ruxsatsiz kirish, buzish, o'zgartirish yoki yo'q qilishdan himoya qilishga qaratilgan choralar, texnologiyalar va tashkiliy tamoyillar majmuasidir. Kiberxavfsizlikning asosiy maqsadi axborotning ishonchliligi, yaxlitligi va mavjudligini ta'minlashdan iborat.

Kiberxavfsizlikning asosiy tushunchalaridan biri — axborot xavfsizligi bo'lib, u kiberxavfsizlikning kengroq tushunchasidir. Axborot xavfsizligi nafaqat raqamli muhitni, balki qog'oz shaklidagi ma'lumotlar, tashkiliy jarayonlar va inson omilini ham qamrab oladi. Kiberxavfsizlik esa asosan raqamli va tarmoq muhitiga yo'naltirilgan.

Kiberxavfsizlikda muhim tushunchalardan yana biri — tahdid hisoblanadi. Tahdid deganda axborot tizimiga zarar yetkazishi mumkin bo'lgan har qanday holat, hodisa yoki harakat tushuniladi. Tahdidlar ichki va tashqi bo'lishi mumkin. Ichki tahdidlar xodimlarning ehtiyotsizligi yoki qasddan qilgan harakatlari bilan bog'liq bo'lsa, tashqi tahdidlar xakerlar, zararli dasturlar yoki kiberjinoyatchilar faoliyati natijasida yuzaga keladi.

Tahdidlar bilan chambarchas bog'liq tushuncha — zaiflikdir. Zaiflik bu tizimdagi kamchilik yoki nuqson bo'lib, u orqali tahdid amalga oshirilishi mumkin. Masalan, eskirgan dasturiy ta'minot, kuchsiz parollar yoki noto'g'ri sozlangan serverlar tizimning zaif tomonlari hisoblanadi. Kiberxavfsizlikda zaifliklarni aniqlash va ularni bartaraf etish muhim vazifalardan biridir.

Kiberxavfsizlikning nazariy asosi sifatida ko'pincha CIA triadasi (Confidentiality, Integrity, Availability) ko'rsatib o'tiladi. Ushbu tamoyillar kiberxavfsizlikning asosiy ustunlari hisoblanadi. Maxfiylik (Confidentiality) axborotga faqat ruxsat etilgan shaxslar kirishini ta'minlashni anglatadi. Yaxlitlik (Integrity) axborotning ruxsatsiz o'zgartirilmasligi va uning aniqligi saqlanishini

bildiradi. Mavjudlik (Availability) esa axborot va tizimlar zarur bo'lgan paytda foydalanuvchilar uchun ochiq bo'lishini ta'minlashga qaratilgan.

Kiberxavfsizlikning muhim tamoyillaridan biri — avtorizatsiya va autentifikatsiya tamoyilidir. Autentifikatsiya foydalanuvchining kimligini aniqlash jarayoni bo'lsa, avtorizatsiya unga qaysi resurslardan foydalanishga ruxsat berilishini belgilaydi. Kuchli parollar, ikki bosqichli autentifikatsiya va biometrik texnologiyalar ushbu tamoyilni amalga oshirishga xizmat qiladi.

Yana bir muhim tamoyil — minimal huquqlar prinsipidir. Ushbu prinsipga ko'ra, foydalanuvchi yoki dastur faqat o'z vazifasini bajarish uchun zarur bo'lgan minimal huquqlarga ega bo'lishi kerak. Bu yondashuv xatolik yoki hujum yuz berganda zarar miqyosini kamaytiradi.

Kiberxavfsizlikda doimiy monitoring va yangilash tamoyili ham muhim ahamiyatga ega. Texnologiyalar va hujum usullari tez rivojlanib borayotganligi sababli, xavfsizlik choralari bir marta joriy etish yetarli emas. Tizimlarni muntazam yangilab borish, xavfsizlik jurnallarini tahlil qilish va potentsial tahdidlarni oldindan aniqlash zarur.

Inson omili kiberxavfsizlikda alohida o'rin tutadi. Ko'plab kiberhujumlar texnik kamchiliklardan emas, balki foydalanuvchilarning ehtiyotsizligi natijasida amalga oshiriladi. Shuning uchun kiberxavfsizlik madaniyatini shakllantirish, foydalanuvchilarni phishing, zararli havolalar va ijtimoiy muhandislik usullari haqida xabardor qilish muhim tamoyillardan biri hisoblanadi.

Kiberxavfsizlikda yana bir asosiy yondashuv — risklarni boshqarishdir. Har bir tizimda barcha tahdidlarni to'liq bartaraf etish imkonsiz bo'lgani sababli, xavflarni aniqlash, baholash va ularni maqbul darajagacha kamaytirish strategiyasi ishlab chiqiladi. Bu yondashuv resurslardan samarali foydalanish imkonini beradi.

Xulosa qilib aytganda, kiberxavfsizlik zamonaviy raqamli jamiyatning ajralmas qismi bo'lib, u nafaqat texnik vositalar, balki tashkiliy choralar va inson omiliga asoslangan tamoyillarni ham o'z ichiga oladi. Kiberxavfsizlikning asosiy tushunchalari va tamoyillarini anglash axborot tizimlaridan xavfsiz va samarali foydalanishni ta'minlaydi hamda raqamli muhitda barqaror faoliyat yuritish uchun mustahkam asos bo'lib xizmat qiladi.

Raqamli texnologiyalar keng rivojlanayotgan hozirgi davrda shaxsiy ma'lumotlar jamiyat hayotida muhim axborot resursiga aylanib bormoqda. Ta'lim, sog'liqni saqlash, moliya, davlat boshqaruvi va ijtimoiy tarmoqlarda shaxsga oid ma'lumotlar doimiy ravishda yig'iladi, qayta ishlanadi va uzatiladi. Shu sababli shaxsiy ma'lumotlarni himoyalash masalasi nafaqat texnik, balki huquqiy, tashkiliy va axloqiy jihatdan ham dolzarb ahamiyat kasb etadi.

Shaxsiy ma'lumotlar deganda ma'lum bir jismoniy shaxsni bevosita yoki bilvosita aniqlash imkonini beruvchi har qanday axborot tushuniladi. Bunga

shaxsning ismi, familiyasi, tug‘ilgan sanasi, pasport ma’lumotlari, telefon raqami, elektron pochta manzili, yashash joyi, biometrik ma’lumotlari, shuningdek, internetdagi faoliyati bilan bog‘liq axborotlar kiradi. Raqamli muhitda bunday ma’lumotlarning hajmi va turlari tobora kengayib bormoqda.

Shaxsiy ma’lumotlarning asosiy xususiyati shundan iboratki, ular noto‘g‘ri qo‘llanilganda yoki ruxsatsiz tarqalganda shaxsning huquqlari va xavfsizligiga jiddiy zarar yetkazishi mumkin. Masalan, shaxsiy ma’lumotlar asosida moliyaviy firibgarlik, shaxsni soxtalashtirish, ijtimoiy muhandislik hujumlari yoki shaxsiy hayotga aralashuv holatlari yuzaga keladi. Shu bois shaxsiy ma’lumotlarni himoyalash zamonaviy axborot xavfsizligining muhim yo‘nalishlaridan biri hisoblanadi.

Shaxsiy ma’lumotlarni himoyalash tushunchasi ularni yig‘ish, saqlash, qayta ishlash va uzatish jarayonlarida maxfiylik, yaxlitlik va mavjudlikni ta’minlashga qaratilgan choralar majmuasini anglatadi. Bu tushuncha kiberxavfsizlik va axborot xavfsizligi bilan uzviy bog‘liq bo‘lib, ularning amaliy ko‘rinishlaridan biri sifatida qaraladi. Himoya jarayonida texnik vositalar bilan bir qatorda huquqiy me‘yorlar va tashkiliy qoidalar ham muhim rol o‘ynaydi.

Shaxsiy ma’lumotlarni himoyalashning asosiy tamoyillaridan biri — **qonuniylik va maqsadga muvofiqlik** tamoyilidir. Ushbu tamoyilga ko‘ra, shaxsiy ma’lumotlar faqat aniq belgilangan va qonuniy maqsadlar uchun yig‘ilishi va qayta ishlanishi lozim. Ma’lumotlar egasining rozilgisiz yoki belgilangan maqsaddan tashqari foydalanish shaxsiy huquqlarning buzilishiga olib keladi.

Yana bir muhim tamoyil — **ma’lumotlarni minimallashtirish** hisoblanadi. Bu tamoyilga muvofiq, tizim yoki tashkilot shaxsiy ma’lumotlarning faqat zarur bo‘lgan qismini yig‘ishi va saqlashi kerak. Ortiqcha ma’lumotlarni to‘plash xavflarni oshiradi va himoyalashni murakkablashtiradi. Ayniqsa, ta’lim platformalarida talaba yoki o‘quvchi haqida faqat o‘quv jarayoni uchun zarur bo‘lgan axborot bilan cheklanish muhimdir.

Shaxsiy ma’lumotlarni himoyalashda **maxfiylik** tamoyili asosiy o‘rinni egallaydi. Maxfiylik ma’lumotlarga faqat ruxsat etilgan shaxslar kirishini ta’minlashni anglatadi. Buning uchun parollar, autentifikatsiya va avtorizatsiya mexanizmlari, shifrlash texnologiyalari qo‘llaniladi. Kuchli parollar va ikki bosqichli autentifikatsiya shaxsiy ma’lumotlarni himoyalashning samarali usullaridan hisoblanadi.

Yaxlitlik tamoyili shaxsiy ma’lumotlarning ruxsatsiz o‘zgartirilmasligini ta’minlashga qaratilgan. Ma’lumotlar buzilishi yoki noto‘g‘ri tahrirlanishi shaxs haqida noto‘g‘ri qarorlar qabul qilinishiga olib kelishi mumkin. Shu sababli ma’lumotlar bazalarini himoyalash, zaxira nusxalar yaratish va nazorat mexanizmlarini joriy etish muhim hisoblanadi.

Shaxsiy ma'lumotlarning **mavjudligi** ham muhim omil bo'lib, ma'lumotlar zarur bo'lgan vaqtda qonuniy foydalanuvchilar uchun ochiq bo'lishi kerak. Texnik nosozliklar yoki kiberhujumlar natijasida ma'lumotlarga kirish imkonining yo'qolishi ta'lim va boshqaruv jarayonlarida jiddiy muammolarni keltirib chiqaradi. Shu bois tizimlarning barqarorligi va uzluksiz ishlashi ta'minlanishi lozim.

Shaxsiy ma'lumotlarni himoyalashda inson omili alohida ahamiyatga ega. Ko'plab ma'lumotlar sizib chiqishi holatlari texnik kamchiliklar emas, balki foydalanuvchilarning ehtiyotsizligi, phishing xabarlariga aldanishi yoki parollarni oshkor qilishi bilan bog'liq. Shu sababli foydalanuvchilarda axborot xavfsizligi madaniyatini shakllantirish, ularni shaxsiy ma'lumotlarni ehtiyot qilishga o'rgatish muhim vazifa hisoblanadi.

Ta'lim muassasalarida shaxsiy ma'lumotlarni himoyalash alohida mas'uliyat talab qiladi. O'quvchilar va talabalar haqidagi ma'lumotlar, baholar, davomat va shaxsiy fayllar maxfiy axborot sifatida ko'rilishi lozim. Raqamli ta'lim platformalarida ushbu ma'lumotlarni himoyalash nafaqat texnik talab, balki pedagogik va axloqiy majburiyatdir.

Xulosa qilib aytganda, shaxsiy ma'lumotlar zamonaviy raqamli jamiyatda muhim strategik resurs bo'lib, ularni himoyalash shaxs huquqlari va axborot xavfsizligini ta'minlashning ajralmas qismi hisoblanadi. Shaxsiy ma'lumotlarni himoyalashning asosiy tamoyillarini tushunish va amaliyotda qo'llash raqamli muhitda xavfsiz va mas'uliyatli faoliyat yuritish uchun mustahkam asos yaratadi.

Raqamli texnologiyalar va internet infratuzilmasining kengayishi bilan birga axborot tizimlariga nisbatan kiberxavflar va tahdidlar ham tobora murakkablashib bormoqda. Bugungi kunda kiberxavflar nafaqat alohida foydalanuvchilar, balki ta'lim muassasalari, davlat tashkilotlari, bank-moliya tizimi va yirik korxonalar faoliyatiga jiddiy xavf tug'dirmoqda. Shu sababli kiberxavflar va tahdidlar turlarini o'rganish, ularning mohiyatini tushunish hamda oldini olish zamonaviy axborot xavfsizligining muhim vazifalaridan biridir.

Kiberxavf tushunchasi axborot tizimlariga zarar yetkazishi mumkin bo'lgan potentsial holat yoki sharoitni anglatadi. Kiberxavf real hujum bo'lmasligi mumkin, ammo u mavjud zaifliklar orqali tahdidga aylanish ehtimoliga ega. Kiberxavf amalga oshirilganda esa u **kiber tahdid** sifatida namoyon bo'ladi. Demak, tahdid — bu xavfning amaliy ko'rinishi hisoblanadi.

Kiberxavflar va tahdidlar kelib chiqish manbasiga ko'ra **ichki** va **tashqi** turlarga bo'linadi. Ichki tahdidlar tashkilot ichidagi xodimlar yoki foydalanuvchilar bilan bog'liq bo'lib, ko'pincha ehtiyotsizlik, malakasizlik yoki qasddan qilingan harakatlar natijasida yuzaga keladi. Masalan, kuchsiz parollardan foydalanish, maxfiy ma'lumotlarni begonalarga yuborish yoki xavfsizlik qoidalariga rioya qilmaslik ichki

tahdidlar sirasiga kiradi. Tashqi tahdidlar esa xakerlar, kiberjinoyatchilar, zararli dasturlar va tashqi tarmoqlar orqali amalga oshiriladigan hujumlar bilan bog‘liq.

Kiber tahdidlarning eng keng tarqalgan turlaridan biri — **zararli dasturlar (malware)** hisoblanadi. Zararli dasturlar foydalanuvchi tizimiga ruxsatsiz kirib, ma’lumotlarni o‘g‘irlash, o‘zgartirish yoki yo‘q qilishga qaratilgan bo‘ladi. Ularning turlariga viruslar, qurtlar (worm), troyan dasturlar, josuslik dasturlari (spyware) va reklama dasturlari (adware) kiradi. Zararli dasturlar ko‘pincha elektron pochta ilovalari, shubhali havolalar yoki noqonuniy dasturiy ta’minot orqali tarqaladi.

Kiberxavflarning yana bir xavfli turi — **ransomware** (tovon puli talab qiluvchi dasturlar) hisoblanadi. Ushbu turdagi tahdidlar foydalanuvchi ma’lumotlarini shifrlab, ularni qayta ochish evaziga pul talab qiladi. Ransomware hujumlari ta’lim muassasalari va tibbiyot tashkilotlari uchun ayniqsa xavfli bo‘lib, muhim axborotlarning vaqtincha yoki doimiy yo‘qolishiga olib kelishi mumkin.

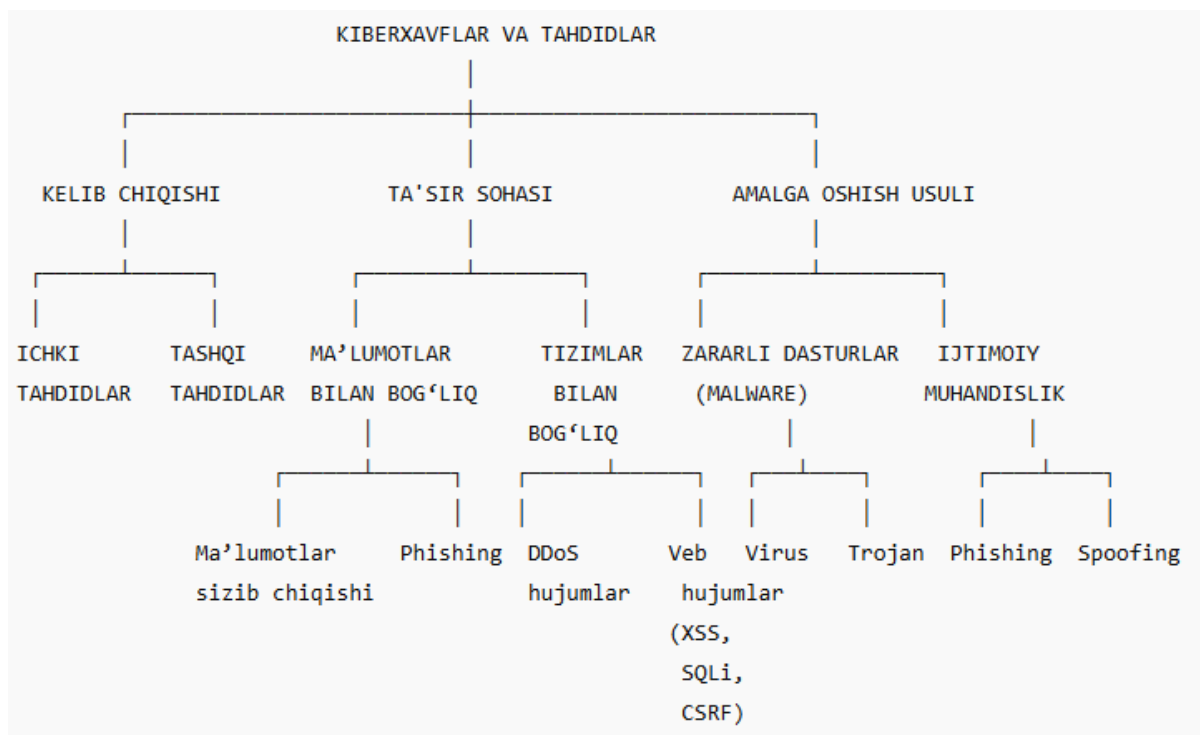
Phishing (firibgarlik orqali ma’lumotlarni qo‘lga kiritish) ham keng tarqalgan kiber tahdidlardan biridir. Phishing hujumlarida foydalanuvchilarga soxta elektron xatlar, xabarlar yoki veb-sahifalar yuborilib, ularning parollari, bank rekvizitlari yoki shaxsiy ma’lumotlari qo‘lga kiritiladi. Bu turdagi hujumlar ko‘pincha ijtimoiy muhandislik usullariga asoslanadi va foydalanuvchining ishonuvchanligidan foydalanadi.

Kiberxavflar orasida **tarmoq hujumlari** ham muhim o‘rin tutadi. Bunga xizmatni rad etish hujumlari (DDoS), tarmoq trafikini tinglash (sniffing), soxta marshrutlash va man-in-the-middle hujumlari kiradi. DDoS hujumlarida serverga juda ko‘p so‘rovlar yuborilib, tizimning ishlashi izdan chiqariladi. Natijada foydalanuvchilar xizmatlardan foydalana olmay qoladilar.

Ma’lumotlar sizib chiqishi (data breach) ham jiddiy kiber tahdid hisoblanadi. Bu holatda shaxsiy, moliyaviy yoki maxfiy ma’lumotlar ruxsatsiz shaxslarga ma’lum bo‘lib qoladi. Ma’lumotlar sizib chiqishi tashkilot obro‘sigga putur yetkazishi, moliyaviy zarar va huquqiy muammolarga olib kelishi mumkin. Ko‘pincha bunday holatlar zaif himoyalangan bazalar yoki noto‘g‘ri sozlangan serverlar sababli yuzaga keladi.

Kiberxavflarning yana bir turi — **veb-ilovalar bilan bog‘liq tahdidlardir**. SQL injection, XSS (cross-site scripting), CSRF (cross-site request forgery) kabi hujumlar veb-saytlar va onlayn platformalardagi zaifliklardan foydalanadi. Ta’lim platformalarida bunday hujumlar talabalar ma’lumotlari va baholash tizimlariga zarar yetkazishi mumkin.

Kiberxavflar va tahdidlar turlari



So‘nggi yillarda **insider tahdidlar**, ya’ni ichki foydalanuvchilar tomonidan qasddan yoki bexosdan yetkaziladigan zararlar ham dolzarb muammoga aylanmoqda. Xodimning norozi bo‘lishi, bilim yetishmasligi yoki e’tiborsizligi natijasida muhim ma’lumotlar yo‘qotilishi yoki oshkor bo‘lishi mumkin. Bu tahdidlarni aniqlash tashqi hujumlarga qaraganda murakkabroq bo‘ladi.

Kiberxavflar va tahdidlarning yana bir zamonaviy turi — **sun’iy intellekt va avtomatlashtirilgan hujumlar** bilan bog‘liqdir. Avtomatik botlar va aqlli hujum mexanizmlari qisqa vaqt ichida ko‘plab tizimlarni nishonga olishi mumkin. Bu esa an’anaviy himoya usullarining yetarli bo‘lmavchiligiga olib keladi.

Xulosa qilib aytganda, kiberxavflar va tahdidlar ko‘p qirrali va doimiy rivojlanib boruvchi hodisalardir. Ular texnik, tashkiliy va inson omiliga bog‘liq jihatlarni o‘z ichiga oladi. Kiberxavflar turlarini chuqur tushunish, ularning kelib chiqish sabablarini tahlil qilish va mos himoya choralarini ishlab chiqish axborot tizimlarining barqarorligi va xavfsizligini ta’minlashda muhim ahamiyat kasb etadi.

Raqamli texnologiyalar rivoji natijasida shaxsiy va korporativ ma’lumotlar jamiyatning eng muhim axborot resurslaridan biriga aylandi. Kundalik hayotda shaxsiy ma’lumotlar turli onlayn xizmatlar, ijtimoiy tarmoqlar va ta’lim platformalari orqali qayta ishlanadi. Korporativ muhitda esa moliyaviy hujjatlar, mijozlar bazasi, ichki hujjatlar va strategik axborotlar bilan ishlanadi. Ushbu ma’lumotlarning ruxsatsiz oshkor bo‘lishi yoki yo‘qolishi jiddiy huquqiy, moliyaviy va obro‘ga putur yetkazuvchi oqibatlariga olib kelishi mumkin. Shu sababli shaxsiy va korporativ ma’lumotlarni himoyalash zamonaviy axborot xavfsizligining ustuvor yo‘nalishlaridan biri hisoblanadi.

Shaxsiy ma'lumotlarni himoyalash usullari avvalo foydalanuvchining o'ziga bog'liq choralarni o'z ichiga oladi. Eng muhim usullardan biri kuchli parollardan foydalanishdir. Parollar kamida sakkiz belgidan iborat bo'lishi, harflar, raqamlar va maxsus belgilarni o'z ichiga olishi tavsiya etiladi. Bir xil paroldan bir nechta xizmatlarda foydalanish xavf tug'diradi, shuning uchun har bir platforma uchun alohida parol tanlash maqsadga muvofiqdir. Parollarni boshqarish uchun maxsus parol menejerlaridan foydalanish shaxsiy ma'lumotlar xavfsizligini oshiradi.

Shaxsiy ma'lumotlarni himoyalashda ikki bosqichli autentifikatsiya muhim ahamiyatga ega. Bu usul foydalanuvchi tizimga kirishda nafaqat parolni, balki qo'shimcha tasdiqlash vositasini ham talab qiladi. Masalan, SMS orqali yuborilgan kod, mobil ilova yoki biometrik ma'lumotlar. Ushbu yondashuv hisoblarni noqonuniy egallab olish xavfini sezilarli darajada kamaytiradi.

Yana bir muhim usul — shaxsiy ma'lumotlarni ongli va ehtiyotkorlik bilan tarqatishdir. Foydalanuvchilar ijtimoiy tarmoqlarda, veb-saytlarda yoki onlayn so'rovnomalarda shaxsiy axborotni oshkor qilishda ehtiyot bo'lishlari lozim. Noma'lum havolalarga bosmaslik, shubhali elektron xatlarni ochmaslik va rasmiy bo'lmagan manbalarga ma'lumot kiritmaslik phishing va firibgarlik hujumlaridan himoyalalanishning samarali choralardan biridir.

Korporativ ma'lumotlarni himoyalash esa yanada kompleks va tizimli yondashuvni talab qiladi. Tashkilotlarda axborot xavfsizligini ta'minlash uchun texnik, tashkiliy va huquqiy choralar birgalikda qo'llaniladi. Eng avvalo, korporativ tarmoqlarda kirishni boshqarish tizimlari joriy etiladi. Foydalanuvchilarga faqat o'z vazifalarini bajarish uchun zarur bo'lgan minimal huquqlar beriladi. Bu yondashuv "minimal huquqlar prinsipi" deb ataladi va ichki tahdidlar xavfini kamaytiradi.

Korporativ muhitda ma'lumotlarni shifrlash keng qo'llaniladigan himoya usullaridan biridir. Shifrlash jarayonida ma'lumotlar maxsus algoritmlar yordamida o'qib bo'lmaydigan shaklga keltiriladi. Natijada, ma'lumotlar ruxsatsiz qo'lga tushgan taqdirda ham ulardan foydalanish imkonsiz bo'ladi. Shifrlash saqlanayotgan ma'lumotlar uchun ham, tarmoq orqali uzatilayotgan axborotlar uchun ham qo'llaniladi.

Zaxira nusxalarini yaratish ham shaxsiy va korporativ ma'lumotlarni himoyalashda muhim usullardan biridir. Ma'lumotlarning doimiy ravishda zaxira qilinishi texnik nosozliklar, kiberhujumlar yoki tasodifiy o'chirish holatlarida axborotni tez tiklash imkonini beradi. Zaxira nusxalar alohida, ishonchli va imkon qadar oflayn muhitda saqlanishi tavsiya etiladi.

Korporativ muhitda tarmoq xavfsizligini ta'minlash uchun xavfsizlik devorlari (firewall), antivirus dasturlari va hujumlarni aniqlash tizimlari qo'llaniladi. Ushbu vositalar zararli trafikni aniqlash, ruxsatsiz kirishni cheklash va zararli dasturlarni bartaraf etishga xizmat qiladi. Tizimlarni muntazam yangilab borish ham muhim

ahamiyatga ega, chunki ko‘plab kiberhujumlar eskirgan dasturiy ta‘minotdagi zaifliklardan foydalanadi.

Shaxsiy va korporativ ma‘lumotlarni himoyalashda inson omili alohida o‘rin tutadi. Xodimlarning ehtiyotsizligi yoki axborot xavfsizligi qoidalarini bilmasligi jiddiy muammolarga olib kelishi mumkin. Shu sababli tashkilotlarda axborot xavfsizligi bo‘yicha muntazam treninglar o‘tkazish, xodimlarni phishing va ijtimoiy muhandislik hujumlari haqida xabardor qilish muhim hisoblanadi. Xavfsizlik madaniyatini shakllantirish texnik choralar qanchalik mukammal bo‘lmasin, ularning samaradorligini oshiradi.

Huquqiy va tashkiliy choralar ham ma‘lumotlarni himoyalashning ajralmas qismi hisoblanadi. Tashkilotlarda shaxsiy va korporativ ma‘lumotlar bilan ishlash bo‘yicha ichki siyosat va qoidalar ishlab chiqilishi lozim. Ushbu hujjatlar ma‘lumotlarni yig‘ish, saqlash, qayta ishlash va uzatish tartibini belgilaydi hamda mas‘uliyatni aniq taqsimlaydi.

Zamonaviy raqamli jamiyatda axborot tizimlari, tarmoqlar va raqamli xizmatlardan foydalanish bilan bog‘liq xavf-xatarlar tobora ko‘payib bormoqda. Kiberhujumlar, ma‘lumotlar sizib chiqishi, texnik nosozliklar va inson omili bilan bog‘liq xatolar axborot xavfsizligiga jiddiy tahdid soladi. Shu sababli xavf-xatarlarning oldini olish choralari nafaqat texnik masala, balki tashkiliy, huquqiy va madaniy yondashuvlarni ham o‘z ichiga olgan kompleks jarayon hisoblanadi. Xavflarni oldindan aniqlash va ularni kamaytirish raqamli tizimlarning barqarorligi va ishonchliligini ta‘minlashda muhim ahamiyatga ega.

Xavf-xatarlarning oldini olishning eng muhim choralaridan biri — **risklarni aniqlash va baholash**dir. Har bir axborot tizimi uchun mavjud tahdidlar, zaifliklar va ularning mumkin bo‘lgan oqibatlarini tahlil qilinishi lozim. Risklarni baholash orqali qaysi xavflar ustuvor ekanini aniqlash va resurslarni aynan shu yo‘nalishlarga yo‘naltirish mumkin. Bu yondashuv tasodifiy emas, balki ilmiy asoslangan xavfsizlik siyosatini shakllantirishga yordam beradi.

Texnik choralar xavf-xatarlarning oldini olishda asosiy o‘rin tutadi. Ularga axborot tizimlarini muntazam yangilab borish, operatsion tizim va dasturiy ta‘minotdagi zaifliklarni bartaraf etish kiradi. Ko‘plab kiberhujumlar aynan eskirgan dasturiy ta‘minotdan foydalanish orqali amalga oshiriladi. Shu bois avtomatik yangilash mexanizmlarini joriy etish va xavfsizlik yamoqlarini o‘z vaqtida o‘rnatish muhim hisoblanadi.

Tarmoq darajasida xavf-xatarlarning oldini olish uchun **xavfsizlik devorlari (firewall)**, tarmoq trafikini filtrlash va hujumlarni aniqlash tizimlaridan foydalaniladi. Ushbu vositalar ruxsatsiz kirishlarni cheklash, zararli trafikni aniqlash va potentsial hujumlarni erta bosqichda to‘xtatishga xizmat qiladi. Ayniqsa, korporativ va ta‘lim muassasalari tarmoqlarida bunday himoya mexanizmlari majburiy ahamiyatga ega.

Ma'lumotlarni himoyalash xavf-xatarlarning oldini olishda muhim choralar sirasiga kiradi. Ma'lumotlarni shifrlash orqali ularni ruxsatsiz foydalanishdan himoyalash mumkin. Shuningdek, muhim axborotlar uchun muntazam **zaxira nusxalarini yaratish** kiberhujumlar, texnik nosozliklar yoki tasodifiy o'chirish holatlarida ma'lumotlarni tez tiklash imkonini beradi. Zaxira nusxalarni alohida va ishonchli muhitda saqlash tavsiya etiladi.

Xavf-xatarlarning oldini olishda **foydalanuvchilarni autentifikatsiya va avtorizatsiya qilish** mexanizmlari alohida ahamiyatga ega. Kuchli parollar, ikki bosqichli autentifikatsiya va foydalanuvchi huquqlarini aniq belgilash ruxsatsiz kirish xavfini sezilarli darajada kamaytiradi. Ayniqsa, minimal huquqlar prinsipi asosida foydalanuvchilarga faqat zarur bo'lgan ruxsatlarni berish tizimdagi zarar miqyosini cheklaydi.

Inson omili bilan bog'liq xavf-xatarlarning oldini olish uchun **axborot xavfsizligi madaniyatini shakllantirish** zarur. Ko'plab tahdidlar foydalanuvchilarning ehtiyotsizligi, phishing xabarlariga ishonishi yoki xavfsizlik qoidalariga rioya qilmasligi natijasida yuzaga keladi. Shu sababli xodimlar va foydalanuvchilar uchun muntazam treninglar, seminarlar va tushuntirish ishlari olib borilishi muhimdir. Foydalanuvchi tahdidlarni taniy olsa, texnik himoya vositalarining samaradorligi yanada oshadi.

Tashkiliy choralar ham xavf-xatarlarning oldini olishda muhim o'rin tutadi. Tashkilotlarda axborot xavfsizligi siyosati ishlab chiqilishi, xavfsizlik bo'yicha ichki qoidalar va yo'riqnomalar joriy etilishi lozim. Ushbu hujjatlar axborot bilan ishlash tartibini, mas'uliyat va vakolatlarni aniq belgilaydi. Shuningdek, favqulodda holatlar uchun harakat rejasi ishlab chiqilishi tizimlarning tez tiklanishini ta'minlaydi.

Huquqiy choralar xavf-xatarlarning oldini olishda muhim qo'shimcha vosita hisoblanadi. Shaxsiy va korporativ ma'lumotlarni himoyalash bo'yicha qonun va me'yoriy hujjatlarga rioya qilish tashkilot va foydalanuvchilarni huquqiy javobgarlikdan himoya qiladi. Bu yondashuv axborot xavfsizligini faqat texnik emas, balki huquqiy majburiyat sifatida ham qarashni taqozo etadi.

Monitoring va audit jarayonlari xavf-xatarlarning oldini olishda doimiy nazoratni ta'minlaydi. Tizim faoliyatini muntazam kuzatish, xavfsizlik jurnallarini tahlil qilish va audit o'tkazish orqali potentsial tahdidlar erta aniqlanadi. Bu esa ularning real muammoga aylanishidan oldin choralar ko'rish imkonini beradi.

Raqamli texnologiyalar ta'lim muassasalari va turli tashkilotlar faoliyatining ajralmas qismiga aylanib bormoqda. Elektron jurnal va kundaliklar, onlayn ta'lim platformalari, masofaviy darslar, korporativ axborot tizimlari, elektron hujjat aylanishi va bulutli xizmatlar ta'lim jarayonini yanada qulay va samarali qilish imkonini beradi. Biroq raqamli muhit kengaygani sari axborot xavfsizligiga bo'lgan talablar ham keskin ortmoqda. Shu sababli ta'lim muassasalari va tashkilotlarda

kiberxavfsizlikni ta'minlash zamonaviy boshqaruv va ta'lim tizimining muhim vazifalaridan biridir.

Ta'lim muassasalari axborot tizimlarida o'quvchilar, talabalar, o'qituvchilar va xodimlarga oid katta hajmdagi shaxsiy ma'lumotlar saqlanadi. Baholar, davomat, shaxsiy hujjatlar, tibbiy yoki ijtimoiy ma'lumotlar kabi axborotlarning ruxsatsiz oshkor bo'lishi jiddiy huquqiy va axloqiy muammolarga olib kelishi mumkin. Tashkilotlarda esa moliyaviy hisobotlar, strategik hujjatlar, mijozlar bazasi va ichki yozishmalar muhim axborot resurslari hisoblanadi. Shu bois kiberxavfsizlik nafaqat texnik masala, balki muassasaning obro'si va barqaror faoliyatiga ta'sir qiluvchi strategik omildir.

Ta'lim muassasalari va tashkilotlarda kiberxavfsizlikning asosiy maqsadi axborotning maxfiyligi, yaxlitligi va mavjudligini ta'minlashdan iborat. Maxfiylik ma'lumotlarga faqat ruxsat etilgan shaxslar kirishini anglatadi. Yaxlitlik axborotning ruxsatsiz o'zgartirilmasligini va uning ishonchliligini saqlashni bildiradi. Mavjudlik esa tizimlar va xizmatlar zarur paytda foydalanuvchilar uchun ochiq bo'lishini ta'minlashga qaratilgan. Ushbu tamoyillar kiberxavfsizlikning asosiy ustunlari hisoblanadi.

Ta'lim muassasalarida kiberxavfsizlikka tahdidlar ko'pincha bir necha manbadan kelib chiqadi. Bular tashqi tahdidlar, masalan, xakerlik hujumlari, zararli dasturlar, phishing xatlari bo'lsa, ichki tahdidlar xodimlar yoki o'quvchilarning ehtiyotsizligi, parollarni oshkor qilishi yoki xavfsizlik qoidalariga rioya qilmasligi bilan bog'liq bo'lishi mumkin. Ayniqsa, masofaviy ta'lim sharoitida shaxsiy qurilmalardan foydalanish ichki tahdidlar xavfini oshiradi.

Kiberxavfsizlikni ta'minlashda texnik choralar muhim ahamiyatga ega. Ta'lim muassasalari va tashkilotlarda antivirus dasturlari, xavfsizlik devorlari, hujumlarni aniqlash va oldini olish tizimlari joriy etilishi lozim. Operatsion tizim va dasturiy ta'minotni muntazam yangilab borish, zaifliklarni bartaraf etish kiberhujumlarning oldini olishda samarali vosita hisoblanadi. Shuningdek, tarmoqni segmentlash orqali muhim axborot resurslarini umumiy foydalanish tarmog'idan ajratish xavfsizlik darajasini oshiradi.

Ma'lumotlarni himoyalash ham kiberxavfsizlikning muhim tarkibiy qismidir. Ta'lim muassasalarida saqlanayotgan shaxsiy va akademik ma'lumotlar shifrlangan holda saqlanishi va uzatilishi lozim. Zaxira nusxalarini muntazam yaratish esa texnik nosozliklar yoki ransomware hujumlari natijasida ma'lumotlarni yo'qotish xavfini kamaytiradi. Zaxira nusxalar alohida, ishonchli va imkon qadar oflayn muhitda saqlanishi tavsiya etiladi.

Tashkiliy choralar kiberxavfsizlikni ta'minlashda kamida texnik vositalar qadar muhimdir. Har bir ta'lim muassasasi yoki tashkilotda axborot xavfsizligi siyosati ishlab chiqilishi lozim. Ushbu siyosat foydalanuvchilarning huquq va

majburiyatlarini, axborot bilan ishlash qoidalarini, favqulodda holatlarda harakat qilish tartibini belgilaydi. Shuningdek, foydalanuvchilarga beriladigan ruxsatlar “minimal huquqlar prinsipi” asosida taqsimlanishi zarur.

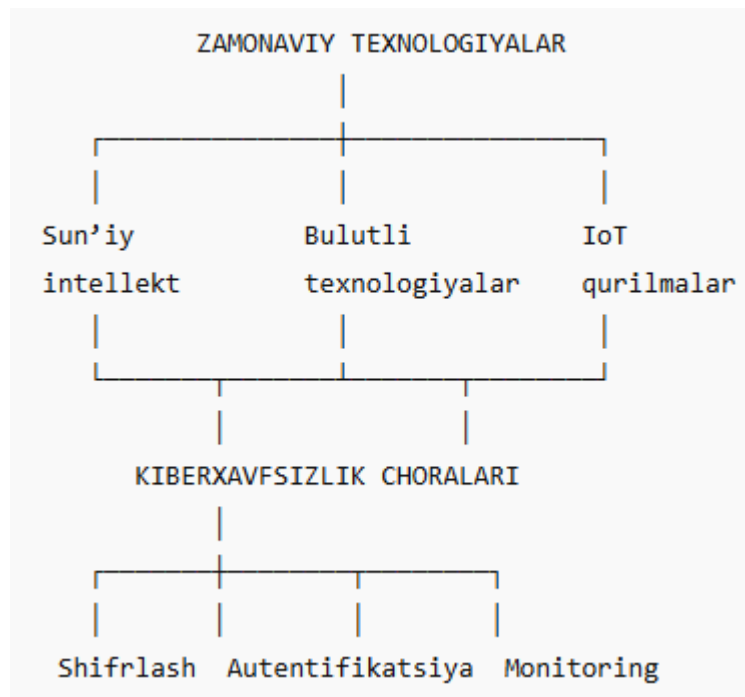
Inson omili ta’lim muassasalari va tashkilotlarda kiberxavfsizlikning eng zaif, ammo eng muhim jihatlaridan biridir. Ko‘plab kiberhodisalar texnik kamchiliklardan emas, balki foydalanuvchilarning xatolari natijasida yuzaga keladi. Shu sababli o‘qituvchilar, xodimlar va talabalar uchun kiberxavfsizlik bo‘yicha muntazam treninglar o‘tkazish, phishing va ijtimoiy muhandislik usullari haqida xabardorlikni oshirish muhimdir. Axborot xavfsizligi madaniyatini shakllantirish umumiy himoya tizimining samaradorligini sezilarli darajada oshiradi.

Ta’lim muassasalarida kiberxavfsizlik masalasi pedagogik jarayon bilan ham chambarchas bog‘liq. O‘quvchilarga raqamli savodxonlik, mas’uliyatli internetdan foydalanish, shaxsiy ma’lumotlarni himoyalash ko‘nikmalarini o‘rgatish zamonaviy ta’limning muhim vazifalaridan biridir. Bu nafaqat muassasa xavfsizligini, balki jamiyatning umumiy kiberxavfsizlik darajasini ham oshiradi.

Huquqiy jihatlar ham kiberxavfsizlikda muhim o‘rin tutadi. Ta’lim muassasalari va tashkilotlar shaxsiy ma’lumotlarni himoyalashga oid milliy va xalqaro me’yoriy hujjatlarga rioya qilishi lozim. Bu foydalanuvchilar huquqlarini himoyalash bilan birga, tashkilotning huquqiy javobgarligini kamaytiradi.

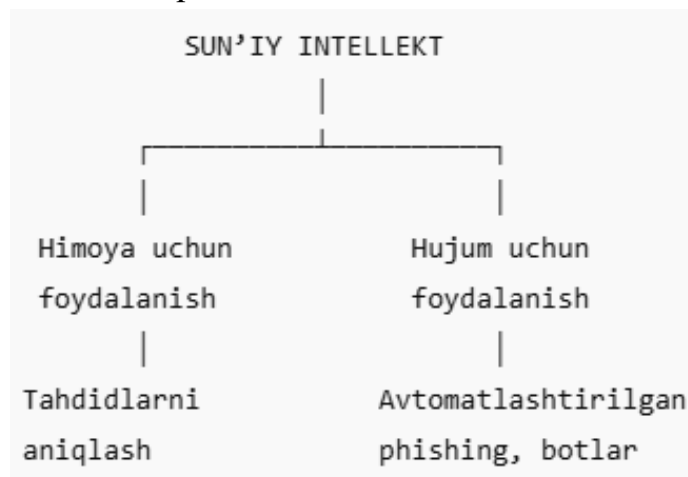
Zamonaviy texnologiyalar jadal rivojlanib borayotgan bugungi davrda jamiyatning deyarli barcha sohalari raqamli muhit bilan uzviy bog‘liq holda faoliyat yuritmoqda. Sun’iy intellekt, bulutli hisoblash, katta ma’lumotlar (Big Data), Internet of Things (IoT), mobil texnologiyalar va blokcheyn kabi innovatsion yechimlar iqtisodiyot, ta’lim, sog‘liqni saqlash va davlat boshqaruvi tizimlarida samaradorlikni oshirish imkonini bermoqda. Biroq ushbu texnologiyalar bilan bir qatorda kiberxavfsizlik masalalari ham yanada murakkablashib, dolzarb ahamiyat kasb etmoqda.

Zamonaviy texnologiyalar va kiberxavfsizlikning o‘zaro bog‘liqligi

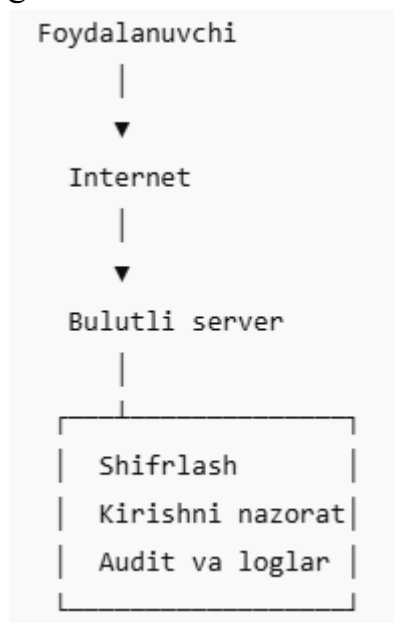


Zamonaviy texnologiyalar axborot almashinuvi tezligini oshiradi, ma'lumotlar hajmini ko'paytiradi va masofaviy ishlash imkoniyatlarini kengaytiradi. Shu bilan birga, raqamli tizimlarga bo'lgan ishonch kiberxavfsizlik darajasiga bevosita bog'liq bo'lib qolmoqda. Agar axborot tizimlari yetarli darajada himoyalangan bo'lsa, zamonaviy texnologiyalarning afzalliklari jiddiy xavflarga aylanishi mumkin.

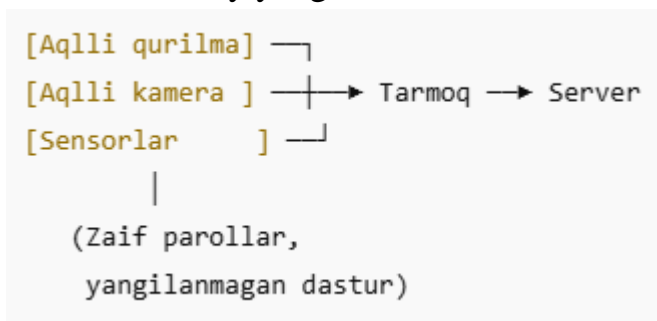
Sun'iy intellekt va mashinali o'qitish texnologiyalari bugungi kunda ko'plab sohalarda qo'llanilmoqda. Ular ma'lumotlarni tahlil qilish, qarorlar qabul qilish va jarayonlarni avtomatlashtirishda katta imkoniyatlar yaratadi. Biroq sun'iy intellekt kiberxavfsizlik sohasida ikki tomonlama rol o'ynaydi. Bir tomondan, u tahdidlarni aniqlash, anomal xatti-harakatlarni topish va hujumlarni oldindan bashorat qilishda samarali vosita bo'lib xizmat qiladi. Ikkinchi tomondan esa, kiberjinoyatchilar ham sun'iy intellektdan foydalangan holda murakkab va avtomatlashtirilgan hujumlarni amalga oshirishlari mumkin. Shu sababli sun'iy intellekt asosidagi tizimlarni himoyalash alohida e'tibor talab qiladi.



Bulutli texnologiyalar zamonaviy axborot infratuzilmasining muhim qismiga aylangan. Bulutli xizmatlar ma'lumotlarni masofadan saqlash va ularga istalgan joydan kirish imkonini beradi. Bu qulayliklar bilan birga, ma'lumotlar xavfsizligi masalalarini ham keltirib chiqaradi. Bulutli muhitda shaxsiy va korporativ ma'lumotlar ko'pincha uchinchi tomon serverlarida saqlanadi, bu esa maxfiylik va nazorat masalalarini murakkablashtiradi. Shu bois zamonaviy kiberxavfsizlik strategiyalari bulutli texnologiyalar uchun shifrlash, kuchli autentifikatsiya va xavfsizlik monitoringini o'z ichiga olishi lozim.

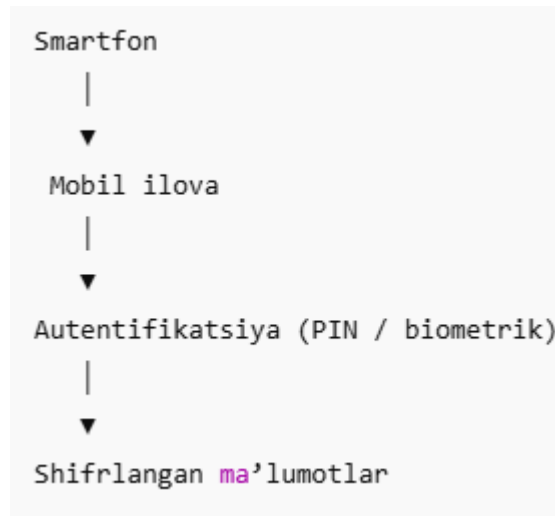


Internet of Things texnologiyalari ham kiberxavfsizlik nuqtai nazaridan muhim ahamiyatga ega. Aqlli qurilmalar, sensorlar va avtomatlashtirilgan tizimlar kundalik hayot va sanoat jarayonlarida keng qo'llanilmoqda. Biroq IoT qurilmalari ko'pincha cheklangan hisoblash imkoniyatlariga ega bo'lgani sababli yetarli darajada himoyalanganmaydi. Natijada ular kiberhujumlar uchun oson nishonga aylanishi mumkin. IoT muhitida kiberxavfsizlikni ta'minlash uchun qurilmalarni identifikatsiya qilish, tarmoqni segmentlash va doimiy yangilab borish muhim hisoblanadi.



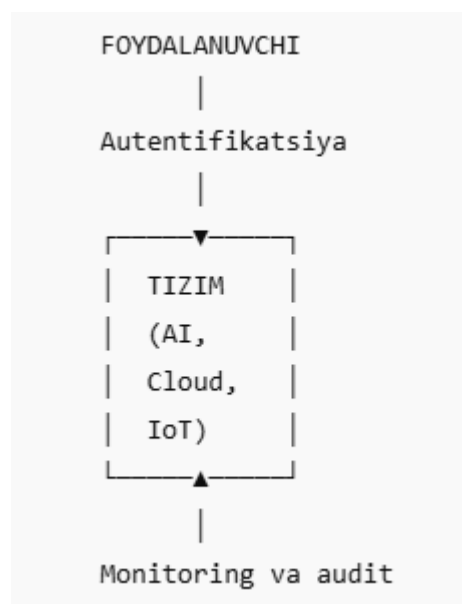
Zamonaviy mobil texnologiyalar ham kiberxavfsizlik bilan chambarchas bog'liq. Smartfonlar va planshetlar orqali foydalanuvchilar bank xizmatlari, ta'lim platformalari va korporativ tizimlarga ulanadilar. Mobil qurilmalarning yo'qolishi yoki zararli dasturlar bilan zararlanishi shaxsiy va korporativ ma'lumotlar sizib

chiqishiga olib kelishi mumkin. Shu sababli mobil qurilmalarda parol va biometrik himoya, ilovalarni rasmiy manbalardan yuklash va doimiy yangilash kiberxavfsizlikning muhim choralaridan hisoblanadi.



Katta ma'lumotlar texnologiyalari (Big Data) zamonaviy boshqaruv va tahlil jarayonlarida muhim o'rin tutadi. Katta hajmdagi ma'lumotlar asosida strategik qarorlar qabul qilinadi, biroq bu ma'lumotlarning himoyasi yetarli darajada ta'minlanmasa, jiddiy xavflar yuzaga keladi. Katta ma'lumotlar bilan ishlashda kiberxavfsizlik nafaqat texnik, balki huquqiy va axloqiy masala sifatida ham qaraladi.

Zamonaviy texnologiyalar sharoitida kiberxavfsizlik faqat texnik vositalar bilan cheklanib qolmaydi. Inson omili ham muhim rol o'ynaydi. Foydalanuvchilarning raqamli savodxonligi, xavfsizlik qoidalariga rioya qilishi va mas'uliyatli xatti-harakati zamonaviy texnologiyalarni xavfsiz qo'llashning muhim shartidir. Shu sababli ta'lim muassasalari va tashkilotlarda kiberxavfsizlik madaniyatini shakllantirish, treninglar va tushuntirish ishlarini olib borish zarur.



Zamonaviy texnologiyalar va kiberxavfsizlik o'zaro uzviy bog'liq tushunchalar bo'lib, ularni alohida ko'rib chiqish mumkin emas. Har qanday yangi texnologiya

joriy etilayotganda uning kiberxavfsizlikka ta'siri oldindan baholanishi lozim. "Xavfsizlikni dizayn bosqichidan boshlab ta'minlash" tamoyili zamonaviy axborot tizimlari uchun muhim yondashuv hisoblanadi.

IV. AMALIY MASHG‘ULOT MATERIALLARI

AMALIY MASHG‘ULOT MAZMUNI

1-amaliy mashg‘ulot: Ta’lim jarayonini raqamli texnologiyalar asosida optimallashtirish. Pandas, NumPy, Flask, Django kabi kutubxonalar va freymvorklardan ta’limiy vazifalarda foydalanish. (2 SOAT)

Ta’lim jarayonini raqamli texnologiyalar asosida optimallashtirishning nazariy asoslari. NumPy kutubxonasidan ta’limiy vazifalarda foydalanish imkoniyatlari. Pandas kutubxonasi orqali ta’lim jarayonini tahlil qilish va optimallashtirish. Flask freymvorki asosida yengil ta’limiy veb-illovalar yaratish. Django freymvorki asosida kompleks ta’lim platformalarini ishlab chiqish. Kutubxonalar va freymvorklarni integratsiyalashgan holda qo‘llash. Ta’limda raqamli yechimlarni joriy etish muammolari va ularni hal etish yo‘llari

Flask — Python tilidagi yengil (micro) web-freymvork bo‘lib, kichik va o‘rta hajmdagi ta’limiy veb-illovalarni tez yaratish uchun juda qulay. U “keragicha minimal” tamoyiliga asoslanadi: ya’ni dasturchi faqat zarur funksiyalarni qo‘shadi, ortiqcha murakkablik bo‘lmaydi. Shu sababli Flask ta’lim jarayonida ham o‘qituvchilar, ham talaba-o‘quvchilar uchun amaliy loyihalar yaratishda yaxshi tanlov hisoblanadi. Quyida Flask asosida yengil ta’limiy veb-illovalar yaratishning mohiyati, nazariy asoslari, amaliy qo‘llanish yo‘nalishlari va misollar bilan yoritiladi.

Ta’limda veb-illovalardan foydalanishning asosiy maqsadi — o‘quv jarayonini raqamlashtirish, o‘quvchilar faolligini oshirish, topshiriqlarni qulay tarqatish va natijalarni tez tahlil qilishdir. Masalan, oddiy test platforma, davomatni belgilash sahifasi, uy vazifasi topshirish shakli, o‘quv materiallarini joylashtirish, savol-javob (FAQ) yoki mini-LMS kabi tizimlar aynan yengil veb-ilovalar formatida juda tez ishga tushadi. Flask bu kabi loyihalarda server qismi (backend) vazifasini bajaradi: foydalanuvchi brauzer orqali so‘rov yuboradi, Flask uni qabul qiladi, kerakli ishlov beradi va javob qaytaradi.

Flaskning nazariy asosini HTTP protokoli, “request-response” (so‘rov-javob) modeli va marshrutlash (routing) tushunchalari tashkil etadi. Veb-ilovada har bir manzil (URL) ma’lum bir funksiyaga bog‘lanadi. Masalan, / bosh sahifa, /quiz test sahifasi, /result natijalar sahifasi bo‘lishi mumkin. Flaskda “route decorator” orqali URL va funksiyalar bog‘lanadi. Shu yondashuv o‘quvchilar uchun ham juda tushunarli: ular veb-dasturlashning amaliy mexanizmini tez anglab oladilar.

Flask ta’limda ayniqsa “tez prototiplash” (rapid prototyping) uchun qulay. O‘qituvchi yoki talaba murakkab tizimni boshidan qurib o‘tirmaydi, avval minimal

ishlaydigan versiyani (MVP) yaratadi: masalan, 10 ta savolli test, natijani chiqarish, oddiy baholash. Keyin esa bosqichma-bosqich imkoniyatlarni kengaytiradi: foydalanuvchi ro'yxatdan o'tishi, natijalar bazaga yozilishi, statistik tahlil, admin sahifasi va hokazo. Bu uslub ta'lim metodikasiga ham mos: o'quvchi "kichikdan kattaga" tamoyili asosida loyiha yaratishni o'rganadi.

Yengil ta'limiy veb-ilovalarning eng keng tarqalgan turlaridan biri — mini-test tizimidir. Bunda o'quvchi savollarga javob beradi, tizim esa avtomatik tekshiradi va natijani ko'rsatadi. Flask yordamida buni juda oddiy amalga oshirish mumkin. Masalan, savollarni Python lug'ati yoki JSON ko'rinishida saqlab, forma orqali javoblarni qabul qilish va baholash mumkin. Bu usul o'quv jarayonida baholashni tezlashtiradi, o'qituvchi vaqtini tejaydi, o'quvchi esa natijani darhol ko'rib, xatolarini tahlil qiladi.

Flaskning yana bir kuchli jihati — Jinja2 shablon tizimi (template engine). Bu orqali HTML sahifalarni dinamik yaratish mumkin: masalan, har bir o'quvchi uchun alohida natijalar, turli mavzu bo'yicha testlar, topshiriqlar ro'yxati, ball jadvali. Template yondashuvi o'quvchilarga frontend va backend aloqasini tushuntirishda juda qulay: Python tomonida ma'lumot tayyorlanadi, HTML tomonida esa chiroyli ko'rinishda chiqariladi. Masalan, o'quvchining ismi, to'plagan balli, to'g'ri va noto'g'ri javoblar soni sahifada avtomatik ko'rsatiladi.

Ta'limiy ilovalarda muhim masalalardan biri — ma'lumotlarni saqlash. Flaskning o'zi "micro" bo'lgani uchun, bazani o'zingiz tanlaysiz: oddiy loyihada SQLite, kengaytirilgan loyihada PostgreSQL yoki MySQL. Odatda, Flask + SQLAlchemy kombinatsiyasi ishlatiladi. Bu orqali o'quvchilar "ma'lumotlar bazasi + web" integratsiyasini amalda o'rganadilar: o'quvchi login qiladi, test topshiradi, natija bazaga yoziladi, o'qituvchi esa admin sahifada ko'radi. Bu real hayotdagi LMS va ERP tizimlariga o'xshash tajriba beradi.

Flask orqali ta'lim jarayonini optimallashtirishning yana bir yo'li — davomat va monitoring tizimlari yaratishdir. Masalan, har bir darsda o'quvchilar telefon yoki kompyuter orqali QR kod orqali sahifaga kirib, o'zini belgilaydi. Tizim vaqtni qayd etadi va Excel ko'rinishida hisobot chiqaradi. Bu o'qituvchining qog'ozbozligini kamaytiradi va aniq statistika beradi. Bunday loyihalarda Flask foydalanuvchi formasidan ma'lumot olib, bazaga yozish va natijani ko'rsatish vazifasini bajaradi.

Flask asosida yaratiladigan yana bir yengil, lekin foydali ta'limiy loyiha — "uy vazifasi topshirish" sahifasidir. O'quvchi topshiriq matnini yoki faylini yuklaydi, tizim esa ularni tartib bilan saqlaydi. O'qituvchi esa ro'yxatdan ko'rib, baho qo'yadi. Agar bunga oddiy autentifikatsiya (login/parol) qo'shilsa, tizim ancha tartibli bo'ladi. Bu turdagi ilovalar ayniqsa kichik kurslar, to'garaklar yoki bitta fan doirasida juda qulay.

Ta'limiy veb-illovalar yaratishda xavfsizlik va etik masalalar ham nazariy jihatdan muhim. Masalan, o'quvchilar ma'lumotlarini himoyalash, parolni ochiq saqlamaslik, formalarni CSRF hujumlaridan himoya qilish, ruqsatsiz kirishni cheklash, test savollarini oshkor bo'lishini oldini olish. Flask ekotizimida bu masalalar uchun tayyor yechimlar bor: Flask-Login, Flask-WTF kabi kengaytmalar yordamida autentifikatsiya va forma xavfsizligini kuchaytirish mumkin. Ta'lim jarayonida bunday masalalarni ko'tarish o'quvchilarda raqamli madaniyat va kiberxavfsizlik tushunchalarini ham shakllantiradi.

Flaskning ta'limdagi yana bir katta afzalligi — u API (masalan, JSON qaytaruvchi servis) yaratish uchun ham juda qulay. Bu degani, siz Flaskda “backend” yozib, mobil ilova yoki Telegram bot bilan bog'lashingiz mumkin. Masalan, o'quvchi botga sinfini yuboradi, bot Flask API'dan jadvalni olib qaytaradi. Bu siz ilgari qilgan “jadval Excel'dan olinadi” kabi g'oyalar bilan juda mos. Shunday integratsiya ta'lim jarayonini yanada avtomatlashtiradi va o'quvchilarni zamonaviy IT yechimlarga yaqinlashtiradi.

Quyida **Flask asosida bitta yengil ta'limiy loyihaga** misol keltiramiz: “**Mini Quiz (Test) ilovasi**”.

Loyiha imkoniyatlari

- / — bosh sahifa (testga kirish)
- /quiz — savollar sahifasi (radio button)
- /result — natijani chiqarish (ball, to'g'ri/noto'g'ri)
- Savollar vaqtincha Python ichida saqlanadi (keyin DB/JSONga oson ko'chadi)

Papka tuzilishi

```
miniquiz/  
  app.py  
  templates/  
    index.html  
    quiz.html  
    result.html
```

1) app.py

```
from flask import Flask, render_template, request  
app = Flask(__name__)  
# Savollar bazasi (oddiy ko'rinishda)  
QUESTIONS = [  
    {  
        "id": 1,  
        "question": "Python tilida massivga o'xshash  
tuzilma qaysi?",  
        "options": ["list", "int", "float", "bool"],  
        "answer": "list"  
    },  
    {
```

```

        "id": 2,
        "question": "NumPy kutubxonasi ning asosiy obyekt
turi nima?",
        "options": ["ndarray", "DataFrame", "Series",
"dict"],
        "answer": "ndarray"
    },
    {
        "id": 3,
        "question": "Flask odatda qaysi turdagi
framework?",
        "options": ["micro", "desktop", "game",
"mobile"],
        "answer": "micro"
    }
]

```

```
@app.route("/")
```

```
def index():
```

```
    return render_template("index.html")
```

```
@app.route("/quiz", methods=["GET", "POST"])
```

```
def quiz():
```

```
    if request.method == "GET":
```

```
        return render_template("quiz.html",
```

```
questions=QUESTIONS)
```

```
    # POST: javoblarni tekshiramiz
```

```
    correct = 0
```

```
    total = len(QUESTIONS)
```

```
    details = []
```

```
    for q in QUESTIONS:
```

```
        user_answer = request.form.get(f"q{q['id']}")
```

```
        is_correct = (user_answer == q["answer"])
```

```
        if is_correct:
```

```
            correct += 1
```

```
    details.append({
```

```
        "question": q["question"],
```

```
        "your_answer": user_answer,
```

```
        "correct_answer": q["answer"],
```

```
        "is_correct": is_correct
```

```
    })
```

```

score_percent = round((correct / total) * 100, 2)

return render_template(
    "result.html",
    correct=correct,
    total=total,
    score_percent=score_percent,
    details=details
)

```

```

if __name__ == "__main__":
    app.run(debug=True)

```

2) templates/index.html

```

<!DOCTYPE html>
<html lang="uz">
<head>
  <meta charset="UTF-8" />
  <title>MiniQuiz</title>
</head>
<body>
  <h2>MiniQuiz – Flask asosida test</h2>
  <p>Testni boshlash uchun quyidagi tugmani bosing.</p>
  <a href="/quiz">✔ Testni boshlash</a>
</body>
</html>

```

3) templates/quiz.html

```

<!DOCTYPE html>
<html lang="uz">
<head>
  <meta charset="UTF-8" />
  <title>Test</title>
</head>
<body>
  <h2>Test savollari</h2>

  <form method="POST">
    {% for q in questions %}
      <div style="margin-bottom: 16px;">
        <p><b>{{ loop.index }}. {{ q.question }}</b></p>

        {% for opt in q.options %}
          <label>
            <input type="radio" name="q{{ q.id }}"
value="{{ opt }}" required>
            {{ opt }}

```

```

        </label><br>
        {% endfor %}
    </div>
    <hr>
    {% endfor %}

    <button type="submit">Natijani ko`rish</button>
</form>
</body>
</html>

```

4) templates/result.html

```

<!DOCTYPE html>
<html lang="uz">
<head>
    <meta charset="UTF-8" />
    <title>Natija</title>
</head>
<body>
    <h2>Natija</h2>

    <p><b>To`g`ri:</b> {{ correct }} / {{ total }}</p>
    <p><b>Foiz:</b> {{ score_percent }}%</p>

    <h3>Tafsilotlar</h3>
    {% for d in details %}
        <div style="margin-bottom: 14px;">
            <p><b>Savol:</b> {{ d.question }}</p>
            <p><b>Sizning javob:</b> {{ d.your_answer }}</p>
            <p><b>To`g`ri javob:</b> {{ d.correct_answer }}</p>
            <p>
                <b>Holat:</b>
                {% if d.is_correct %}
                    ✓ To`g`ri
                {% else %}
                    ✗ Noto`g`ri
                {% endif %}
            </p>
        </div>
    <hr>
    {% endfor %}

    <a href="/quiz">🔄 Qayta ishlash</a>
</body>
</html>

```

Ishga tushirish tartibi

4. Papkani yarating: `miniquiz/`
5. Ichiga `app.py` va `templates/` papkasini qo‘ying
6. Terminalda:

```
pip install flask  
python app.py
```

5. Brauzerda quyidagi havolani oching: <http://127.0.0.1:5000/>

Zamonaviy ta‘lim tizimida raqamli texnologiyalarning keng joriy etilishi ta‘lim jarayonini boshqarish, tashkil etish va baholash usullarini tubdan o‘zgartirmoqda. An‘anaviy auditoriya va qog‘ozga asoslangan ta‘lim modellaridan farqli ravishda, raqamli platformalar o‘quvchilar faolligini oshirish, o‘qituvchi faoliyatini optimallashtirish hamda ta‘lim sifatini oshirish imkonini beradi. Shu jarayonda **kompleks ta‘lim platformalari (Learning Management System – LMS)** alohida ahamiyat kasb etadi.

Python dasturlash tilining kuchli va ommabop freymvorklaridan biri bo‘lgan **Django** murakkab, kengaytiriladigan va xavfsiz ta‘lim platformalarini ishlab chiqish uchun samarali vosita hisoblanadi. Ushbu bobda Django freymvorki asosida kompleks ta‘lim platformalarini yaratishning nazariy va amaliy asoslari yoritiladi.

Kompleks ta‘lim platformasi — bu o‘qituvchi, talaba va ta‘lim muassasasi ehtiyojlarini qamrab oluvchi, o‘quv jarayonining barcha bosqichlarini raqamli muhitda boshqarishga xizmat qiluvchi axborot tizimidir. Bunday platformalar quyidagi funksiyalarni o‘z ichiga oladi:

- foydalanuvchilarni boshqarish;
- kurslar va fanlarni tashkil etish;
- o‘quv materiallarini joylashtirish;
- topshiriqlar va testlar orqali baholash;
- natijalarni monitoring qilish va tahlil qilish;
- aloqa va bildirishnomalarni ta‘minlash.

Django asosida yaratilgan platformalar yuqoridagi vazifalarni yagona tizim doirasida amalga oshirish imkonini beradi.

Django — bu “batteries included” tamoyiliga asoslangan to‘liq funksional web-freymvork bo‘lib, u murakkab tizimlarni tez va sifatli ishlab chiqish imkonini beradi. Ta‘lim platformalari uchun Django quyidagi afzalliklarga ega:

Birinchidan, Django‘da foydalanuvchilarni autentifikatsiya va avtorizatsiya qilish tizimi tayyor holatda mavjud. Bu ta‘lim platformalarida rollar (administrator, o‘qituvchi, talaba) asosida kirish huquqlarini belgilashni osonlashtiradi.

Ikkinchidan, Django‘ning obyektga yo‘naltirilgan ma‘lumotlar modeli (ORM) ma‘lumotlar bazasi bilan ishlashni soddalashtiradi. Kurslar, darslar, topshiriqlar, natijalar kabi ma‘lumotlar tuzilmali va izchil tarzda saqlanadi.

Uchinchidan, Django admin paneli o'qituvchi va administratorlar uchun qulay boshqaruv muhiti yaratadi. O'quv materiallarini qo'shish, talabalarni ro'yxatdan o'tkazish, baholarni tahrirlash kabi jarayonlar kod yozmasdan amalga oshiriladi.

Kompleks ta'lim platformasini ishlab chiqishda modulga asoslangan arxitektura muhim ahamiyatga ega. Django loyihasi odatda bir nechta mustaqil ilovalarga (apps) bo'linadi. Har bir ilova platformaning ma'lum funksional qismini bajaradi.

Asosiy modullar quyidagilardan iborat bo'lishi mumkin:

- foydalanuvchilar va rollar moduli;
- kurslar va fanlar moduli;
- o'quv kontenti moduli;
- topshiriqlar va baholash moduli;
- test va quiz moduli;
- monitoring va analitika moduli;
- bildirishnoma va aloqa moduli.

Bunday yondashuv platformani keyinchalik kengaytirish, texnik xizmat ko'rsatish va jamoaviy ishlab chiqishni yengillashtiradi.

Ta'lim platformasining asosiy subyektlari — bu foydalanuvchilardir. Django asosida foydalanuvchilar bilan ishlashda rollarga asoslangan yondashuv qo'llaniladi. Administrator tizimni umumiy boshqaradi, o'qituvchi kurslar va baholashni yuritadi, talaba esa o'quv materiallari bilan ishlaydi va topshiriqlarni bajaradi.

Bu model pedagogik jarayonda mas'uliyat va vakolatlarni aniq ajratishga yordam beradi. Talaba faqat o'z kurslari va natijalarini ko'ra oladi, o'qituvchi esa faqat o'z fanlari doirasida boshqaruvni amalga oshiradi.

Django asosidagi ta'lim platformalarida kurslar ierarxik tuzilishda tashkil etiladi. Har bir kurs bo'limlarga, bo'limlar esa alohida darslarga bo'linadi. Dars tarkibida matnli tushuntirish, video materiallar, fayllar va tashqi havolalar bo'lishi mumkin.

Bunday tuzilma didaktik jihatdan qulay bo'lib, o'quvchilarga materialni ketma-ket va tizimli o'zlashtirish imkonini beradi. O'qituvchi esa kursni istalgan vaqtda yangilashi yoki kengaytirishi mumkin.

Kompleks ta'lim platformalarining muhim qismi — bu baholash mexanizmlaridir. Django asosida topshiriqlarni yaratish, ularni belgilangan muddatda qabul qilish va baholash avtomatlashtiriladi. Talabalar o'z ishlanmalarini matn yoki fayl ko'rinishida topshiradilar, o'qituvchi esa baho va izoh qoldiradi.

Test va quiz modullari esa bilimlarni tezkor va obyektiv baholash imkonini beradi. Savollar banki, vaqt cheklovi, bir nechta urinishlar kabi mexanizmlar ta'lim sifatini oshiradi va akademik halollikni ta'minlashga xizmat qiladi.

Django asosidagi platformalarda barcha ta'limiy jarayonlar raqamli iz qoldiradi. Bu esa o'quvchilar faolligi, o'zlashtirish darajasi va rivojlanish dinamikasini tahlil

qilish imkonini beradi. Analitik modul orqali o'qituvchi qaysi mavzular qiyinligini, qaysi talabalar qo'shimcha yordamga muhtojligini aniqlay oladi.

Bunday yondashuv ta'lim jarayonini ma'lumotlarga asoslangan holda boshqarish imkonini yaratadi va ta'limni optimallashtirishga xizmat qiladi.

Ta'lim platformalarida axborot xavfsizligi va akademik halollik masalalari alohida e'tibor talab qiladi. Django freymvorki CSRF va XSS hujumlariga qarshi himoya mexanizmlariga ega bo'lib, foydalanuvchi ma'lumotlarini ishonchli saqlashni ta'minlaydi.

Shuningdek, test savollarini tasodifiylashtirish, vaqt cheklovlari va urinishlar sonini belgilash orqali baholash jarayonining adolatliligi oshiriladi.

Mini LMS (Django) loyihasini ishlab chiqishni ko'rib chiqamiz:

Loyiha g'oyasi quyidagicha

O'qituvchi kurs yaratadi, dars qo'shadi, topshiriq beradi. Talaba topshiriqni matn ko'rinishida topshiradi. Tizim topshirilgan ishlarni ro'yxatda ko'rsatadi.

Funksiyalar (minimal)

- Kurslar ro'yxati
- Kurs ichida darslar ro'yxati
- Darsga topshiriq biriktirish
- Talaba topshiriqni topshirishi (submission)
- O'qituvchi submissions ro'yxatini ko'rishi

1) Loyihani yaratish

Terminal:

```
pip install django
django-admin startproject eduplatform
cd eduplatform
python manage.py startapp lms
eduplatform/settings.py ichida:
```

```
INSTALLED_APPS = [
    # ...
    'lms',
]
```

2) Model (lms/models.py)

```
from django.db import models
```

```
class Course(models.Model):
    title = models.CharField(max_length=200)
    description = models.TextField(blank=True)

    def __str__(self):
```

```

        return self.title

class Lesson(models.Model):
    course = models.ForeignKey(Course,
on_delete=models.CASCADE, related_name="lessons")
    title = models.CharField(max_length=200)
    content = models.TextField()

    def __str__(self):
        return f"{self.course.title} - {self.title}"

class Assignment(models.Model):
    lesson = models.OneToOneField(Lesson,
on_delete=models.CASCADE, related_name="assignment")
    title = models.CharField(max_length=200)
    description = models.TextField()
    deadline = models.DateField(null=True,
blank=True)

    def __str__(self):
        return self.title

class Submission(models.Model):
    assignment = models.ForeignKey(Assignment,
on_delete=models.CASCADE, related_name="submissions")
    student_name = models.CharField(max_length=120)
    answer_text = models.TextField()
    submitted_at =
models.DateTimeField(auto_now_add=True)

    def __str__(self):
        return f"{self.student_name} -
{self.assignment.title}"

```

Migratsiya:

```
python manage.py makemigrations
python manage.py migrate
```

3) Admin panel (lms/admin.py)

```
from django.contrib import admin
```

```
from .models import Course, Lesson, Assignment,
Submission
```

```
admin.site.register(Course)
admin.site.register(Lesson)
admin.site.register(Assignment)
admin.site.register(Submission)
```

Admin yaratish:

```
python manage.py createsuperuser
```

4) View'lar (lms/views.py)

```
from django.shortcuts import render,
get_object_or_404, redirect
from .models import Course, Lesson, Assignment,
Submission

def course_list(request):
    courses = Course.objects.all()
    return render(request, "lms/course_list.html",
{"courses": courses})

def lesson_list(request, course_id):
    course = get_object_or_404(Course, id=course_id)
    return render(request, "lms/lesson_list.html",
{"course": course})

def lesson_detail(request, lesson_id):
    lesson = get_object_or_404(Lesson, id=lesson_id)
    assignment = getattr(lesson, "assignment", None)
    return render(request, "lms/lesson_detail.html",
{"lesson": lesson, "assignment": assignment})

def submit_assignment(request, assignment_id):
    assignment = get_object_or_404(Assignment,
id=assignment_id)

    if request.method == "POST":
        student_name =
request.POST.get("student_name")
```

```

        answer_text = request.POST.get("answer_text")

        Submission.objects.create(
            assignment=assignment,
            student_name=student_name,
            answer_text=answer_text
        )
        return redirect("submissions",
assignment_id=assignment.id)

    return render(request, "lms/submit.html",
{"assignment": assignment})

def submissions(request, assignment_id):
    assignment = get_object_or_404(Assignment,
id=assignment_id)
    subs = assignment.submissions.order_by("-
submitted_at")
    return render(request, "lms/submissions.html",
{"assignment": assignment, "subs": subs})

```

5) URL'lar

```

eduplatform/urls.py:
from django.contrib import admin
from django.urls import path, include

urlpatterns = [
    path('admin/', admin.site.urls),
    path('', include('lms.urls')),
]

lms/urls.py:
from django.urls import path
from . import views

urlpatterns = [
    path('', views.course_list, name="course_list"),
    path('course/<int:course_id>/',
views.lesson_list, name="lesson_list"),

```

```

        path('lesson/<int:lesson_id>/',
views.lesson_detail, name="lesson_detail"),
        path('assignment/<int:assignment_id>/submit/',
views.submit_assignment, name="submit_assignment"),

path('assignment/<int:assignment_id>/submissions/',
views.submissions, name="submissions"),
    ]

```

6) Template'lar (minimal)

```

lms/templates/lms/course_list.html
<h2>Kurslar</h2>
<ul>
    {% for c in courses %}
        <li><a href="/course/{{ c.id }}/">{{ c.title
}}</a></li>
    {% endfor %}
</ul>
lms/templates/lms/lesson_list.html
<h2>{{ course.title }} - Darslar</h2>
<ul>
    {% for l in course.lessons.all %}
        <li><a href="/lesson/{{ l.id }}/">{{ l.title
}}</a></li>
    {% endfor %}
</ul>
<a href="/">← Kurslarga qaytish</a>
lms/templates/lms/lesson_detail.html
<h2>{{ lesson.title }}</h2>
<p>{{ lesson.content }}</p>

{% if assignment %}
    <h3>Topshiriq: {{ assignment.title }}</h3>
    <p>{{ assignment.description }}</p>
    <a href="/assignment/{{ assignment.id
}}/submit/">✓ Topshirish</a> |
    <a href="/assignment/{{ assignment.id
}}/submissions/">■ Topshirilganlar</a>
{% else %}

```

```

    <p><i>Bu dars uchun topshiriq
biriktirilmagan.</i></p>
    {% endif %}
    lms/templates/lms/submit.html
    <h2>Topshiriqni topshirish: {{ assignment.title
}}</h2>

    <form method="POST">
        {% csrf_token %}
        <label>Ism:</label><br>
        <input type="text" name="student_name"
required><br><br>

        <label>Javob:</label><br>
        <textarea name="answer_text" rows="6"
required></textarea><br><br>

        <button type="submit">Yuborish</button>
    </form>
    lms/templates/lms/submissions.html
    <h2>Topshirilganlar: {{ assignment.title }}</h2>
    <ul>
        {% for s in subs %}
            <li>
                <b>{{ s.student_name }}</b> ({{ s.submitted_at
}})<br>
                {{ s.answer_text }}
            </li>
            <hr>
        {% empty %}
            <li>Hali topshirilgan ish yo`q.</li>
        {% endfor %}
    </ul>

    <a href="/lesson/{{ assignment.lesson.id }}/">←
Darsga qaytish</a>

```

7) Ishga tushirish

```
python manage.py runserver
```

Brauzer: <http://127.0.0.1:8000/>

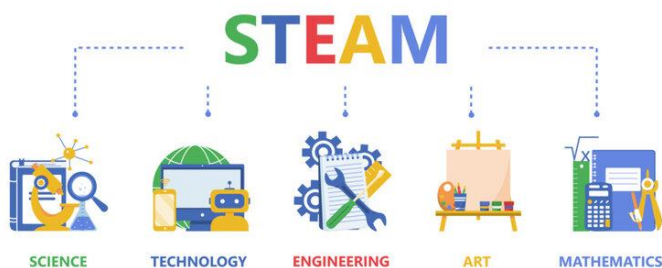
Admin: <http://127.0.0.1:8000/admin/> (kurs, dars, topshiriqni admin'dan qo'shasiz)

Ushbu kichik loyiha Django freymvorkining ta'limiy platformalar yaratishdagi asosiy imkoniyatlarini ko'rsatadi: ma'lumotlar modeli (Course-Lesson-Assignment-Submission), admin panel orqali kontent boshqarish, URL-View-Template bog'lanishi hamda foydalanuvchi bilan interaktiv ishlash. Loyiha keyingi bosqichlarda autentifikatsiya (talaba login), fayl yuklash, baholash, test moduli va analitik hisobotlar bilan kengaytirilishi mumkin.

2-amaliy mashg'ulot: STEAM yondashuvi asosida informatika fanini boshqa fanlar bilan integratsiyalash yo'llari. (2 soat)

STEAM yondashuvining nazariy asoslari. Informatika fanining STEAM tizimidagi ro'li. Informatika fanini tabiiy fanlar bilan integratsiyalash. Informatika fanini muhandislik va texnologiya bilan integratsiyalash. Informatika va san'at (Arts) integratsiyasi. STEAM loyihalarini loyihalash va amalga oshirish. STEAM darslarida raqamli vositalar va platformalar. STEAM yondashuvini joriy etishdagi muammolar va yechimlar.

Zamonaviy jamiyatning jadal rivojlanishi, raqamli texnologiyalarning keng joriy etilishi va mehnat bozorida talablarning o'zgarishi ta'lim tizimidan yangi yondashuvlarni talab qilmoqda. Bugungi kunda ta'lim faqat tayyor bilimlarni berish bilan cheklanib qolmasdan, balki o'quvchilarda muammoni hal qilish, tanqidiy fikrlash, ijodkorlik va fanlararo bog'liqlikni tushunish kompetensiyalarini shakllantirishi lozim. Ana shunday sharoitda STEAM yondashuvi (Science, Technology, Engineering, Art, Mathematics) zamonaviy ta'limning samarali va innovatsion modeli sifatida shakllandi.



STEAM yondashuvi fanlarni alohida-alohida o'qitishdan ko'ra, ularni yagona tizimda, o'zaro integratsiyalashgan holda o'qitishni nazarda tutadi. Ushbu yondashuvning asosiy g'oyasi bilimlarni real hayot muammolari bilan bog'lash,

nazariyani amaliyot orqali mustahkamlash va o'quvchini ta'lim jarayonining faol ishtirokchisiga aylantirishdan iborat.

STEAM yondashuvi dastlab STEM (Science, Technology, Engineering, Mathematics) modeli sifatida vujudga kelgan. STEM yondashuvi ilmiy-texnik fanlarga e'tiborni kuchaytirish va texnologik taraqqiyot uchun zarur kadrlarni tayyorlashga qaratilgan edi. Biroq vaqt o'tishi bilan bu modelning ayrim cheklovlari, xususan, ijodkorlik va estetik tafakkurning yetarli darajada rivojlanmasligi sezila boshladi. Natijada Art (san'at) komponenti qo'shib, STEAM yondashuvi shakllandi.

San'at komponentining kiritilishi STEAM yondashuviga insonparvarlik, dizayn tafakkuri va ijodiy yondashuvni olib kirdi. Bu esa texnik bilimlarni estetik, madaniy va ijodiy jihatlar bilan uyg'unlashtirish imkonini berdi. Shu tariqa STEAM yondashuvi nafaqat texnik mutaxassislarni, balki innovatsion fikrlovchi, moslashuvchan va ijodkor shaxslarni tayyorlashga yo'naltirilgan modelga aylandi.

STEAM yondashuvi bir nechta muhim pedagogik va falsafiy nazariyalarga tayanadi. Ulardan biri konstruktivizm nazariyasidir. Konstruktivizmga ko'ra, bilim o'qituvchi tomonidan tayyor holda berilmaydi, balki o'quvchi tomonidan faoliyat jarayonida mustaqil ravishda shakllantiriladi. STEAM muhitida o'quvchi tajriba o'tkazadi, kuzatadi, tahlil qiladi va xulosalar chiqaradi. Bu jarayon bilimning chuqur va barqaror o'zlashtirilishini ta'minlaydi.

Yana bir muhim asos — tajribaviy o'qitish nazariyasidir. Ushbu nazariyada o'rganish bevosita tajriba va amaliy faoliyat orqali amalga oshiriladi. STEAM yondashuvida laboratoriya ishlari, modellashtirish, tajribalar va loyihalar muhim o'rin tutadi. O'quvchilar nazariy bilimlarni real vaziyatlarda sinab ko'rish orqali ularni chuqurroq anglaydilar.

STEAM yondashuvining markaziy nazariy tamoyillaridan biri — fanlararo integratsiyadir. An'anaviy ta'limda fanlar ko'pincha alohida o'qitiladi, bu esa o'quvchilarda bilimlar o'rtasidagi bog'liqlikni tushunishda qiyinchilik tug'diradi. STEAM yondashuvida esa fanlar muammo va loyiha atrofida birlashtiriladi.

Masalan, bitta loyiha doirasida matematika hisob-kitoblari, fizika qonunlari, informatika orqali modellashtirish, muhandislik dizayni va san'at orqali estetik bezak birgalikda qo'llaniladi. Bu yondashuv o'quvchilarda kompleks fikrlash va tizimli tahlil ko'nikmalarini rivojlantiradi.

STEAM yondashuvi kompetensiyaviy ta'lim konsepsiyasiga to'liq mos keladi. Kompetensiyaviy yondashuvda asosiy e'tibor bilim miqdoriga emas, balki bilimlarni qo'llay olish qobiliyatiga qaratiladi. STEAM muhitida o'quvchilar quyidagi asosiy kompetensiyalarni egallaydilar:

- muammoni aniqlash va tahlil qilish;
- yechim ishlab chiqish va sinovdan o'tkazish;
- jamoada ishlash va muloqot qilish;

- texnologik va raqamli savodxonlik;
- ijodkorlik va dizayn tafakkuri.

Ushbu kompetensiyalar zamonaviy mehnat bozori talablariga mos bo'lib, o'quvchilarning kelajakdagi kasbiy faoliyati uchun mustahkam poydevor yaratadi.

STEAM yondashuvida loyiha asosida o'qitish (Project-Based Learning) muhim metodik asos hisoblanadi. Loyihalar real hayot muammolariga yo'naltirilgan bo'lib, o'quvchilarni mustaqil izlanishga, tadqiqot olib borishga va ijodiy yechimlar topishga undaydi.

Loyiha jarayonida o'quvchilar muammoni aniqlaydi, reja tuzadi, texnik va ilmiy yechimlarni ishlab chiqadi, natijani sinovdan o'tkazadi va taqdim etadi. Ushbu jarayon STEAM komponentlarining barchasini yagona faoliyatda birlashtiradi.

STEAM yondashuvi dizayn tafakkuri (design thinking) bilan uzviy bog'liq. Dizayn tafakkuri inson ehtiyojlariga asoslangan holda innovatsion yechimlar ishlab chiqishni nazarda tutadi. Ushbu yondashuvda muammoni chuqur tushunish, g'oyalar ishlab chiqish, prototip yaratish va takomillashtirish muhim bosqichlar hisoblanadi.

San'at komponenti aynan dizayn tafakkurini rivojlantirishga xizmat qiladi. Bu esa texnik yechimlarni nafaqat funksional, balki qulay va estetik jihatdan ham mukammal qilish imkonini beradi.

Nazariy jihatdan STEAM yondashuvi quyidagi muhim pedagogik natijalarga erishishni ta'minlaydi:

- ta'lim jarayonini real hayot bilan bog'laydi;
- o'quvchilarning o'qishga bo'lgan motivatsiyasini oshiradi;
- tanqidiy va ijodiy fikrlashni rivojlantiradi;
- fanlarga bo'lgan qiziqishni kuchaytiradi;
- raqamli va texnologik kompetensiyalarni shakllantiradi.

Bundan tashqari, STEAM yondashuvi ta'lim jarayonini moslashuvchan va ochiq tizimga aylantiradi, bu esa uni turli ta'lim bosqichlari va sharoitlarga moslashtirish imkonini beradi.

STEAM yondashuvi zamonaviy ta'limning chuqur nazariy asoslariga ega bo'lgan integratsiyalashgan pedagogik modeldir. Konstruktivizm, tajribaviy o'qitish, kompetensiyaviy va fanlararo integratsiya nazariyalari ushbu yondashuvning ilmiy poydevorini tashkil etadi. STEAM yondashuvi orqali ta'lim jarayoni mazmunan boyiydi, o'quvchilar esa bilimlarni real muammolarni hal etishda qo'llashga o'rganadilar. Shu bois STEAM yondashuvi zamonaviy ta'lim tizimining barqaror rivojlanishi va raqobatbardosh kadrlar tayyorlashda muhim ahamiyat kasb etadi.

Zamonaviy ta'lim tizimida STEAM (Science, Technology, Engineering, Art, Mathematics) yondashuvi fanlararo integratsiyaga asoslangan, real hayot muammolarini hal etishga yo'naltirilgan pedagogik model sifatida keng e'tirof

etilmoqda. Ushbu yondashuvda har bir fan alohida ahamiyatga ega bo'lsa-da, **informatika fani STEAM tizimining markaziy va birlashtiruvchi elementi** sifatida alohida o'rin tutadi. Informatika nafaqat texnologik vosita, balki fanlararo integratsiyani amalga oshiruvchi **intellektual va metodologik platforma** vazifasini bajaradi.

STEAM yondashuvining nazariy asoslarini ishlab chiqqan olimlar, xususan, **G. Yakman, R. Bybee, J. Wing, R. Capraro** va **S. Grover** o'z tadqiqotlarida informatika va hisoblash tafakkurining zamonaviy ta'limdagi hal qiluvchi rolini alohida ta'kidlaydilar. Ularning ilmiy qarashlariga ko'ra, informatika STEAM fanlarini o'zaro bog'lash, modellashtirish, tahlil qilish va loyihalash jarayonlarini raqamli muhitga olib chiqish imkonini beradi.

G. Yakman tomonidan ishlab chiqilgan STEAM konsepsiyasida fanlar alohida emas, balki **ijtimoiy-madaniy va texnologik muammolar** atrofida integratsiyalashgan holda qaraladi. Ushbu modelda informatika fanlararo integratsiyaning texnik asosi bo'lib xizmat qiladi. Chunki har qanday ilmiy, muhandislik yoki matematik g'oya bugungi kunda ko'pincha kompyuter modellashtiruvi, algoritmlash va dasturiy yechimlar orqali amalga oshiriladi.

Informatika:

- fanlararo ma'lumotlarni raqamli shaklda qayta ishlash;
- murakkab jarayonlarni modellashtirish;
- natijalarni vizuallashtirish;
- loyihalarni avtomatlashtirish

imkonini berib, STEAM tizimining yaxlit ishlashini ta'minlaydi.

J. Wing tomonidan ilgari surilgan **hisoblash tafakkuri (computational thinking)** tushunchasi informatikaning STEAM tizimidagi nazariy asosini tashkil etadi. Hisoblash tafakkuri muammoni algoritmik tahlil qilish, abstraksiyalash, modellashtirish va bosqichma-bosqich yechim ishlab chiqish qobiliyatini anglatadi.

STEAM yondashuvida:

•matematika → formal mantiq va hisoblash;

•tabiiy fanlar → jarayon va hodisalarni tahlil qilish;

•muhandislik → tizimli loyihalash;

•san'at → ijodiy yechim va dizayn

hisoblash tafakkuri orqali yagona intellektual modelga birlashadi.

Bu jarayonda informatika o'quvchilarda **algoritmik va tizimli fikrlashni** shakllantiruvchi asosiy fan bo'lib xizmat qiladi.

STEAM tizimida informatika matematika bilan eng yaqin integratsiyaga ega fanlardan biridir. R. Capraro va hamkorlari o'z tadqiqotlarida matematik tushunchalarni loyiha asosida o'qitishda dasturlash va modellashtirish muhim rol o'ynashini ta'kidlaydilar.

Informatika orqali:

- matematik formulalar algoritmga aylantiriladi;
- funksiyalar grafigi vizuallashtiriladi;
- statistik va ehtimollik masalalari real ma'lumotlar asosida tahlil qilinadi.

Bu esa matematik bilimlarni abstrakt holatdan amaliy va tushunarli shaklga olib chiqadi.

Fizika, kimyo va biologiya fanlarida ko'plab jarayonlar murakkab va dinamik xarakterga ega. Grover va Pea tadqiqotlarida ta'kidlanganidek, informatika tabiiy fanlarni o'qitishda **simulyatsiya va modellashtirish** vositasi sifatida muhim ahamiyatga ega.

Masalan:

- fizik harakatlarni kompyuter modeli orqali o'rganish;
- biologik jarayonlarning o'sish dinamikasini dasturiy tahlil qilish;
- kimyoviy reaksiyalarni virtual laboratoriyalarda sinash.

Bu integratsiya o'quvchilarda ilmiy tajriba va tadqiqot kompetensiyalarini rivojlantiradi.

Bybee va Capraro ishlarida STEAM yondashuvida muhandislik dizayni markaziy o'rin tutishi ta'kidlanadi. Informatika bu jarayonda muhandislik fikrlashining "raqamli tili" hisoblanadi.

Informatika muhandislik faoliyatida:

- texnik tizimlarni boshqarish algoritmlarini yaratish;
- robototexnika va avtomatlashtirish;
- sensorlar va mikroprotessorlar bilan ishlash

uchun asosiy vosita bo'lib xizmat qiladi.

Shu orqali o'quvchilar muhandislik yechimlarini nafaqat loyihalaydi, balki dasturiy jihatdan amalga oshiradilar.

Yakman va Land tomonidan STEAM modeliga san'at komponentining kiritilishi tasodifiy emas. Informatika san'at bilan integratsiyada **ijodiy texnologiyalar** asosini yaratadi.

Bunga quyidagilar kiradi:

- grafik dizayn va animatsiya;

- veb-dizayn;
- raqamli hikoyalar;
- interaktiv ilovalar yaratish.

Bu integratsiya texnik bilimlarni estetik va ijodiy yondashuv bilan uyg'unlashtirib, o'quvchilarda dizayn tafakkurini rivojlantiradi.

STEAM yondashuvining amaliy asosi — loyiha asosida o'qitishdir. Informatika loyiha faoliyatining barcha bosqichlarida ishtirok etadi:

- muammoni tahlil qilish;
- raqamli reja va model yaratish;
- texnik yechimni dasturlash;
- natijani taqdim etish.

Natijada informatika STEAM loyihalarining **boshqaruvchi va bog'lovchi mexanizmi** sifatida namoyon bo'ladi.

Yuqoridagi nazariy tahlillar shuni ko'rsatadiki, informatika STEAM tizimida:

- fanlararo integratsiyani ta'minlaydi;
- hisoblash va algoritmik tafakkurni shakllantiradi;
- raqamli va texnologik savodxonlikni rivojlantiradi;
- innovatsion va ijodiy fikrlashni qo'llab-quvvatlaydi.

Zamonaviy ta'lim tizimida informatika fani nafaqat mustaqil fan sifatida, balki boshqa fanlar bilan integratsiyalashgan holda o'qitiladigan muhim metodologik vosita sifatida qaralmoqda. Ayniqsa, **tabiiy fanlar** — fizika, kimyo, biologiya va geografiya bilan informatika fanining integratsiyasi ta'lim jarayonining ilmiylik darajasini oshirish, murakkab jarayonlarni tushunarli qilish va o'quvchilarning amaliy kompetensiyalarini rivojlantirishda katta ahamiyatga ega. Ushbu integratsiya STEAM yondashuvining muhim tarkibiy qismi bo'lib, fanlararo bog'liqlikni ta'minlaydi.

Tabiiy fanlar ko'plab abstrakt tushunchalar, murakkab jarayonlar va dinamik hodisalarni o'z ichiga oladi. Informatika esa ushbu jarayonlarni **modellashtirish, simulyatsiya qilish, hisoblash va vizuallashtirish** imkonini berib, tabiiy fanlarni o'rganishni samaraliroq qiladi. Natijada o'quvchilar nazariy bilimlarni real jarayonlar bilan bog'lab tushunish imkoniga ega bo'ladilar.

Fizika fanida harakat, kuch, energiya, tebranish, to'lqin va elektr hodisalari kabi jarayonlar matematik va mantiqiy modellar asosida tushuntiriladi. Informatika fani ushbu jarayonlarni **kompyuter modellarida aks ettirish** imkonini beradi.

Informatika va fizika integratsiyasi quyidagi yo'llar orqali amalga oshiriladi:

- harakat tenglamalarini dasturlash orqali modellashtirish;
- fizik jarayonlarni simulyatsiya qilish (masalan, erkin tushish, tebranishlar);
- grafik va animatsiyalar orqali hodisalarni vizuallashtirish;
- tajriba natijalarini raqamli tahlil qilish.

Masalan, o'quvchilar dasturlash muhiti yordamida jismning tezligi va yo'li vaqtga bog'liqligini hisoblab, grafik ko'rinishda tasvirlashlari mumkin. Bu jarayon fizika qonunlarini chuqurroq anglashga yordam beradi va informatika orqali ilmiy tafakkurni rivojlantiradi.

Kimyo fanida moddalarning tuzilishi, reaksiyalar tezligi va energiya almashinuvi kabi jarayonlar muhim o'rin tutadi. Informatika bu jarayonlarni **hisoblash va virtual tajribalar** orqali o'rganish imkonini beradi.

Integratsiya yo'llari quyidagilardan iborat:

- kimyoviy reaksiyalarni algoritmik modellashtirish;
- moddalarning molekulyar tuzilishini grafik modellar orqali ko'rsatish;
- tajriba natijalarini elektron jadval va dasturiy vositalar yordamida tahlil qilish;
- virtual laboratoriyalardan foydalanish.

Masalan, reaksiya tezligining haroratga bog'liqligini dasturiy modellashtirish orqali o'quvchilar kimyoviy jarayonlarning dinamikasini yaxshiroq tushunadilar. Bu yondashuv xavfli yoki qimmat tajribalarni virtual muhitda xavfsiz amalga oshirish imkonini beradi.

Biologiya fanida ko'plab jarayonlar vaqt davomida o'zgarib boruvchi va murakkab tizimlar bilan bog'liq. Informatika biologik jarayonlarni **simulyatsiya qilish va tahlil qilish** uchun qulay vosita hisoblanadi.

Integratsiya quyidagi shakllarda amalga oshiriladi:

- populyatsiya o'sishini matematik va dasturiy modellashtirish;
- biologik jarayonlarni (masalan, hujayra bo'linishi) animatsiyalar orqali tushuntirish;
- tajriba ma'lumotlarini raqamli qayta ishlash;
- ekologik tizimlarni modellashtirish.

Masalan, o'quvchilar populyatsiya sonining vaqt bo'yicha o'zgarishini dasturiy model yordamida tahlil qilib, muhit omillarining ta'sirini ko'rishlari mumkin. Bu biologik tafakkurni tizimli va mantiqiy asosda rivojlantiradi.

Geografiya fanida fazoviy ma'lumotlar, xaritalar va statistik axborotlar bilan ishlash muhim ahamiyatga ega. Informatika bu fan bilan integratsiyada **geografik axborot tizimlari (GIS)**, ma'lumotlar tahlili va vizuallashtirish vositasi sifatida xizmat qiladi.

Integratsiya quyidagicha amalga oshiriladi:

- raqamli xaritalar bilan ishlash;
- iqlim va demografik ma'lumotlarni tahlil qilish;
- statistik ma'lumotlarni grafik ko'rinishda tasvirlash;
- ekologik muammolarni raqamli modellashtirish.

Masalan, o'quvchilar iqlim o'zgarishiga oid ma'lumotlarni tahlil qilib, grafik va xaritalar orqali natijalarni ko'rsatishlari mumkin. Bu geografiya fanini informatika bilan uyg'unlashtirib, real muammolarni o'rganishga yordam beradi.

Informatika fanini tabiiy fanlar bilan integratsiyalash quyidagi pedagogik afzalliklarni beradi:

- o'quvchilarning fanlarga bo'lgan qiziqishini oshiradi;
- murakkab tushunchalarni vizual va tushunarli qiladi;
- ilmiy-tadqiqot kompetensiyalarini rivojlantiradi;
- hisoblash va algoritmik fikrlashni shakllantiradi;
- nazariy bilimlarni amaliy faoliyat bilan bog'laydi.

Bundan tashqari, bunday integratsiya STEAM yondashuvi doirasida o'quvchilarning **tanqidiy fikrlash, muammo yechish va jamoada ishlash** ko'nikmalarini ham rivojlantiradi.

Loyiha asosida o'qitish informatika va tabiiy fanlar integratsiyasining eng samarali shakllaridan biridir. Loyihalarda o'quvchilar real muammolarni hal qilish jarayonida bir nechta fan bilimlaridan foydalanadilar. Masalan:

- "Aqlli ob-havo stansiyasi" (fizika + informatika);
- "O'simlik o'sishini modellashtirish" (biologiya + informatika);
- "Kimyoviy reaksiyalar tezligini tahlil qilish" (kimyo + informatika).

Bu loyihalar o'quvchilarning mustaqil izlanish va ijodiy yondashuvini qo'llab-quvvatlaydi.

Zamonaviy ta'lim tizimida informatika fani muhandislik va texnologiya sohalari bilan uzviy bog'liq holda rivojlanmoqda. Bugungi kunda har qanday muhandislik yechimi va texnologik jarayon raqamli tizimlarsiz tasavvur etilmaydi. Shu sababli informatika fanini muhandislik va texnologiya bilan integratsiyalash ta'lim jarayonining tabiiy va zarur yo'nalishiga aylanmoqda. Ushbu integratsiya STEAM yondashuvining muhim tarkibiy qismi bo'lib, o'quvchilarda tizimli fikrlash, amaliy muammo yechish va texnologik savodxonlik kompetensiyalarini shakllantirishga xizmat qiladi.

Muhandislik faoliyati asosan muammoni aniqlash, texnik yechim ishlab chiqish, loyihalash va sinovdan o'tkazish jarayonlarini o'z ichiga oladi. Informatika bu jarayonlarning barchasida markaziy rol o'ynaydi. Avvalo, muammoni tahlil qilish va yechim variantlarini ishlab chiqishda algoritmik fikrlash va modellashtirish muhim ahamiyatga ega. Informatika fani orqali o'quvchilar texnik jarayonlarni matematik va mantiqiy modellar ko'rinishida ifodalashni, ularni dasturiy muhitda sinovdan o'tkazishni o'rganadilar. Bu esa muhandislik tafakkurining shakllanishiga bevosita ta'sir ko'rsatadi.

Texnologiya sohasi bilan integratsiyada informatika raqamli vositalar va avtomatlashtirilgan tizimlar orqali namoyon bo'ladi. Zamonaviy texnologik

jarayonlar ko‘pincha dasturiy boshqaruv asosida ishlaydi. Sensorlar, mikroprotsessorlar, robototexnika qurilmalari va aqlli tizimlar informatikasiz faoliyat yurita olmaydi. Ta‘lim jarayonida informatika fanini texnologiya bilan bog‘lab o‘qitish o‘quvchilarga ushbu qurilmalar qanday ishlashi, ma‘lumotlar qanday yig‘ilishi va qayta ishlanishi hamda texnologik jarayonlar qanday boshqarilishini tushunishga yordam beradi.

Informatika va muhandislik integratsiyasi loyihaviy faoliyat orqali ayniqsa samarali amalga oshiriladi. Loyiha jarayonida o‘quvchilar texnik muammoni aniqlaydi, yechim algoritmini ishlab chiqadi, dasturiy ta‘minot yaratadi va uni real yoki virtual qurilmada sinovdan o‘tkazadi. Masalan, avtomatik yoritish tizimi, aqlli issiqxona yoki robot modeli yaratishda o‘quvchilar muhandislik dizayni va texnologik komponentlarni informatika orqali boshqarishni o‘rganadilar. Bu jarayon nazariy bilimlarning amaliyot bilan uyg‘unlashuvini ta‘minlaydi.

Texnologiya bilan integratsiyada informatika nafaqat boshqaruv, balki tahlil va optimallashtirish vositasi sifatida ham xizmat qiladi. Ma‘lumotlarni yig‘ish va tahlil qilish, jarayon samaradorligini baholash va takomillashtirish informatikaning muhim vazifalaridan biridir. Muhandislik tizimlarida yuzaga keladigan nosozliklarni aniqlash, resurslardan samarali foydalanish va energiya tejamkorligini ta‘minlash kabi masalalar dasturiy tahlil va modellashtirish orqali hal etiladi. Bu esa o‘quvchilarda muhandislik muammolariga ilmiy va tizimli yondashuvni shakllantiradi.

Informatika fanining muhandislik va texnologiya bilan integratsiyasi o‘quvchilarning ijodiy va innovatsion fikrlashini ham rivojlantiradi. Texnik yechimlarni faqat funksional emas, balki qulay va samarali qilish zarurati dizayn tafakkurini talab etadi. Informatika bu jarayonda raqamli dizayn, simulyatsiya va prototiplash imkoniyatlarini taqdim etadi. Natijada o‘quvchilar texnologik mahsulotni loyihalashdan tortib, uni dasturiy boshqarishgacha bo‘lgan barcha bosqichlarni tushunib yetadilar.

Pedagogik nuqtayi nazardan, informatika fanini muhandislik va texnologiya bilan integratsiyalash ta‘lim jarayonini faol va amaliy yo‘naltirilgan holga keltiradi. O‘quvchilar passiv tinglovchi emas, balki faol ijodkor va tadqiqotchi sifatida ishtirok etadilar. Bu esa ularning o‘qishga bo‘lgan qiziqishini oshiradi, mustaqil fikrlash va jamoada ishlash ko‘nikmalarini rivojlantiradi. Shuningdek, bunday integratsiya o‘quvchilarni kelajakdagi kasbiy faoliyatga tayyorlashda muhim ahamiyat kasb etadi, chunki zamonaviy muhandislik va texnologiya sohalarida informatika bilimlari asosiy talab hisoblanadi.

Informatika fanini muhandislik va texnologiya bilan integratsiyalashda dasturiy vositalar majmui, avvalo, **loyihalash – modellashtirish – dasturlash – sinov**

zanjirini to'liq qamrab olishi lozim. Shu sababli bunday integratsiyada bitta emas, balki bir-birini to'ldiruvchi bir nechta dasturiy muhitlar majmuasi qo'llaniladi.

Muhandislik dizayni va texnik modellashtirish bosqichida **AutoCAD** va **Fusion 360** kabi CAD dasturlari muhim o'rin tutadi. Ushbu dasturlar yordamida o'quvchilar mexanik qismlar, qurilmalar va konstruksiyalarni raqamli muhitda loyihalaydilar. Informatika fanida bu jarayon fayl formatlari, koordinata tizimi, geometrik obyektlar va parametrik modellashtirish tushunchalari bilan bog'lanadi. Texnologiya va muhandislik fanlarida esa konstruksion yechimlar va texnik chizmalar ustida ishlanadi.

Modellashtirish va simulyatsiya jarayonida **MATLAB** va **Simulink** dasturiy majmuasi keng qo'llaniladi. Bu vositalar texnologik jarayonlar, boshqaruv tizimlari va signal oqimlarini modellashtirish imkonini beradi. Informatika fanida algoritmlar va matematik modellar bilan ishlash mustahkamlanadi, muhandislik fanlarida esa real texnik tizimlarning ishlash mexanizmlari tushuntiriladi. Ayniqsa, avtomatika va robototexnika yo'nalishlarida ushbu dasturlar integratsiya uchun juda qulay.

Amaliy texnologik integratsiyada **Arduino IDE** va **Micro:bit** muhitlari muhim ahamiyatga ega. Ushbu dasturlar orqali o'quvchilar mikrocontrollerlar bilan ishlash, sensorlardan ma'lumot olish va qurilmalarni dasturiy boshqarishni o'rganadilar. Informatika fanida bu jarayon algoritmlash va dasturlash asoslari bilan bog'lanadi, texnologiya fanida esa real qurilmalar va avtomatlashtirish jarayonlari bilan mustahkamlanadi. Bu majmua "aqli qurilmalar" va "IoT" tushunchalarini o'qitishda ayniqsa samaralidir.

Robototexnika bilan bog'liq integratsiyada **Tinkercad Circuits** va **Proteus** dasturlari muhim vosita hisoblanadi. Ushbu muhitlar yordamida elektron sxemalarni virtual tarzda yig'ish, sinovdan o'tkazish va tahlil qilish mumkin. Informatika fanida mantiqiy bloklar va dastur oqimi tushuntiriladi, muhandislik va texnologiyada esa elektronika asoslari va qurilma ishlashi o'rganiladi. Bu yondashuv xavfsiz va tejamkor o'quv muhiti yaratadi.

Texnologik jarayonlarni tahlil qilish va optimallashtirishda **Python** dasturlash tili va uning kutubxonalari muhim rol o'ynaydi. Python yordamida o'quvchilar sensorlardan kelayotgan ma'lumotlarni qayta ishlash, grafiklar chizish va texnologik jarayon samaradorligini baholashni o'rganadilar. Informatika bu yerda asosiy dasturiy vosita bo'lsa, muhandislik va texnologiya real jarayonlar bilan ta'minlaydi. Bu integratsiya sanoatga yaqin masalalarni yechishda juda samarali.

Ishlab chiqarish texnologiyalari bilan integratsiyada **CNC** va **3D-printer** dasturiy ta'minotlari ham muhim o'rin tutadi. Masalan, **Cura** yoki **PrusaSlicer** dasturlari yordamida 3D modellar ishlab chiqarishga tayyorlanadi. Informatika fanida raqamli model va kod tushunchalari mustahkamlanadi, texnologiya fanida esa ishlab

chiqarish jarayoni va materiallar bilan ishlash o'rganiladi. Bu jarayon muhandislik tafakkurini real natija bilan bog'laydi.

Zamonaviy ta'lim tizimida informatika va san'at integratsiyasi STEAM yondashuvining eng muhim va o'ziga xos jihatlaridan biri hisoblanadi. Ushbu integratsiya texnik bilimlar bilan ijodiy tafakkurni uyg'unlashtirishga xizmat qilib, o'quvchilarda estetik did, dizayn tafakkuri va raqamli ijodkorlikni rivojlantiradi. Informatika fani san'at bilan birlashganda faqat texnik vosita bo'lib qolmay, balki ijodiy g'oyalarni amalga oshirish uchun kuchli platformaga aylanadi. Ayniqsa, zamonaviy dasturiy vositalar bu jarayonda hal qiluvchi rol o'ynaydi.

Informatika va san'at integratsiyasining asosiy yo'nalishlaridan biri raqamli grafika va dizayn bilan bog'liq. Grafik dizayn dasturlari orqali o'quvchilar ranglar uyg'unligi, shakl va kompozitsiya qonuniyatlarini amaliy faoliyatda o'rganadilar. Raqamli muhitda ishlash jarayoni san'at asarini yaratish bilan birga, fayllar bilan ishlash, formatlar, qatlamlar va effektlar tushunchalarini o'zlashtirishga ham imkon beradi. Bu jarayon informatika fanining amaliy qo'llanilishini ko'rsatib, san'atni raqamli texnologiyalar bilan boyitadi.

Animatsiya va multimediali loyihalar informatika va san'at integratsiyasining yana bir muhim yo'nalishidir. Animatsiya yaratish jarayonida harakat, vaqt va ketma-ketlik tushunchalari algoritmik fikrlash bilan uyg'unlashadi. O'quvchilar har bir kadrning joylashuvi va o'zgarishini rejalashtirib, ijodiy g'oyani dasturiy muhitda jonlantiradilar. Bu jarayonda san'atga xos tasviriy ifoda bilan informatikaning mantiqiy tuzilmasi bir-birini to'ldiradi.

Raqamli hikoyalar va interaktiv kontent yaratish ham ushbu integratsiyaning samarali shakllaridan biridir. Dasturiy vositalar yordamida matn, rasm, ovoz va videoni birlashtirish o'quvchilarning ijodiy salohiyatini kengaytiradi. Shu bilan birga, ular foydalanuvchi bilan o'zaro aloqani tashkil etish, interaktiv elementlar qo'shish va ssenariy asosida ishlash ko'nikmalarini ham egallaydilar. Bu esa san'atni faqat tomosha qilinadigan emas, balki faol ishtirok etiladigan jarayonga aylantiradi.

Veb-dizayn va veb-texnologiyalar informatika va san'at integratsiyasida muhim o'rin tutadi. Veb-sahifa yaratish jarayonida estetik dizayn, ranglar, shriftlar va joylashuv san'at elementlari sifatida namoyon bo'ladi, informatika esa ularni texnik jihatdan amalga oshirish vositasini taqdim etadi. Dasturlash tillari va veb-muhitlar yordamida o'quvchilar dizayn g'oyalarni funksional va interaktiv shaklda ifodalashni o'rganadilar. Bu integratsiya raqamli madaniyat va dizayn tafakkurini rivojlantirishga xizmat qiladi.

So'nggi yillarda algoritmik san'at va generativ dizayn yo'nalishlari ham informatika va san'at integratsiyasining yorqin namunalaridan biri sifatida rivojlanmoqda. Algoritmilar asosida yaratilgan tasvirlar va vizual kompozitsiyalar

san'at va matematika, informatika o'rtasidagi chuqur bog'liqlikni namoyon etadi. Bu jarayonda o'quvchilar tasvirlarni avtomatik yaratish, parametrlarni o'zgartirish va natijalarni tahlil qilish orqali ijodiy eksperimentlar o'tkazadilar. Natijada san'at jarayoni ilmiy-tadqiqot faoliyatiga yaqinlashadi.

Musiqqa va ovoz bilan ishlash ham informatika va san'at integratsiyasining muhim yo'nalishidir. Raqamli musiqa yaratish dasturlari yordamida o'quvchilar tovushlarni tahrirlash, effektlar qo'llash va kompozitsiyalar yaratishni o'rganadilar. Bu jarayon algoritmik ketma-ketlik, vaqt va chastota tushunchalarini tushunishga yordam beradi. Informatika orqali musiqa san'ati raqamli shaklda yangi imkoniyatlarga ega bo'ladi.

Pedagogik jihatdan informatika va san'at integratsiyasi o'quvchilarning o'qishga bo'lgan motivatsiyasini oshiradi va ularni ijodiy faoliyatga jalb etadi. Texnik va estetik bilimlarning uyg'unlashuvi o'quvchilarda erkin fikrlash, yangicha yondashuv va innovatsion qarorlar qabul qilish ko'nikmalarini rivojlantiradi. Shu bilan birga, raqamli vositalar bilan ishlash jarayonida o'quvchilar zamonaviy texnologiyalarni ongli va samarali qo'llashni o'rganadilar.

Raqamli grafika va dizayn yo'nalishida eng ko'p qo'llaniladigan dasturlardan biri **Adobe Photoshop** hisoblanadi. Ushbu dastur tasvirlarni tahrirlash, kollajlar yaratish, ranglar bilan ishlash va vizual effektlar qo'llash imkonini beradi. Informatika fanida Photoshop orqali o'quvchilar piksel tushunchasi, rang modellarini (RGB, CMYK) va qatlamlar bilan ishlashni o'rganadilar. Shu bilan birga, san'at darslarida estetik did va kompozitsiya tushunchalari rivojlanadi. Vektorli grafika bilan ishlash uchun **Adobe Illustrator** yoki **CorelDRAW** dasturlari qulay bo'lib, ular orqali logotiplar, belgilar va dizayn elementlari yaratiladi. Bu dasturlar algoritmik fikrlashni ham talab qiladi, chunki har bir obyekt aniq parametrlar asosida boshqariladi.

Boshlang'ich va o'rta sinflar uchun juda qulay muhitlardan biri **Scratch** hisoblanadi. Scratch yordamida o'quvchilar animatsiya, interaktiv hikoya va o'yinlar yaratadilar. Bu jarayonda dasturlashning asosiy tushunchalari san'at bilan uyg'unlashadi. O'quvchi harakatni rejalashtiradi, rang va shakllarni tanlaydi hamda ijodiy g'oyasini algoritm orqali ifodalaydi. Shuningdek, **Processing** muhiti algoritmik san'at va generativ dizayn uchun juda mos bo'lib, u orqali kod yordamida vizual kompozitsiyalar yaratiladi. Bu dastur san'at va matematika, informatika o'rtasidagi bog'liqlikni yaqqol ko'rsatadi.

Animatsiya va multimedia yo'nalishida **Adobe Animate** va **Blender** dasturlaridan foydalanish mumkin. Adobe Animate ikki o'lchamli animatsiyalar va interaktiv sahnalar yaratish imkonini bersa, Blender uch o'lchamli modellashtirish, animatsiya va vizual effektlar yaratish uchun kuchli bepul dastur hisoblanadi. Blender orqali o'quvchilar fazoviy tafakkur, obyektlar geometriyasi va render jarayonlari

bilan tanishadilar. Informatika fanida bu jarayonlar algoritmik va texnik jihatdan tahlil qilinadi, san'at fanida esa tasviriy ifoda va dizayn asoslari o'rganiladi.

Veb-dizayn yo'nalishida **HTML, CSS va JavaScript** asosidagi muhitlar muhim ahamiyatga ega. Informatika darslarida veb-sahifa tuzilmasi, dizayn va interaktivlik tushuntirilsa, san'at fanida ranglar uyg'unligi, shrift tanlash va vizual kompozitsiya ustida ishlanadi. Dizaynni qulay va vizual tarzda yaratish uchun **Figma** yoki **Canva** kabi onlayn platformalar ham keng qo'llaniladi. Bu vositalar jamoaviy ishlash va dizayn tafakkurini rivojlantirishga xizmat qiladi.

Musiqqa va ovoz bilan ishlashda **Audacity** va **FL Studio** kabi dasturlar foydali hisoblanadi. Audacity orqali ovozlarni yozish va tahrirlash, FL Studio orqali esa raqamli musiqa yaratish mumkin. Bu jarayon informatika nuqtayi nazaridan raqamli signal, formatlar va fayl tuzilmasini o'rganishga, san'at nuqtayi nazaridan esa musiqiy ijodkorlikni rivojlantirishga yordam beradi.

Raqamli rasm chizish va illustratsiya yo'nalishida **Krita** va **GIMP** dasturlari ham samarali hisoblanadi. Bu bepul dasturlar yordamida o'quvchilar grafik planshetlarda rasm chizish, rang va shakllar bilan ishlashni o'rganadilar. Shu bilan birga, informatika fanida rastr grafika, fayl formatlari va rasmni raqamli qayta ishlash tushunchalari mustahkamlanadi.

Zamonaviy ta'lim tizimida STEAM (Science, Technology, Engineering, Art, Mathematics) yondashuvi fanlararo integratsiyaga asoslangan, amaliy faoliyat orqali bilimlarni shakllantirishni ko'zda tutuvchi samarali pedagogik model sifatida qaralmoqda. Ushbu yondashuvning asosiy amaliy ko'rinishi STEAM loyihalari orqali namoyon bo'ladi. STEAM loyihalari o'quvchilarni real hayot muammolarini hal etishga jalb etib, ularning ilmiy, texnologik, muhandislik, ijodiy va matematik kompetensiyalarini birgalikda rivojlantiradi. Shu bois STEAM loyihalarini to'g'ri loyihalash va samarali amalga oshirish ta'lim jarayonining muhim vazifalaridan biri hisoblanadi.

STEAM loyihalarining mohiyati shundaki, loyiha bitta fan doirasida emas, balki bir nechta fanlar kesishmasida tashkil etiladi. Bu jarayonda o'quvchi faqat bilim oluvchi emas, balki tadqiqotchi, muhandis va ijodkor sifatida ishtirok etadi. Loyiha faoliyati o'quvchilarda muammoni aniqlash, tahlil qilish, rejalashtirish, yechim ishlab chiqish va natijani baholash ko'nikmalarini shakllantiradi.

STEAM loyihalarini loyihalash konstruktivistik va kompetensiyaviy yondashuvlarga asoslanadi. Konstruktivizmga ko'ra, bilim tayyor holda berilmaydi, balki o'quvchi tomonidan faoliyat jarayonida mustaqil ravishda yaratiladi. STEAM loyihalarida o'quvchi tajriba o'tkazadi, kuzatadi, xulosa chiqaradi va o'z bilimini real vaziyatda sinab ko'radi. Kompetensiyaviy yondashuv esa bilimlarni amaliy qo'llashga yo'naltirilgan bo'lib, loyiha faoliyati ushbu yondashuvni to'liq ta'minlaydi.

Loyihalash jarayonida fanlararo integratsiya muhim ahamiyatga ega. Har bir loyiha aniq muammo atrofida shakllanadi va mazkur muammoni hal etishda turli fanlar bilimlari birlashtiriladi. Masalan, ekologik muammoga bag'ishlangan STEAM loyihasida biologiya, fizika, informatika, muhandislik va dizayn elementlari birgalikda qo'llaniladi.

1. STEAM loyihasini rejalashtirish bosqichi

STEAM loyihasini amalga oshirishdan avval puxta rejalashtirish talab etiladi. Rejalashtirish jarayonida loyiha maqsadi, vazifalari, kutilayotgan natijalari va baholash mezonlari aniqlanadi. Maqsad aniq va o'lchab bo'ladigan bo'lishi muhim, chunki u loyiha faoliyatining yo'nalishini belgilaydi.

Loyiha mavzusi o'quvchilarning yosh xususiyatlari, qiziqishlari va bilim darajasiga mos tanlanishi lozim. Mavzu real hayot bilan bog'liq bo'lsa, o'quvchilarning motivatsiyasi yanada oshadi. Shuningdek, loyiha davomida qaysi STEAM komponentlari ishtirok etishi aniqlanib, har bir fan uchun vazifalar belgilanadi. Bu esa loyiha jarayonida fanlararo muvozanatni ta'minlaydi.

2. Muammoni aniqlash va tadqiqot bosqichi

STEAM loyihasining boshlang'ich bosqichi muammoni aniqlash va uni tadqiq qilishdan iborat. O'quvchilar muammo haqida dastlabki ma'lumotlarni to'playdi, mavjud yechimlarni tahlil qiladi va tadqiqot savollarini shakllantiradi. Bu bosqich ilmiy izlanishning asosiy elementlarini o'z ichiga oladi.

Tadqiqot jarayonida o'quvchilar axborot manbalari bilan ishlashni, tajribalar o'tkazishni va kuzatish natijalarini qayd etishni o'rganadilar. Informatika fanining roli aynan shu bosqichda yaqqol namoyon bo'lib, ma'lumotlarni yig'ish, saqlash va tahlil qilishda dasturiy vositalardan foydalaniladi. Bu esa o'quvchilarda raqamli savodxonlikni rivojlantiradi.

3. Loyihalash va yechim ishlab chiqish bosqichi

Muammo tahlil qilingach, loyiha yechimini ishlab chiqish bosqichi boshlanadi. Ushbu bosqichda o'quvchilar muhandislik va dizayn tafakkuridan foydalanib, texnik va ijodiy yechimlarni ishlab chiqadilar. Yechimni modellashtirish, chizmalar tayyorlash va dastlabki prototiplarni yaratish aynan shu bosqichda amalga oshiriladi.

Informatika fanining integratsiyasi orqali o'quvchilar algoritmlar tuzadi, modellarni dasturiy muhitda sinovdan o'tkazadi va natijalarni tahlil qiladi. San'at komponenti esa dizayn, estetik ko'rinish va foydalanuvchiga qulaylikni ta'minlashga xizmat qiladi. Bu jarayon STEAM loyihasining ijodiy va texnik jihatlarini uyg'unlashtiradi.

4. Amaliyot va sinov bosqichi

STEAM loyihalarining muhim bosqichlaridan biri — ishlab chiqilgan yechimni amaliyotda sinovdan o'tkazishdir. Bu bosqichda o'quvchilar yaratgan model yoki

qurilmani real sharoitda tekshiradi, uning ishlash samaradorligini baholaydi va kamchiliklarni aniqlaydi. Sinov natijalari asosida loyiha takomillashtiriladi.

Sinov jarayoni o'quvchilarda tanqidiy fikrlash va muammoni qayta ko'rib chiqish ko'nikmalarini rivojlantiradi. Xatolarni aniqlash va ularni bartaraf etish STEAM loyihalarining ajralmas qismi bo'lib, bu jarayon o'quvchilarga muvaffaqiyatsizlikni o'rganish imkoniyati sifatida qabul qilishni o'rgatadi.

5. Natijalarni taqdim etish va baholash

STEAM loyihasining yakuniy bosqichi natijalarni taqdim etish va baholashdan iborat. O'quvchilar o'z loyihalarini taqdimot, hisobot, poster yoki amaliy namoyish orqali himoya qiladilar. Bu jarayon muloqot va jamoa oldida chiqish kompetensiyalarini rivojlantiradi.

Baholash jarayoni faqat yakuniy natijani emas, balki loyiha davomida o'quvchilarning faoliyatini ham qamrab olishi lozim. Baholash mezonlari loyiha maqsadlariga mos ravishda oldindan belgilanadi va shaffof bo'lishi muhim. Bu o'quvchilarning mas'uliyatini oshiradi va adolatli baholashni ta'minlaydi.

STEAM loyihalarini loyihalash va amalga oshirish ta'lim jarayoniga ko'plab ijobiy ta'sir ko'rsatadi. O'quvchilar nazariy bilimlarni amaliy faoliyatda qo'llashni o'rganadi, fanlararo bog'liqlikni anglaydi va real muammolarni hal qilish tajribasiga ega bo'ladi. Bunday loyihalar ijodiy va tanqidiy fikrlashni rivojlantirib, o'quvchilarning o'ziga bo'lgan ishonchini oshiradi.

Bundan tashqari, STEAM loyihalari o'quvchilarning kelajak kasbiy yo'nalishini aniqlashga yordam beradi. Muhandislik, texnologiya va axborot texnologiyalari sohalariga qiziqish loyihaviy faoliyat jarayonida shakllanadi. Bu esa ta'limning uzluksizligi va amaliy yo'naltirilganligini ta'minlaydi.

Quyida STEAM yondashuvi asosida loyiha ssenariysi bosqichma-bosqich, o'quv-amaliy mashg'ulot uchun keltirib o'tamiz.

STEAM loyihasi ssenariysi: "O'quvchilar faoliyatini tahlil qiluvchi raqamli monitoring tizimi"

1-bosqich. Muammoni aniqlash

Maktab yoki ta'lim muassasasida o'quvchilarning darsdagi faolligi, topshiriqlarni bajarish darajasi va natijalarini doimiy kuzatib borish qiyin. Ma'lumotlar ko'pincha qog'ozda yoki tartibsiz saqlanadi, bu esa tahlil qilishni murakkablashtiradi.

Muammo quyidagicha qo'yiladi: **O'quvchilar faoliyatini raqamli vositalar yordamida qanday qilib yig'ish, saqlash va tahlil qilish mumkin?**

2-bosqich. Loyiha maqsadi

Loyihaning asosiy maqsadi – o‘quvchilar faoliyatini qayd etuvchi va tahlil qiluvchi **oddiy raqamli axborot tizimini** yaratishdir.

Natijada:

- ma’lumotlar elektron shaklda saqlanadi;
- grafik va diagrammalar orqali tahlil qilinadi;
- o‘qituvchi uchun qaror qabul qilish osonlashadi.

3-bosqich. STEAM komponentlarining roli

Ma’lumotlar bazasi, dasturiy interfeys va tahlil mexanizmi yaratiladi. Matematika statistik tahlil va foiz hisoblashda qo‘llaniladi. Texnologiya kompyuter va dasturiy muhit bilan bog‘liq. Muhandislik tizimni loyihalash va strukturalashda namoyon bo‘ladi. San’at esa interfeys dizayni va vizual ko‘rinishda ishtirok etadi.

4-bosqich. Talablarni aniqlash va rejalashtirish

O‘quvchilar dastlab tizim qanday ishlashi kerakligini aniqlaydilar. Qaysi ma’lumotlar kiritiladi, qanday saqlanadi va qanday ko‘rinishda chiqariladi, degan savollarga javob topiladi. Masalan, o‘quvchi ismi, sinfi, topshiriq nomi va bahosi kiritiladi. Reja asosida tizim tuzilmasi belgilanadi.

Bu bosqichda informatika fanidan **algoritmash va tizimli fikrlash** faol qo‘llaniladi.

5-bosqich. Dasturiy muhitni tanlash

Loyiha **Python** dasturlash tili asosida amalga oshiriladi. Ma’lumotlarni saqlash uchun oddiy fayl yoki jadval shakli tanlanadi. Ma’lumotlarni tahlil qilish uchun dasturiy vositalardan foydalaniladi. Grafik interfeys yoki natijalarni ko‘rsatish uchun oddiy vizual oynalar yoki diagrammalar ishlatiladi.

6-bosqich. Dasturlash va ma’lumotlar bilan ishlash

O‘quvchilar dasturda quyidagi jarayonlarni amalga oshiradilar: ma’lumot kiritish, saqlash, tahrirlash va o‘chirish. Shuningdek, o‘rtacha bahoni hisoblash, faol o‘quvchilarni aniqlash va natijalarni grafik ko‘rinishda chiqarish funksiyalari yaratiladi.

Bu jarayonda:

- o‘zgaruvchilar;
- shart operatorlari;
- sikllar;
- ma’lumotlar tuzilmalari amaliy tarzda mustahkamlanadi.

7-bosqich. Tahlil va vizuallashtirish

Yig'ilgan ma'lumotlar grafiklar va diagrammalar orqali tahlil qilinadi. O'quvchilar qaysi fan yoki topshiriqda faollik yuqori yoki past ekanini ko'rish imkoniga ega bo'ladilar.

Bu bosqich informatika va matematika integratsiyasini yaqqol namoyon etadi, chunki sonli ma'lumotlar tahlil qilinadi va xulosalar chiqariladi.

8-bosqich. Sinov va takomillashtirish

Dastur sinovdan o'tkaziladi. Agar xatoliklar aniqlansa, kod qayta ko'rib chiqiladi. O'quvchilar dastur ishlash tezligi, qulayligi va aniqligini baholaydilar.

Bu bosqich dasturchilar uchun muhim bo'lgan **debugging** va optimallashtirish ko'nikmalarini rivojlantiradi.

9-bosqich. Natijalarni taqdim etish

Loyiha yakunida o'quvchilar yaratilgan dastur qanday ishlashini namoyish etadilar. Taqdimotda algoritm, dastur tuzilmasi va olingan natijalar tushuntiriladi. Grafiklar va jadval asosida xulosalar chiqariladi.

Bu bosqich muloqot va axborotni aniq ifodalash kompetensiyalarini rivojlantiradi.

10-bosqich. Baholash va refleksiya

Loyiha baholanadi. O'quvchilar o'z ishlari haqida fikr bildiradilar, qaysi bilimlar mustahkamlanganini va nimalarni yanada rivojlantirish mumkinligini muhokama qiladilar. Refleksiya o'rganilgan bilimlarning ahamiyatini anglashga yordam beradi.

STEAM yondashuvi asosida tashkil etilgan darslarda raqamli vositalar va platformalar ta'lim jarayonining ajralmas qismiga aylanib bormoqda. Chunki STEAM darslarining asosiy maqsadi fanlararo integratsiyani ta'minlash, nazariy bilimlarni amaliy faoliyat bilan bog'lash va o'quvchilarni real muammolarni hal etishga yo'naltirishdan iborat. Raqamli vositalar aynan shu vazifalarni samarali amalga oshirish imkonini beradi. Ular o'quvchilarni faol o'rganishga jalb etadi, murakkab jarayonlarni vizual va tushunarli shaklda namoyon etadi hamda ijodiy va muhandislik tafakkurini rivojlantiradi.

STEAM darslarida raqamli vositalar, avvalo, o'quv jarayonini modellashtirish va simulyatsiya qilish uchun xizmat qiladi. Tabiiy fanlar, muhandislik va matematika bilan bog'liq mavzular ko'pincha murakkab va abstrakt bo'lgani sababli, ularni faqat og'zaki tushuntirish orqali o'zlashtirish qiyin kechadi. Raqamli simulyatsiyalar va virtual muhitlar yordamida esa o'quvchilar jarayonlarni bevosita kuzatish, ularni o'zgartirish va natijalarni tahlil qilish imkoniyatiga ega bo'ladilar. Bu esa tajribaviy o'qitish va konstruktivistik yondashuvni ta'minlaydi.

Informatika fanining STEAM darslaridagi roli raqamli platformalar orqali yanada kuchayadi. Dasturlash muhitlari, vizual kodlash platformalari va ma'lumotlar bilan ishlash vositalari o'quvchilarda hisoblash va algoritmik tafakkurni rivojlantiradi. Shu bilan birga, ushbu vositalar matematika, fizika, biologiya va muhandislik fanlari bilan integratsiyalashgan holda qo'llanilib, fanlararo bog'liqlikni kuchaytiradi. Masalan, dasturlash asosida yaratilgan model yoki simulyatsiya bir vaqtning o'zida ilmiy, texnik va matematik tushunchalarni o'zida mujassam etadi.

STEAM darslarida raqamli platformalar o'quvchilarni loyiha asosida ishlashga undaydi. Loyiha faoliyati jarayonida o'quvchilar ma'lumotlarni yig'ish, saqlash, tahlil qilish va natijalarni taqdim etish uchun turli raqamli vositalardan foydalanadilar. Onlayn platformalar jamoaviy ishlashni, fikr almashishni va loyiha jarayonini bosqichma-bosqich tashkil etishni osonlashtiradi. Bu esa o'quvchilarda hamkorlik, mas'uliyat va muloqot kompetensiyalarini rivojlantiradi.

Raqamli vositalarning yana bir muhim jihati — vizuallashtirish imkoniyatidir. Grafiklar, diagrammalar, animatsiyalar va interaktiv modellar STEAM darslarida keng qo'llaniladi. Ular yordamida o'quvchilar sonli va mantiqiy bog'liqliklarni tezroq anglaydilar, jarayonlarni yaxlit tizim sifatida ko'ra boshlaydilar. Ayniqsa, san'at komponenti bilan integratsiyada raqamli dizayn va vizual platformalar o'quvchilarning ijodiy salohiyatini ochib beradi.

STEAM darslarida raqamli platformalar baholash va monitoring jarayonlarini ham takomillashtiradi. Onlayn testlar, interaktiv topshiriqlar va elektron portfoliolar orqali o'quvchilarning rivojlanish darajasi muntazam kuzatib boriladi. Bu esa baholash jarayonini shaffof va obyektiv qiladi, o'quvchilarga esa o'z natijalarini tahlil qilish va takomillashtirish imkonini beradi. Raqamli baholash vositalari kompetensiyaviy yondashuvni amalga oshirishda muhim ahamiyatga ega.

Pedagogik nuqtayi nazardan, raqamli vositalar STEAM darslarida o'qituvchining rolini ham o'zgartiradi. O'qituvchi endi faqat bilim beruvchi emas, balki o'quv jarayonini tashkil etuvchi, yo'naltiruvchi va maslahat beruvchi sifatida faoliyat yuritadi. Raqamli platformalar orqali o'qituvchi dars mazmunini moslashuvchan tarzda rejalashtiradi, o'quvchilarning individual xususiyatlarini hisobga olgan holda topshiriqlarni tashkil etadi.

STEAM darslari uchun raqamli platformalar jadvali

Platforma / Dastur	STEAM komponenti	Asosiy vazifasi	Qaysi fanlar bilan mos	Ta'limdagi qo'llanishi
Scratch	S, T, E, A	Vizual dasturlash, animatsiya	Informatika, matematika, san'at	Algoritmik fikrlash, interaktiv loyihalar
Python (IDLE, Jupyter)	S, T, E, M	Dasturlash, tahlil, modellashtirish	Informatika, fizika,	Hisoblash, grafiklar, modellar

			matematika	
Arduino IDE	T, E	Qurilmalarni dasturlash	Informatika, texnologiya, fizika	Robototexnika, avtomatlashtirish
Tinkercad (Circuits/3D)	E, T, A	3D model va elektron sxema	Texnologiya, informatika, muhandislik	Virtual prototiplash
Proteus	E, T	Elektron sxemalar simulyatsiyasi	Fizika, informatika	Qurilma sinovi, xavfsiz tajriba
GeoGebra	M, S	Matematik modellashtirish	Matematika, fizika	Grafiklar, geometrik modellar
PhET Simulations	S	Virtual tajribalar	Fizika, kimyo, biologiya	Tajriba, simulyatsiya
Blender	A, E, T	3D modellashtirish, animatsiya	San'at, informatika	Dizayn, vizual loyihalar
Canva	A, T	Dizayn, taqdimot	San'at, informatika	Poster, loyiha taqdimoti
Figma	A, T	UI/UX dizayn	Informatika, san'at	Interfeys dizayni
Google Classroom	T	Kurs va loyiha boshqaruvi	Barcha fanlar	Topshiriq, baholash
Kahoot / Quizizz	T	Interaktiv baholash	Barcha fanlar	Test, refleksiya
Padlet	A, T	G'oya va loyiha almashish	STEAM loyihalar	Jamoaviy ish
Excel / Google Sheets	M, T	Ma'lumotlar tahlili	Matematika, informatika	Jadval, grafik
Power BI (soddalashtirilgan)	T, M	Vizual tahlil	Informatika	Dashboard, natijalar

Ushbu jadvaldan foydalanishda muhim jihat shundaki, **bitta STEAM darsida 2–3 ta platforma yetarli**. Masalan, informatika markazda bo'lgan darsda Python + GeoGebra + Canva kombinatsiyasi juda samarali ishlaydi. Robototexnika yo'nalishida esa Arduino IDE + Tinkercad + Google Classroom uyg'unligi tavsiya etiladi.

Boshlang'ich bosqichlar uchun vizual va soddaroq muhitlar, yuqori bosqichlar uchun esa dasturlash va modellashtirishga yo'naltirilgan platformalar tanlanadi. Muhimi, platforma emas, **pedagogik maqsad yetakchi bo'lishi** lozim.

STEAM yondashuvi zamonaviy ta'lim tizimida fanlararo integratsiyani ta'minlovchi, o'quvchilarning amaliy va ijodiy kompetensiyalarini rivojlantirishga qaratilgan samarali model hisoblanadi. Biroq ushbu yondashuvni ta'lim amaliyotiga joriy etish jarayonida bir qator muammolar yuzaga kelmoqda. Ushbu muammolar tashkiliy, metodik, texnologik va kadrlar bilan bog'liq jihatlarida namoyon bo'lib, STEAM ta'limining to'liq va sifatli amalga oshirilishiga ma'lum darajada to'sqinlik qiladi.

STEAM yondashuvini joriy etishdagi asosiy muammolardan biri o'qituvchilarning tayyorgarlik darajasi bilan bog'liq. An'anaviy fanlarga yo'naltirilgan ta'lim tizimida ishlagan o'qituvchilar uchun fanlararo integratsiya, loyiha asosida o'qitish va raqamli vositalardan kompleks foydalanish dastlab murakkab bo'lib tuyuladi. Ko'plab pedagoglar o'z fanini yaxshi bilsa-da, boshqa fanlar bilan integratsiyalashgan holda dars tashkil etishda metodik qiyinchiliklarga duch keladi. Ushbu muammoni hal etish uchun o'qituvchilarni STEAM metodikasi bo'yicha muntazam malaka oshirish kurslari, amaliy seminarlar va tajriba almashish platformalari bilan ta'minlash zarur.

Yana bir muhim muammo moddiy-texnik bazaning yetarli emasligi bilan bog'liq. STEAM darslari ko'pincha laboratoriya jihozlari, kompyuterlar, sensorlar, robototexnika to'plamlari va maxsus dasturiy vositalarni talab etadi. Ko'plab ta'lim muassasalarida esa bunday resurslar cheklangan yoki umuman mavjud emas. Ushbu holatda STEAM yondashuvini to'liq joriy etish qiyinlashadi. Muammoni bartaraf etish uchun bosqichma-bosqich joriy etish strategiyasini tanlash, virtual laboratoriyalar, onlayn simulyatsiyalar va bepul dasturiy platformalardan foydalanish samarali yechim bo'la oladi.

STEAM ta'limida o'quv dasturlarining moslashuvchan emasligi ham muammolardan biri hisoblanadi. An'anaviy o'quv rejalari fanlarni qat'iy chegaralangan holda o'qitishga yo'naltirilgan bo'lib, fanlararo loyihalar uchun yetarli vaqt va imkoniyat bermaydi. Natijada STEAM loyihalari qo'shimcha yuk sifatida qabul qilinadi. Ushbu muammoni hal qilish uchun o'quv dasturlarini qayta ko'rib chiqish, integratsiyalashgan mavzular va loyihaviy mashg'ulotlar uchun alohida vaqt ajratish lozim.

Baholash tizimi ham STEAM yondashuvini joriy etishda muhim muammo sifatida namoyon bo'ladi. An'anaviy baholash ko'proq bilimni tekshirishga qaratilgan bo'lsa, STEAM ta'limi jarayonni, ijodkorlikni, jamoaviy ishni va muammo yechish qobiliyatini baholashni talab etadi. Mos baholash mezonlarining yo'qligi o'qituvchilarni qiyin vaziyatga soladi. Ushbu muammoni hal etish uchun kompetensiyaga asoslangan baholash, loyiha natijalari, portfoliolar va refleksiya elementlaridan foydalanish tavsiya etiladi.

O'quvchilarning ham STEAM yondashuviga moslashuvi ma'lum qiyinchiliklarni keltirib chiqaradi. Mustaqil izlanish, jamoada ishlash va mas'uliyatni bo'lishish barcha o'quvchilar uchun oson kechavermaydi. Ayniqsa, passiv o'qitish muhitiga o'rgangan o'quvchilar faol loyihaviy ishga moslashishda qiynalishi mumkin. Bunday holatda o'qituvchi tomonidan bosqichma-bosqich yo'naltirish, kichik loyihalardan boshlash va motivatsiyani oshirish muhim ahamiyatga ega.

STEAM yondashuvini joriy etishda tashkiliy va boshqaruv muammolari ham uchraydi. Fan o'qituvchilari o'rtasida hamkorlikning yetarli emasligi, umumiy

rejalashtirishning yoʻqligi integratsiyani susaytiradi. Ushbu muammoni hal etish uchun fanlararo metodik guruhlar tashkil etish, qoʻshma rejalashtirish va umumiy loyiha faoliyatini yoʻlga qoʻyish zarur.

Umuman olganda, STEAM yondashuvini joriy etish murakkab, biroq istiqbolli jarayon boʻlib, mavjud muammolarni tizimli yondashuv, metodik qoʻllab-quvvatlash va raqamli vositalardan oqilona foydalanish orqali samarali hal etish mumkin.

3-amaliy mashgʻulot: 3D-modellashtirish, robototexnika, simulyatsiya hamda VR/AR texnologiyalari. (2 soat)

3D-modellashtirish texnologiyalari. Robototexnika asoslari. Simulyatsiya texnologiyalari. VR (Virtual Reality) texnologiyalari. AR (Augmented Reality) texnologiyalari. Texnologiyalarni integratsiyalashgan holda qoʻllash. Taʼlimda joriy etish muammolari va yechimlar

Zamonaviy taʼlim tizimida raqamli texnologiyalarning jadal rivojlanishi STEAM yondashuvining amaliy imkoniyatlarini sezilarli darajada kengaytirdi. Xususan, 3D-modellashtirish, robototexnika, simulyatsiya hamda virtual va kengaytirilgan reallik texnologiyalari taʼlim jarayonini vizual, interaktiv va amaliy yoʻnaltirilgan holga keltirmoqda. Ushbu texnologiyalar oʻquvchilarga murakkab ilmiy va muhandislik tushunchalarini real tajribaga yaqin muhitda oʻrganish imkonini beradi, shu bilan birga informatika fanining markaziy rolini yanada kuchaytiradi.

3D-modellashtirish texnologiyalari STEAM taʼlimida muhandislik va dizayn tafakkurini shakllantirishda muhim ahamiyatga ega. Uch oʻlchamli modellar orqali obyektlarning tuzilishi, oʻlchamlari va fazoviy joylashuvi aniq tasvirlanadi. Oʻquvchilar raqamli muhitda modellar yaratish jarayonida geometriya, matematika va informatika bilimlarini birgalikda qoʻllaydilar. 3D-modellashtirish nafaqat texnik obyektlarni loyihalash, balki biologik tuzilmalar, arxitektura elementlari va ilmiy modellarni tasvirlash imkonini ham beradi. Bu jarayon oʻquvchilarda fazoviy fikrlash va muhandislik dizayniga asoslangan yondashuvni rivojlantiradi.

Robototexnika STEAM yondashuvining eng yorqin va amaliy yoʻnalishlaridan biri hisoblanadi. Robototexnika orqali informatika, fizika, matematika va texnologiya fanlari yagona faoliyatda birlashadi. Robotni yaratish va dasturlash jarayonida oʻquvchilar algoritmik fikrlash, sensorlardan maʼlumot olish va qurilmalarni boshqarish asoslarini oʻrganadilar. Robototexnika loyihalari real muammolarni texnologik yechimlar orqali hal etishga yoʻnaltirilgan boʻlib, oʻquvchilarda muhandislik tafakkuri va ijodkorlikni shakllantiradi. Shu bilan birga, jamoaviy ishlash va muammoga tizimli yondashuv koʻnikmalari rivojlanadi.

Simulyatsiya texnologiyalari STEAM taʼlimida murakkab jarayonlarni xavfsiz va tejamkor tarzda oʻrganish imkonini beradi. Fizik, kimyoviy yoki biologik

jarayonlarni real sharoitda sinovdan o'tkazish har doim ham mumkin bo'lmagan hollarda, simulyatsiyalar muhim muqobil yechim bo'lib xizmat qiladi. Kompyuter simulyatsiyalari orqali o'quvchilar jarayon parametrlarini o'zgartirib, natijalarni kuzatish va tahlil qilish imkoniyatiga ega bo'ladilar. Bu esa tajribaviy o'qitish tamoyillarini to'liq amalga oshirishga xizmat qiladi. Informatika fanida simulyatsiyalar algoritmlar, modellar va ma'lumotlar bilan ishlashni amaliy tarzda mustahkamlaydi.

Virtual reallik (VR) va kengaytirilgan reallik (AR) texnologiyalari ta'lim jarayoniga yangi o'lcham olib kirdi. VR texnologiyalari o'quvchini to'liq virtual muhitga olib kirib, ilmiy va muhandislik jarayonlarini bevosita "ichidan" ko'rish imkonini beradi. Masalan, murakkab mexanizmlarni virtual muhitda o'rganish yoki biologik tizimlar ichki tuzilishini kuzatish mumkin. AR texnologiyalari esa real muhitga raqamli obyektlar va axborotni qo'shish orqali o'rganishni yanada boyitadi. Ushbu texnologiyalar abstrakt tushunchalarni vizual va interaktiv shaklda taqdim etib, o'quvchilarning qiziqishini oshiradi va bilimni chuqurroq o'zlashtirishga yordam beradi.

STEAM yondashuvida ushbu texnologiyalarni qo'llash ta'lim jarayonini faqat nazariy bilimlar bilan cheklab qo'ymaydi, balki o'quvchilarni faol tadqiqotchi va yaratuvchiga aylantiradi. 3D-modellashtirish orqali loyiha yaratiladi, robototexnika yordamida u harakatga keltiriladi, simulyatsiya orqali sinovdan o'tkaziladi, VR/AR yordamida esa vizual va interaktiv tarzda namoyish etiladi. Bu ketma-ketlik STEAM loyihalarining mantiqiy va texnologik asosini tashkil etadi.

Pedagogik nuqtayi nazardan, 3D-modellashtirish, robototexnika, simulyatsiya hamda VR/AR texnologiyalari o'quvchilarning o'qishga bo'lgan motivatsiyasini oshiradi, ularni murakkab masalalarni mustaqil hal etishga undaydi va raqamli kompetensiyalarni rivojlantiradi. Ushbu texnologiyalar orqali o'quvchilar real hayotda qo'llaniladigan bilim va ko'nikmalarni egallab, zamonaviy texnologik jamiyat talablariga moslashadilar.

Shu tariqa, 3D-modellashtirish, robototexnika, simulyatsiya hamda VR/AR texnologiyalari STEAM ta'limining muhim tarkibiy qismlari bo'lib, informatika faniga tayangan holda fanlararo integratsiyani ta'minlaydi va ta'lim jarayonini innovatsion, samarali hamda amaliy yo'naltirilgan holga keltiradi.

Zamonaviy raqamli texnologiyalar rivoji bilan 3D-modellashtirish texnologiyalari muhandislik, sanoat, tibbiyot, arxitektura va ta'lim sohalarida keng qo'llanila boshladi. Ayniqsa, STEAM yondashuvi doirasida 3D-modellashtirish texnologiyalari o'quvchilarda fazoviy tafakkur, muhandislik dizayni va raqamli ijodkorlikni rivojlantirishda muhim ahamiyat kasb etmoqda. Ushbu texnologiyalar yordamida murakkab obyektlar va jarayonlar raqamli muhitda aniq va tushunarli

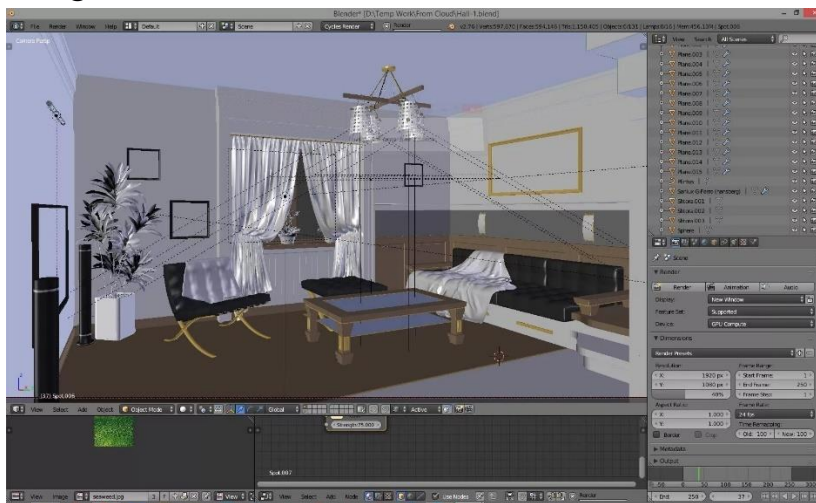
tarzda modellashtiriladi, bu esa nazariy bilimlarni amaliy faoliyat bilan uyg'unlashtirish imkonini beradi.

3D-modellashtirish deganda uch o'lchamli fazoda obyektlarning raqamli modellarini yaratish jarayoni tushuniladi. Bu jarayon informatika, matematika va muhandislik bilimlariga asoslanib, maxsus dasturiy vositalar yordamida amalga oshiriladi. 3D-modellar obyektning shakli, o'lchami, hajmi va tuzilishini aks ettirib, ularni virtual muhitda ko'rish, o'zgartirish va sinovdan o'tkazish imkonini beradi.

3D-modellashtirish texnologiyalari bir nechta yo'nalishlarda rivojlangan bo'lib, ularning har biri ma'lum maqsadlarga xizmat qiladi. Muhandislik va texnologiya sohasida asosan texnik va funksional modellar yaratilsa, san'at va dizayn yo'nalishida estetik va vizual modellar ustuvor hisoblanadi. Ta'lim jarayonida esa ushbu yo'nalishlar integratsiyalashgan holda qo'llanilib, o'quvchilarning kompleks fikrlashini rivojlantiradi.

Blender dasturi

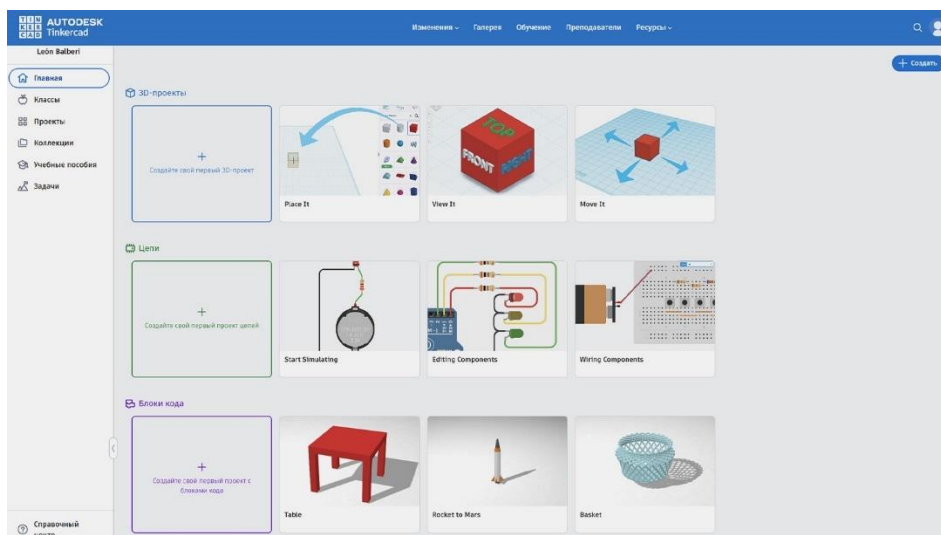
3D-modellashtirish texnologiyalari orasida **Blender** dasturi alohida o'rin tutadi. Blender bepul va ochiq manbali dastur bo'lib, 3D-modellashtirish, animatsiya, teksturalash va vizual effektlar yaratish imkonini beradi. Ushbu dastur orqali o'quvchilar murakkab obyektlarni modellashtirish, ularni harakatga keltirish va sahnalar yaratishni o'rganadilar.



Ta'lim jarayonida Blender fazoviy tafakkurni rivojlantirish uchun juda qulay vosita hisoblanadi. Masalan, mexanik detallar, biologik tuzilmalar yoki arxitektura elementlarini modellashtirish orqali o'quvchilar real obyektlarning tuzilishini chuqurroq tushunadilar. Informatika fanida Blender yordamida fayl formatlari, koordinata tizimi va uch o'lchamli grafika asoslari mustahkamlanadi.

Tinkercad

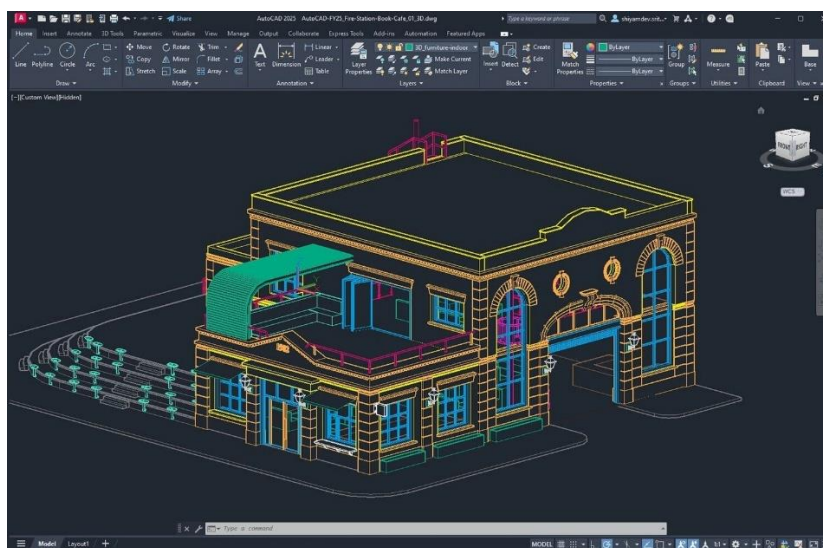
Tinkercad — bu brauzer orqali ishlaydigan, soddalashtirilgan 3D-modellashtirish muhiti bo'lib, ayniqsa boshlang'ich va o'rta bosqich o'quvchilari uchun juda mos keladi. Tinkercad yordamida murakkab kod yozmasdan, tayyor geometrik shakllardan foydalanib modellar yaratish mumkin.



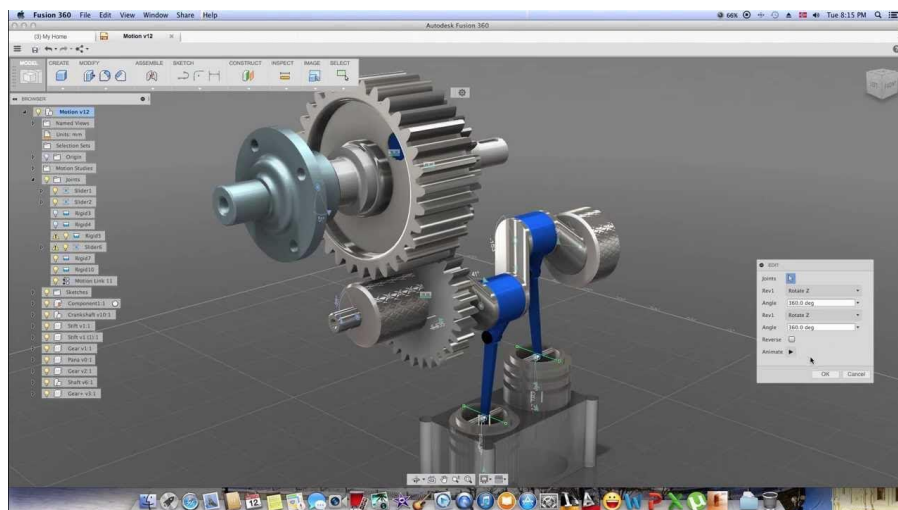
Ta'limda Tinkercad ko'pincha robototexnika va texnologiya fanlari bilan integratsiyada qo'llaniladi. Masalan, o'quvchilar 3D-printer uchun detallar yaratib, ularni real qurilmada chop etishlari mumkin. Bu jarayon informatika va muhandislik fanlari o'rtasidagi bog'liqlikni aniq namoyon etadi.

AutoCAD va Fusion 360 dasturlari

Muhandislik va sanoat dizayni yo'nalishida **AutoCAD** va **Fusion 360** dasturlari keng qo'llaniladi. AutoCAD texnik chizmalar va aniq o'lchamli modellar yaratish uchun mo'ljallangan bo'lib, muhandislik ta'limida muhim o'rin tutadi.



Fusion 360 esa parametrik modellashtirish va simulyatsiya imkoniyatlarini birlashtirgan zamonaviy platforma hisoblanadi.



Ta'lim jarayonida ushbu dasturlar yordamida o'quvchilar texnik tafakkur va dizayn asoslarini o'rganadilar. Informatika fanida bu dasturlar algoritmik fikrlash, modellashtirish va raqamli loyiha tushunchalarini mustahkamlashga xizmat qiladi.

SketchUp

SketchUp dasturi arxitektura va interyer dizaynida keng qo'llaniladi. Ushbu dastur oddiy interfeysga ega bo'lib, binolar va inshootlarning uch o'lchamli modellarini tez va qulay yaratish imkonini beradi. Ta'limda SketchUp yordamida shaharsozlik, ekologik loyihalar va dizayn ishlari bajariladi.

Bu jarayonda o'quvchilar geometrik figuralar, o'lchamlar va fazoviy joylashuvni tushunadilar, bu esa matematika va informatika fanlari bilan integratsiyani kuchaytiradi.

3D-modellashtirish va 3D-print texnologiyalari

3D-modellashtirish texnologiyalari ko'pincha 3D-print bilan uzviy bog'liq holda qo'llaniladi. Yaratilgan raqamli modellar maxsus dasturlar orqali printerga mos formatga keltiriladi va real obyekt sifatida chop etiladi. Bu jarayon o'quvchilarda "raqamli model — real mahsulot" zanjirini tushinishga yordam beradi.

Ta'limda bu texnologiya muhandislik loyihalarini amaliy natija bilan yakunlash imkonini beradi. O'quvchilar o'z modellarini real buyum sifatida ko'rib, dizayn va funktsionallikni baholaydilar.

3D-modellashtirish texnologiyalari ta'lim jarayonida o'quvchilarning ijodiy va mantiqiy fikrlashini birgalikda rivojlantiradi. Ushbu texnologiyalar yordamida o'quvchilar murakkab tushunchalarni vizual va interaktiv shaklda o'rganadilar. Bu esa bilimlarni chuqurroq o'zlashtirishga xizmat qiladi.

Pedagogik jihatdan 3D-modellashtirish loyiha asosida o‘qitish, fanlararo integratsiya va STEAM yondashuvini samarali amalga oshirish imkonini beradi. O‘quvchilar o‘z bilimlarini real muammolarni hal etishda qo‘llashni o‘rganadilar.

Robototexnika — bu mexanika, elektronika, informatika va sun‘iy intellekt elementlarini birlashtirgan fanlararo soha bo‘lib, u avtomatlashtirilgan qurilmalar va robot tizimlarini loyihalash, yaratish hamda boshqarish bilan shug‘ullanadi. Zamonaviy jamiyatda robototexnika sanoat, tibbiyot, qishloq xo‘jaligi, transport va ta‘lim sohalarida keng qo‘llanilmoqda. Ayniqsa, STEAM yondashuvi doirasida robototexnika o‘quvchilarda muhandislik tafakkuri, algoritmik fikrlash va amaliy muammo yechish ko‘nikmalarini rivojlantirishda muhim ahamiyat kasb etadi.

Robototexnika asoslarini o‘rganish, avvalo, robot tushunchasini anglashdan boshlanadi. Robot — bu ma‘lum vazifalarni mustaqil yoki yarim mustaqil bajarishga mo‘ljallangan, dastur asosida boshqariladigan texnik qurilma hisoblanadi. Robotlar atrof-muhitdan ma‘lumot olish, ushbu ma‘lumotni qayta ishlash va natijaga ko‘ra harakat qilish qobiliyatiga ega bo‘ladi. Bu jarayon informatika fanidagi “kiritish – qayta ishlash – chiqarish” modeli bilan bevosita bog‘liq.

Robototexnik tizimning asosiy tarkibiy qismlari apparat va dasturiy qismlardan iborat. Apparat qismiga mexanik konstruktsiya, dvigatellar, sensorlar va elektron plata kiradi. Mexanik qism robotning harakatini ta‘minlasa, dvigatellar aylanish yoki siljish funksiyasini bajaradi. Sensorlar esa atrof-muhitdan ma‘lumot to‘playdi, masalan, masofa, yorug‘lik, harorat yoki bosimni aniqlaydi. Elektron plata, odatda mikrocontroller bo‘lib, u barcha komponentlarning o‘zaro ishlashini boshqaradi.

Robototexnikaning dasturiy qismi informatika faniga bevosita tayanadi. Robotning xatti-harakati maxsus dastur yordamida boshqariladi. Ushbu dastur algoritmlar, shart operatorlari va sikllar asosida tuziladi. Dastur robotga qachon va qanday harakat qilish, sensorlardan kelgan ma‘lumotlarga qanday javob berish kerakligini belgilaydi. Shu bois robototexnika informatika fanida algoritmik fikrlash va dasturlashni o‘rganish uchun juda qulay amaliy muhit hisoblanadi.

Ta‘lim jarayonida robototexnika ko‘pincha mikrocontroller platformalari asosida o‘rganiladi. Arduino, Micro:bit yoki shunga o‘xshash muhitlar yordamida o‘quvchilar dastur yozish, qurilmalarni ulash va sinovdan o‘tkazishni o‘rganadilar. Ushbu platformalar soddaligi va ochiq manbaliligi bilan ta‘lim uchun ayniqsa mos keladi. Robototexnika mashg‘ulotlarida o‘quvchilar real qurilmalar bilan ishlagan holda nazariy bilimlarini amaliyotda mustahkamlaydilar.

Robototexnika asoslarida muhim tushunchalardan biri bu avtomatlashtirishdir. Avtomatlashtirish jarayonida robot ma‘lum vazifani inson aralashuvisiz bajaradi. Bunga misol sifatida chiziq bo‘ylab harakatlanuvchi robot yoki to‘siqlardan qochuvchi robotni keltirish mumkin. Bunday robotlar sensorlardan olingan

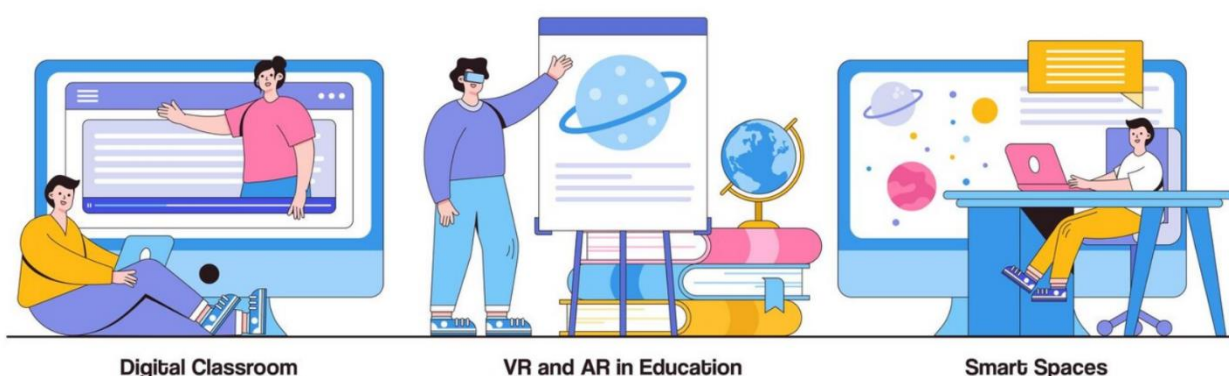
ma'lumotlar asosida qaror qabul qiladi, bu esa mantiqiy fikrlash va shartli algoritmlarni o'rganishga imkon beradi.

Robototexnika STEAM yondashuvida fanlararo integratsiyaning yorqin namunasi hisoblanadi. Fizika fanidan harakat va kuch tushunchalari, matematika fanidan o'lchovlar va hisob-kitoblar, informatika fanidan dasturlash va algoritmlar, texnologiya fanidan esa qurilma yig'ish va dizayn elementlari birgalikda qo'llaniladi. Bu integratsiya o'quvchilarga bilimlarni alohida fanlar doirasida emas, balki yagona tizim sifatida anglashga yordam beradi.

Pedagogik jihatdan robototexnika o'quvchilarning faolligini oshiradi va ularni tadqiqotga undaydi. Robot yaratish jarayonida o'quvchilar xatolarga duch keladi, ularni tahlil qiladi va yechim topadi. Bu esa tanqidiy fikrlash, muammoni tahlil qilish va ijodkorlikni rivojlantiradi. Shuningdek, robototexnika loyihalari ko'pincha jamoaviy ishlashni talab qiladi, bu esa muloqot va hamkorlik kompetensiyalarini shakllantiradi.

Ta'limda robototexnika asoslarini o'qitish o'quvchilarni kelajak kasblariga tayyorlashda muhim rol o'ynaydi. Zamonaviy texnologiyalar sharoitida robotlar va avtomatlashtirilgan tizimlar tobora ko'proq qo'llanilmoqda. Shu sababli robototexnika asoslarini erta bosqichdan o'rganish o'quvchilarning texnologik savodxonligini oshiradi va ularni raqamli jamiyatga moslashtiradi.

Ta'lim tizimining raqamli transformatsiyasi sharoitida virtual va kengaytirilgan reallik texnologiyalariga alohida e'tibor qaratilmoqda. Mazkur texnologiyalar o'quv jarayonini tashkil etish uchun yangi imkoniyatlarni ochadi, o'quvchilarning darsga faol jalb qilinishi va materialni o'zlashtirish darajasini vizualizatsiya va interaktivlik orqali sezilarli darajada oshiradi.



Virtual reallik (VR) — bu foydalanuvchi maxsus qurilmalar (masalan, Oculus Rift, HTC Vive, Meta Quest kabi VR shlemlari) yordamida o'zaro ta'sirlashadigan sun'iy raqamli muhitni yaratuvchi texnologiyadir. Bu texnologiya foydalanuvchini butunlay virtual olamga "sho'ng'itadi" va tashqi haqiqatdan ajratadi. Ta'lim muhitida VR yordamida laboratoriya tajribalari, tarixiy voqealarning rekonstruksiya, inson

tanasiga sayohatlar va oddiy auditoriyada amalda bajarish qiyin bo'lgan boshqa murakkab jarayonlarni modellashtirish mumkin.



Kengaytirilgan reallik (AR) esa raqamli elementlarni (rasmlar, matnlar, 3D obyektlar) real jismoniy dunyoga qo'shuvchi texnologiyadir. AR texnologiyasi atrofdagi haqiqatni o'zgartirmaydi, balki uni to'ldiradi. AR texnologiyalaridan foydalanish uchun eng keng tarqalgan qurilmalar — bu smartfon va planshetlardir. Ta'limda AR texnologiyasi darsliklarni jonlantirish (masalan, kamera orqali rasmga qaralganda animatsiya paydo bo'lishi), uch o'lchamli modellarni aks ettirish, interaktiv anatomiya, geometriya yoki muhandislik fanlarida amaliy ko'nikmalarni rivojlantirish uchun ishlatiladi.



Mazkur texnologiyalar yo‘nalishi o‘xshash bo‘lishiga qaramay, ular o‘rtasida asosiy farqlar mavjud:

Xususiyatlari	Virtual reallik (VR)	Kengaytirilgan reallik (AR)
Sho‘ng‘ish darajasi	To‘liq sun‘iy muhitga sho‘ng‘ish	Haqiqiy muhitga raqamli elementlarning qisman qo‘shilishi
Kerakli qurilmalar	VR shlemlari, boshqaruv pultrlari, sensorlar	Smartfon, planshet, AR ko‘zoynaklar
Real dunyodan uzilish	To‘liq	Yo‘q
Ta‘limdagi qo‘llanishi	Virtual laboratoriyalar, tarixiy sayohatlar	3D vizualizatsiya, o‘quv ilovalari, interaktiv darsliklar

Xalqaro ta‘lim texnologiyalari jamiyati (ISTE) tomonidan olib borilgan tadqiqotlarga ko‘ra, VR va AR texnologiyalaridan ta‘limda foydalanish chuqur bilimlarning shakllanishiga, fazoviy tafakkurni rivojlantirishga, vizual axborotni samarali idrok etishga, shuningdek, o‘quvchilarning tadqiqotchilik faoliyatini rag‘batlantirishga xizmat qiladi.



Bundan tashqari, O‘zbekiston muhitida ushbu texnologiyalar raqamli ta‘lim strategiyasi va Oliy ta‘lim, fan va innovatsiyalar vazirligining ta‘lim makonini raqamlashtirishga qaratilgan tashabbuslari doirasida belgilangan maqsad va vazifalarga to‘liq mos keladi. VR va AR texnologiyalarining ta‘lim jarayoniga integratsiyalashuvi an‘anaviy auditoriya ta‘limining chegaralarini yengib o‘tishga imkon beradi hamda hududidan yoki ta‘lim muassasasining texnik imkoniyatlaridan qat‘i nazar, barcha talabalar uchun sifatli ta‘lim resurslariga keng kirishni ta‘minlaydi.

Zamonaviy pedagogik nazariyalar texnologik innovatsiyalar ta'sirida sezilarli o'zgarishlarga uchramoqda. Virtual (VR) va kengaytirilgan (AR) reallik kabi immersiv texnologiyalar konstruktivizm, faoliyatga yo'naltirilgan yondashuv, ko'p intellekt nazariyasi, multimedia orqali bilim olishning kognitiv nazariyasi va immersiv (sho'ng'ish orqali) ta'lim kabi bir qator nazariy-pedagogik yondashuvlar bilan chambarchas bog'liqdir.

Konstruktivizm (Jan Piaje, Lev Vygotskiy va boshqalar) nazariyasiga ko'ra, o'quvchi bilimni o'zining individual tajribasiga asoslanib shakllantiradi, va eng samarali o'qitish ularning faol ishtiroki orqali yuz beradi. VR va AR texnologiyalari interaktiv va vizual jihatdan boy muhit yaratib, bu yondashuvga to'liq mos keladi. Talaba nafaqat tayyor ma'lumotni qabul qiladi, balki u bilan o'zaro ta'sirga kiradi: ob'ektlar bilan ishlaydi, virtual makonlarni o'rganadi, simulyatsiyalarda ishtirok etadi. Bu esa chuqurroq tushunish va bilimlarning mustahkam o'rnashishiga yordam beradi.

Vygotskiyning ijtimoiy-madaniy nazariyasi, ya'ni o'qitishda muloqot va hamkorlikning o'rni, VR/AR texnologiyalari orqali samarali tarzda amalga oshiriladi. Masalan, ko'p foydalanuvchili onlayn VR platformalar orqali talabalarning hamkorlikda masalalarni hal qilishi, muhokamalar olib borishi va geografik jihatdan turli joyda bo'lsa ham, birgalikda ishlashi ta'minlanadi.

Faoliyatga asoslangan yondashuv, ayniqsa ta'lim tizimida keng qo'llaniladi va amaliy ko'nikmalarni o'quv faoliyati orqali shakllantirishga urg'u beradi. VR texnologiyasi orqali bu faoliyatni virtual muhitda tashkil etish mumkin: virtual laboratoriyalar, real ishlab chiqarish jarayonlarini simulyatsiya qilish — ayniqsa texnik va tibbiyot oliygohlari uchun dolzarbdir.

Multimediaviy o'qitishning kognitiv nazariyasi (Richard Mayer)ga ko'ra, ma'lumotlar vizual va audio kanallar orqali birgalikda taqdim etilganda samaraliroq o'zlashtiriladi. VR va AR texnologiyalari uch o'lchamli grafikalar, tovush va ba'zida teginish (haptik) orqali o'quv materiallarini ko'p kanalli shaklda yetkazadi, bu esa tushunish va eslab qolishni yaxshilaydi.

Govard Gardnerning ko'p intellekt nazariyasiga ko'ra, har bir o'quvchida ustun bo'lgan intellekt turi mavjud (vizual-fazoviy, kinestetik, mantiqiy-matematik va hokazo). Immersiv texnologiyalar ta'lim jarayonini turli sezgi kanallari orqali moslashtirish imkonini beradi: vizual modellar, interaktiv topshiriqlar, audiotavsiyalar va harakatga asoslangan o'zaro ta'sirlar orqali.

Immersiv ta'lim — bu talaba “sho'ng'ib kiradigan” va faol ishtirokchiga aylangan muhitni yaratishga asoslangan zamonaviy ta'lim yondashuvidir. Tadqiqotlar shuni ko'rsatmoqdaki, VR texnologiyasi orqali yuzaga keluvchi “ishtirok effektlari” o'quvchining emotsional va kognitiv jalb qilinishini kuchaytiradi, motivatsiyani va o'quv faolligini oshiradi. Bu yondashuv ayniqsa murakkab, mavhum yoki real

sharoitda modellashtirish qiyin bo‘lgan jarayonlarni (masalan, molekulalarning harakati, anatomiya, tarixiy voqealar yoki murakkab muhandislik tizimlari) o‘rgatishda foydalidir.

VR va AR texnologiyalaridan foydalanish, shuningdek, O‘zbekistonning ta’lim sohasidagi **kompetensiyaga asoslangan yondashuvi** bilan uyg‘unlashadi. Bu yondashuv shunchaki bilim emas, balki real kasbiy faoliyatda zarur bo‘lgan ko‘nikma va malakalarni rivojlantirishga qaratilgan. Immersiv texnologiyalar talabalarga kasbiy holatlarni modellashtirish va xavfsiz muhitda amaliy harakatlarni mashq qilish imkonini beradi.

Nihoyat, **inklyuziv ta’lim** nuqtai nazaridan qaraganda, VR va AR texnologiyalari imkoniyati cheklangan talabalarga moslashtirilgan ta’lim muhiti yaratishga xizmat qiladi. Masalan, eshitish yoki ko‘rishda muammoga ega bo‘lgan talabalar ovozli boshqaruvli VR shlemlari yoki yirik vizual elementlar orqali o‘quv jarayonida to‘laqonli ishtirok etishlari mumkin.

VR va AR texnologiyalari nafaqat an’anaviy pedagogik tamoyillarni qo‘llab-quvvatlaydi, balki ularni yangi bosqichga olib chiqib, o‘qitish va o‘rganish imkoniyatlarini kengaytiradi. Bu texnologiyalar bilimlarni passiv qabul qilishdan ko‘ra, faol va shaxsga yo‘naltirilgan o‘quv jarayonini yo‘lga qo‘yish imkonini beradi. Bu esa raqamli jamiyat talablariga mos va ta’lim tizimini modernizatsiya qilish maqsadlariga xizmat qiladi.

VR va AR texnologiyalarining ta’limdagi rivoji texnologik jihatdan rivojlangan davlatlarda boshlangan bo‘lib, bu yo‘nalishda hukumat, xususiy kompaniyalar va oliy ta’lim muassasalari ta’lim jarayonining raqamli transformatsiyasiga faol investitsiya kiritmoqda. Ushbu texnologiyalar nafaqat nazariyani o‘rganishga, balki o‘quv materialiga “sho‘ng‘ish”, uni yanada ko‘rgazmali, qiziqarli va amaliyotga yo‘naltirilgan qilishga imkon beradi. Quyida turli davlatlarda VR va AR texnologiyalarining ta’limda qo‘llanilishiga oid misollar keltirilgan.

AQSh

AQSh VR va AR texnologiyalarini ta’limga joriy etish bo‘yicha yetakchi mamlakatlardan biridir. Bu yerda immersiv ta’limni rivojlantirishga qaratilgan ko‘plab tashabbuslar mavjud. Ulardan biri — **Google Expeditions** platformasi bo‘lib, u orqali o‘qituvchilar o‘quvchilarga muzeylar, tarixiy yodgorliklar, okean tubi yoki inson tanasi bo‘ylab virtual ekskursiyalarni tashkil etishlari mumkin. O‘quvchilar virtual reallik ko‘zoynaklari yordamida Misr piramidalari, Oy sathi yoki okean tubiga “sayohat” qilishadi — bularning barchasi sinf xonasidan chiqmasdan amalga oshiriladi.



Shuningdek, AQShda **zSpace**, **Labster**, **Nearpod VR** kabi VR/AR ta'lim texnologiyalari sohasida faoliyat yurituvchi ko'plab startaplar mavjud.



Masalan, **Labster** biologiya, kimyo va fizika bo'yicha virtual laboratoriyalarni taqdim etadi, bu esa real laboratoriyalarni qurishga bo'lgan ehtiyojni kamaytiradi hamda xavfsiz va arzon ta'lim imkonini yaratadi.

AQSh Ta'lim vazirligi VR/AR joriy qilayotgan maktab va universitetlarni grantlar va texnik yordam bilan qo'llab-quvvatlaydi. O'qituvchilar maxsus malaka oshirish kurslarida ushbu texnologiyalarni amaliyotda qo'llashni o'rganadilar.



Buyuk Britaniya

Buyuk Britaniyada VR va AR texnologiyalari maktab va oliy ta'limda faol qo'llanilmoqda. **Birmingham universiteti** va **University College London** singari nufuzli oliygohlar VR texnologiyalaridan tibbiyot, muhandislik va arxitektura yo'nalishlarida foydalanadi.

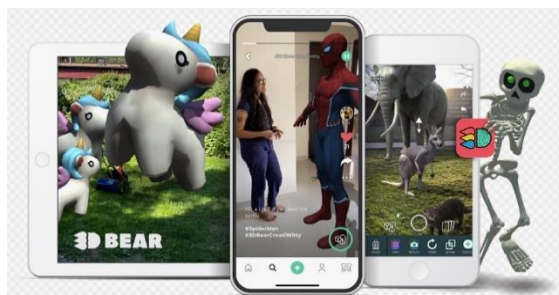
“Immerse Education” dasturi doirasida tarix, geografiya va adabiyot fanlari virtual muhitda o‘rgatilgan. Masalan, o‘quvchilar Uilyam Shekspir asarlarini o‘rganayotganda 16-asrdagi London “Globus” teatriga virtual sayohat qilishgan.



Shuningdek, maktab muzeylari va ko‘rgazmalari interaktiv tus olgan — o‘quvchilar AR ilovalari o‘rnatilgan planshet yordamida ekspozitsiyaga qarab 3D modellar, audiogidlar va videomateriallarni ko‘rishlari mumkin.

Finlandiya

Innovatsion ta’lim tizimi bilan mashhur bo‘lgan Finlandiyada AR texnologiyalari boshlang‘ich va o‘rta maktablarda faol joriy qilinmoqda. Bu yerda asosiy e’tibor bolalarda ijodkorlik, mustaqillik va o‘qishga qiziqishni rivojlantirishga qaratilgan. **3DBear** kompaniyasi tomonidan ishlab chiqilgan



platforma orqali o‘quvchilar o‘zlarining AR sahnalarini yaratishlari mumkin. Masalan, bolalar virtual shahar qurib, unda binolar, odamlar va transportni joylashtiradilar, so‘ng ularning tuzilmasi haqida hikoya qilishadi. Bu yondashuv tasavvur, fazoviy tafakkur va jamoada ishlash ko‘nikmalarini rivojlantiradi.

Shuningdek, Finlandiya maktablari migratsion fon bilan bog‘liq muammolarni hal qilishda VR texnologiyalaridan foydalanmoqda. Yangi muhitga moslashayotgan migrant o‘quvchilar uchun do‘kon, poliklinika va boshqa kundalik hayotiy holatlar simulyatsiyasi orqali ijtimoiy integratsiya tezlashadi.

Janubiy Koreya

Axborot texnologiyalari jihatdan rivojlangan Janubiy Koreya VR va AR texnologiyalarini ta’limning davlat strategiyasiga kiritgan. Mamlakat Ta’lim vazirligi maktablar va kollejlarda uchun VR-kontent yaratishga moliyaviy ko‘mak beradi. 2020-yildan boshlab “Smart Education” dasturi doirasida 200 dan ortiq maktabga VR sinflari joriy qilindi.

Muhandislik yo‘nalishidagi talabalar **Korea Advanced Institute of Science and Technology (KAIST)** universitetida VR-laboratoriyalarda uch o‘lchamli muhitda murakkab tizimlar va mexanizmlarni o‘rganmoqdalar. Shuningdek, xorijiy tillarni o‘rganish uchun ham VR platformalar ishlab chiqilgan: talabalar real vaqt rejimida virtual personajlar bilan suhbatlashib, og‘zaki nutq ko‘nikmalarini mustahkamlashlari mumkin.

Yaponiya

Yaponiya VR va AR texnologiyalarini favqulodda vaziyatlarga tayyorgarlikda keng

qo'llamoqda. Maktab o'quvchilari zilzilalar vaqtida qanday harakat qilish kerakligini VR-senaryilar yordamida o'rganishadi — bu orqali real evakuatsiya va birinchi yordam holatlari simulyatsiya qilinadi. Bu esa xavfsiz, ammo real muhitga yaqin sharoitda mashq qilish imkonini beradi.

Oliy ta'lim muassasalarida esa VR texnologiyalari tibbiyot, arxitektura va dizayn yo'nalishlarida keng qo'llaniladi. Talabalar kelajakdagi bino ichida "sayr qilishlari", inson anatomiyasini o'rganishlari yoki virtual bemorni operatsiya qilishlari mumkin. Bu yondashuv mutaxassis tayyorlash sifatini sezilarli darajada oshiradi.

Kanada

Kanadada ham VR/AR ta'lim yo'nalishlari faol rivojlanmoqda. Universitetlar texnologik kompaniyalar bilan hamkorlikda interaktiv kurslar ishlab chiqishmoqda. Masalan, **British Columbia universiteti** VR orqali neyrobiologiya fanini o'rgatadi — talabalar miya tuzilishini 3D formatda o'rganish imkoniyatiga ega bo'lishadi.

Shuningdek, uzoq hududlardagi maktablarga VR orqali ta'lim resurslarini yetkazish mexanizmlari mavjud. Bu maktablar virtual ekskursiyalar va laboratoriya mashg'ulotlaridan foydalanib, sifatli ta'lim oladi.

Hindiston

Hindistonda ta'limdagi tengsizliklarga qaramay, AR ilovalari keng qo'llanilmoqda. **Next Education** va **Edukart** kabi kompaniyalar ingliz va hind tillarida arzon va qulay AR kontent ishlab chiqmoqda, bu esa oddiy smartfonlar orqali ishlaydi. AR-kitoblar boshlang'ich maktablarda ommalashmoqda — bolalar kamera orqali rasmga qarab animatsiyalarni ko'radi, ovozli sharhlarni eshitadi. Bu, ayniqsa, kam ta'minlangan oilalardan chiqqan o'quvchilarni o'qishga jalb qilishda samarali vosita hisoblanadi.

Hindiston hukumati **Digital India** dasturi doirasida AR/VR joriy qilish tashabbuslarini qo'llab-quvvatlamoda, bu esa mamlakatning ta'lim tizimini texnologik rivojlantirishga bo'lgan intilishini namoyon qiladi. Xalqaro tajriba shuni ko'rsatadiki, VR va AR texnologiyalarini ta'limning barcha bosqichlarida — maktabgacha ta'limdan to oliy ta'limgacha — samarali qo'llash mumkin. Ular axborotni yetkazishning yangi shakllarini yaratib, o'qishni qiziqarli, ochiq va samarali qiladi. Virtual laboratoriyalar, tarixiy rekonstruksiyalar, til o'rganish, kasbiy simulyatsiyalar — bularning barchasi immersiv texnologiyalar orqali amalga oshirilmoqda.

Qurilmalar narxining yuqoriligi va pedagoglar tayyorlash zaruriyatiga qaramay, aksariyat davlatlar VR/AR texnologiyalarini ta'limga integratsiyalash bo'yicha jiddiy harakat qilmoqda. Sababi, ularning uzoq muddatli afzalliklari yaqqol ko'rinadi: amaliy ko'nikmalar shakllanishi, ijodkorlik rivojlanishi, o'quvchilarning motivatsiyasi va ishtirokining ortishi.

V.TESTLAR

Savol	To'g'ri javob	Muqobil javob	Muqobil javob	Muqobil javob
STEAM yondashuvida "S" harfi nimani anglatadi?	Science	Software	System	Society
STEAM yondashuvida fanlar qanday o'qitiladi?	Integratsiyalashgan holda	Alohida	Faqat nazariy	Mustaqil
STEAM yondashuvining asosiy maqsadi nima?	Amaliy muammo yechish	Faqat test yechish	Yodlash	Nazariy baholash
Informatika STEAMda qanday rol o'ynaydi?	Markaziy integrator	Yordamchi fan	Ikkinchi darajali	Faqat texnik
3D-modellashtirish nima?	Uch o'lchamli model yaratish	Grafik chizish	Matn yozish	Video montaj
3D-modellashtirishda qaysi dastur ochiq manbali?	Blender	AutoCAD	Fusion 360	SolidWorks
Tinkercad qaysi turdagi platforma?	Brauzerga asoslangan	Faqat offline	Mobil ilova	Matn muharriri
3D-print uchun model qaysi formatda bo'ladi?	STL	DOCX	PDF	TXT
Robototexnika qaysi fanlar asosida quriladi?	Informatika va muhandislik	Faqat matematika	Faqat san'at	Faqat biologiya
Robotning "miyasi" nima hisoblanadi?	Mikrocontroller	Sensor	Motor	Korpus
Arduino nima?	Mikrocontroller platformasi	Grafik dastur	O'yin	Operatsion tizim
Sensor vazifasi nima?	Ma'lumot yig'ish	Harakat	Dizayn	Saqlash
Aktuator nima vazifa bajaradi?	Harakatni amalga oshiradi	O'lchaydi	Hisoblaydi	Tahlil qiladi
Robot dasturi nimaga asoslanadi?	Algoritm'larga	Rang'larga	Dizaynga	Musiqaga
Chiziq kuzatuvchi robot nimaga tayanadi?	Sensor ma'lumotiga	Tasodifga	Ovozga	Yorug'liksiz
Simulyatsiya nima?	Jarayonni virtual modellashtirish	Real tajriba	Chizma	Matn
Simulyatsiyaning afzalligi nimada?	Xavfsiz va tejamkor	Qimmat	Murakkab	Sekin
PhET platformasi nimaga xizmat qiladi?	Virtual tajribalar	Dasturlash	Dizayn	Test
VR texnologiyasi nimani anglatadi?	Virtual reallik	Kengaytirilgan matn	3D chizma	Video
AR texnologiyasining asosiy xususiyati nima?	Real muhitga raqamli obyekt qo'shish	To'liq virtual muhit	Faqat video	Faqat rasm
VR/AR STEAMda	Vizual tushunishni	Yodlashni	Passivlikni	Nazariyani

Savol	To'g'ri javob	Muqobil javob	Muqobil javob	Muqobil javob
nimani kuchaytiradi?				
Informatika VR/ARda nimani ta'minlaydi?	Dasturlash va mantiq	Faqat dizayn	Faqat ovoz	Faqat rang
STEAM loyihasi nimaga asoslanadi?	Muammo va loyiha faoliyatiga	Testga	Ma'ruza	Yodlash
STEAMda baholash qanday bo'ladi?	Kompetensiyaga asoslangan	Faqat yozma	Og'zaki	Reytingsiz
Robototexnika darslarida qaysi ko'nikma rivojlanadi?	Algoritmik fikrlash	Yod olish	Nusxa ko'chirish	Passivlik
3D-modellashtirish qaysi tafakkurni rivojlantiradi?	Fazoviy fikrlash	Eshitish	Nutq	Yodlash
Fusion 360 qaysi sohada qulay?	Muhandislik dizayni	Matn tahriri	Audio	Test
AutoCAD asosan nimaga mo'ljallangan?	Texnik chizmalar	O'yin	Video	Web
STEAMda san'at (A) nimani ifodalaydi?	Dizayn va ijod	Nazariya	Formulalar	Kod
Canva STEAMda nimaga xizmat qiladi?	Vizual taqdimot	Dasturlash	Sensor	Simulyatsiya
Figma qaysi yo'nalishda ishlatiladi?	UI/UX dizayn	Robot yig'ish	Matematika	Fizika
Google Classroom vazifasi nima?	Ta'limni boshqarish	Modellashtirish	Robot yaratish	3D print
Kahoot STEAMda nimaga xizmat qiladi?	Interaktiv baholash	Dasturlash	Sensor	Dizayn
Excel STEAMda qayerda ishlatiladi?	Ma'lumot tahlili	Animatsiya	Robot	VR
STEAMda informatika nimani birlashtiradi?	Barcha fanlarni	Faqat texnologiyani	Faqat san'atni	Faqat matematikani
Robototexnika qaysi yondashuvga mos?	Loyiha asosida	Passiv	Nazariy	Yodlash
3D-modeldan real buyum olish qanday ataladi?	3D-print	Render	Chizma	Eskiz
Simulyatsiya qaysi fanlarda foydali?	Fizika va kimyo	Faqat tarix	Faqat til	Faqat adabiyot
VR qaysi sezgiga ko'proq ta'sir qiladi?	Ko'rish	Hid	Ta'm	Eshitish
AR ta'limda nimani oshiradi?	Qiziqish va tushunish	Zerikish	Passivlik	Charchoq
STEAMda texnologiya (T) nimani anglatadi?	Raqamli vositalar	Faqat kompyuter	Faqat internet	Faqat telefon
Robototexnika darslari	Jamoaviy ish	Yakka yodlash	Nusxa	Passivlik

Savol	To'g'ri javob	Muqobil javob	Muqobil javob	Muqobil javob
nimani talab qiladi?				
STEAM loyihasi oxirida nima bo'ladi?	Taqdimot	Imtihon	Diktant	Nazorat ishi
Informatika STEAMda nimani o'rgatadi?	Muammoni algoritm orqali yechish	Yodlash	Ko'chirish	Takrorlash
3D-modellashtirish qaysi fan bilan chambarchas bog'liq?	Matematika	Ona tili	Tarix	Adabiyot
Robototexnikada xatoni topish nima deyiladi?	Debugging	Rendering	Printing	Modeling
STEAM yondashuvi nimaga yo'naltirilgan?	Kelajak kompetensiyalariga	Faqat bahoga	Faqat nazariyaga	Faqat testga
Ta'lim jarayonini raqamli texnologiyalar asosida optimallashtirishning asosiy mezonini qaysi?	Resurslar va natijalar nisbatining yaxshilanishi	Texnika sonining ko'payishi	Dars vaqtining qisqarishi	Platformalar xilma-xilligi
Raqamli optimallashtirishda ma'lumotlar tahlili nega muhim?	Qaror qabul qilishni asoslaydi	Dizaynni yaxshilaydi	Internet tezligini oshiradi	Vizual effekt beradi
NumPy massivlarining asosiy ustunligi nimada?	Tez va vektorli hisoblashlarda	Vizual interfeysda	Fayl boshqaruvida	Grafik animatsiyada
NumPy'dan foydalanish qaysi holatda ayniqsa samarali?	Katta hajmdagi sonli ma'lumotlarda	Matnli hujjatlarda	Rasm tahririda	Veb dizaynda
Pandas DataFrame tushunchasi nimani ifodalaydi?	Tuzilgan jadval ko'rinishidagi ma'lumotlar modelini	Grafik obyektini	Algoritmni	Veb sahifani
Pandas ta'lim jarayonini qaysi jihatdan optimallashtiradi?	Statistik va tahliliy qarorlarni qo'llab-quvvatlab	Darsliklarni almashtirib	Platformani tezlashtirib	Internetni cheklab
Pandas va NumPy integratsiyasi nimani ta'minlaydi?	Kompleks tahlil va hisoblashni	Faqat vizual dizaynni	Faqat fayl almashishni	Faqat ma'ruza tayyorlashni
Flask freymvorkining minimalistik yondashuvi nimani anglatadi?	Faqat zarur komponentlar bilan ishlashni	Kam funktsionallikni	Faqat frontendni	Faqat lokal tizimni
Flask ta'limiy ilovalarda qaysi holatda maqbulroq?	Tez prototiplash talab etilganda	Katta LMS yaratishda	Mobil o'yin ishlab chiqishda	CAD tizimda

Savol	To'g'ri javob	Muqobil javob	Muqobil javob	Muqobil javob
Django freymvorkining "batareyalar bilan birga" tamoyili nimani bildiradi?	Tayyor modullar kengligi	Past tezlik	Murakkab sozlash	Faqat backend
Django'ning ta'lim platformalarida afzalligi nimada?	Kengayuvchan va xavfsiz tuzilishda	Faqat dizaynda	Minimalizm	Oddiy skriptlarda
Flask va Django tanlovi nimaga bog'liq?	Loyiha ko'lami va murakkabligiga	O'qituvchi yoshi	Internet tezligiga	Operatsion tizimga
Kutubxonalarni integratsiyalashdagi asosiy xavf nimada?	Muvofiqlik va murakkablik oshishida	Tezlik oshishida	Funksiya ko'payishida	Dizayn yaxshilanishida
Ta'limda raqamli yechimlarni joriy etishda eng barqaror yondashuv qaysi?	Bosqichma-bosqich joriy etish	Bir martalik o'rnatish	Faqat pilot loyiha	To'liq avtomatlashtirish
Raqamli transformatsiyada inson omili nimaga tenglashtiriladi?	Texnologiya muvaffaqiyatining kalitiga	Ikkinchi darajali omilga	Texnik muammoga	Resurs isrofga
Kiberxavfsizlikning asosiy tamoyillari qaysi triadaga asoslanadi?	Maxfiylik, yaxlitlik, mavjudlik	Tezlik, dizayn, qulaylik	Grafik, audio, video	Kod, test, server
Axborot xavfsizligida "yaxlitlik" nimani bildiradi?	Ma'lumot o'zgarishsizligini	Tez uzatilishini	Ochiqligini	Dizaynini
Shaxsiy ma'lumotlarni qayta ishlashda asosiy talab qaysi?	Foydalanuvchi roziligi	Texnik qulaylik	Ochiq saqlash	Nusxa olish
Phishing hujumining asosiy xavfi nimada?	Ishonchni suiiste'mol qilishda	Tizimni sekinlashtirishda	Faylni o'chirishda	Dizaynni buzishda
Kiberxavflarning ta'lim muassasalari uchun xavfi nimada?	Talaba va xodim ma'lumotlari yo'qotilishida	Dars jadvali o'zgarishida	Auditoriya bandligida	Baholashda
Kuchli autentifikatsiya nimani anglatadi?	Bir nechta tekshiruv omillaridan foydalanishni	Faqat parolni	Faqat loginni	Faqat biometrikani
Korporativ ma'lumotlar uchun eng muhim himoya qaysi?	Kirish huquqlarini qat'iy boshqarish	Dizaynni yangilash	Platformani almashtirish	Serverni ko'paytirish
Xavf-xatarlarning oldini olish strategiyasi	Oldindan rejalashtirilganda	Hujumdan keyin	Muammo yuzaga	Tasodifan

Savol	To'g'ri javob	Muqobil javob	Muqobil javob	Muqobil javob
qachon samarali? Antivirus va firewall farqi nimada?	Zararli kod va tarmoq nazoratida	Dizayn uslubida	kelganda Saqlash hajmida	Tezlikda
Zamonaviy texnologiyalar kiberxavfsizlikni qanday murakkablashtiradi?	Hujum sirtini kengaytirib	Xavflarni yo'qotib	Tahdidlarni kamaytirib	Nazoratni osonlashtirib
Bulutli texnologiyalarda asosiy xavf nimada?	Ma'lumotlarni masofadan boshqarishda	Qurilma og'irligida	Grafik sifatida	Kod uzunligida
Ta'lim muassasalarida kiberxavfsizlik siyosati nimani ta'minlaydi?	Tizimli va uzluksiz himoyani	Faqat texnik himoyani	Faqat hujjatlarni	Faqat serverni
Amaliy mashg'ulotlarda Pandas va NumPy qo'llashning asosiy pedagogik foydasi nimada?	Nazariyani real ma'lumot bilan bog'lashda	Yodlashni oshirishda	Test sonini ko'paytirishda	Darsni qisqartirishda

VI.GLOSSARIY

Termin	O‘zbek tilidagi sharhi	Ingliz tilidagi sharhi
Sun’iy intellekt	Inson aqliga xos bo‘lgan fikrlash, o‘rganish va qaror qabul qilish jarayonlarini bajaruvchi texnologiya	Artificial Intelligence – technology that simulates human intelligence processes
Mashinali o‘qitish	Ma’lumotlar asosida o‘z-o‘zidan o‘rganadigan sun’iy intellekt yo‘nalishi	Machine Learning – a method of teaching computers to learn from data
Neyron tarmoqlar	Inson miyasi faoliyatiga o‘xshash ishlaydigan hisoblash modeli	Neural Networks – computing systems inspired by the human brain
Shaxsiylashtirilgan ta’lim	Har bir o‘quvchining ehtiyojiga moslashtirilgan o‘qitish jarayoni	Personalized Learning – education tailored to individual learners
Adaptiv o‘qitish	O‘quvchi bilim darajasiga qarab avtomatik moslashadigan ta’lim texnologiyasi	Adaptive Learning – learning that adapts to a student’s performance
Aqlli ta’lim tizimi	Sun’iy intellekt asosida ishlovchi, o‘qitish va baholashni avtomatlashtiruvchi tizim	Intelligent Tutoring System – AI-based educational system
Chatbot	Foydalanuvchi bilan avtomatik muloqot qiluvchi sun’iy intellekt dasturi	Chatbot – AI program that communicates with users
Virtual yordamchi	O‘quvchilarga maslahat va yordam beruvchi raqamli yordamchi	Virtual Assistant – digital assistant that supports users
Test generatori	Avtomatik ravishda test savollarini yaratuvchi tizim	Test Generator – system that automatically creates tests
Wayground	O‘yinlashtirilgan test va baholash platformasi	Wayground – interactive assessment platform
Kahoot	Onlayn viktorina va testlar yaratish platformasi	Kahoot – game-based learning platform
Mustaqil ta’lim	O‘quvchining o‘z tashabbusi bilan bilim olishi	Independent Learning – self-directed learning process
Ijodiy ish	O‘quvchining mustaqil va ijodiy faoliyat mahsuli	Creative Work – result of creative activity
Raqamli etika	Raqamli muhitda axloqiy me’yorlarga rioya qilish qoidalari	Digital Ethics – moral principles in digital environments
Akademik halollik	Ta’limda halol, mustaqil va mas’uliyatli faoliyat yuritish	Academic Integrity – honesty and responsibility in education
Plagiat	Boshqa muallif ishini manbasiz o‘zlashtirish	Plagiarism – using others’ work without attribution
Mualliflik huquqi	Ijodiy ish egasining qonuniy huquqlari	Copyright – legal rights of content creators
Axborot xavfsizligi	Ma’lumotlarni ruxsatsiz foydalanishdan	Information Security –

	himoyalash	protection of data from unauthorized access
Shaxsiy ma'lumot	Shaxsga oid maxfiy axborotlar	Personal Data – information related to an individual
Raqamli madaniyat	Raqamli texnologiyalardan ongli va madaniyatli foydalanish	Digital Culture – responsible use of digital technologies
Katta ma'lumotlar	Juda katta hajmdagi va tez yangilanadigan ma'lumotlar majmuasi	Big Data – extremely large and complex data sets
Algoritm	Muammoni yechish uchun ketma-ket amallar majmuasi	Algorithm – a step-by-step procedure to solve a problem
Ma'lumotlarni tahlil qilish	Ma'lumotlardan foydali xulosalar chiqarish jarayoni	Data Analysis – process of examining data to extract insights
Aqli testlash	O'quvchi bilimiga mos savollar beruvchi baholash usuli	Smart Testing – adaptive assessment based on learner level
Interfaol ta'lim	Faol ishtirokga asoslangan o'qitish shakli	Interactive Learning – learning through active engagement
O'yinlashtirish	O'yin elementlarini ta'lim jarayoniga qo'llash	Gamification – use of game elements in education
Tavsiya tizimi	Foydalanuvchiga mos materiallarni taklif qiluvchi tizim	Recommendation System – system suggesting personalized content
Nutqni aniqlash	Inson nutqini kompyuter tomonidan tanish jarayoni	Speech Recognition – technology that identifies spoken language
Matnni tahlil qilish	Matn mazmunini avtomatik tushunish va qayta ishlash	Text Analysis – automatic processing and understanding of text
Vizual tanish	Tasvir va rasmlarni aniqlash texnologiyasi	Computer Vision – technology that enables machines to see images
Aqli platforma	Sun'iy intellekt asosida ishlovchi ta'lim muhiti	Smart Platform – AI-based learning environment
Masofaviy ta'lim	Internet orqali amalga oshiriladigan ta'lim	Distance Learning – education delivered online
Raqamli savodxonlik	Raqamli texnologiyalarni tushunish va qo'llay olish	Digital Literacy – ability to use digital technologies effectively
Avtomatik baholash	Baholashni kompyuter yordamida amalga oshirish	Automated Assessment – evaluation performed by software
O'quv tahlili	O'quv jarayonidagi ma'lumotlarni tahlil qilish	Learning Analytics – analysis of educational data

Sun'iy neyron	Neyron tarmoqlardagi hisoblash elementi	Artificial Neuron – basic unit of neural networks
Aqli kontent	O'quvchiga mos avtomatik yaratilgan material	Intelligent Content – AI-generated adaptive content
Axborot ishonchliligi	Ma'lumotning haqqoniylik darajasi	Information Reliability – credibility of information
Kiberxavfsizlik	Raqamli tizimlarni hujumlardan himoyalash	Cybersecurity – protection of systems from cyber threats
Raqamli iz	Internetda qoldirilgan foydalanuvchi ma'lumotlari	Digital Footprint – data trail left by users online

VII. FOYDALANILGAN ADABIYOTLAR

I. Maxsus adabiyotlar

1. UNESCO. Education in a Post-COVID World: Nine Ideas for Public Action. – Paris: UNESCO, 2020.
2. OECD. Digital Education Outlook 2021: Pushing the Frontiers with AI, Blockchain and Robots. – Paris: OECD Publishing, 2021.
3. World Economic Forum. Schools of the Future: Defining New Models of Education for the Fourth Industrial Revolution. – Geneva, 2020.
4. Redecker, C. European Framework for the Digital Competence of Educators (DigCompEdu). – Luxembourg: Publications Office of the European Union, 2020.
5. Anderson, J., Rainie, L. The Future of Digital Learning. – Pew Research Center, 2021.
6. Hodges, C., Moore, S., Lockee, B., Trust, T., Bond, A. The Difference Between Emergency Remote Teaching and Online Learning. – EDUCAUSE Review, 2020.
7. Selwyn, N. Education and Technology: Key Issues and Debates. – 3rd ed. – London: Bloomsbury Academic, 2022.
8. Holmes, W., Bialik, M., Fadel, C. Artificial Intelligence in Education: Promises and Implications for Teaching and Learning. – Boston: Center for Curriculum Redesign, 2021.
9. Kergel, D., Heidkamp, B., Telléus, P. K., Rachwal, T., Nowakowski, S. Digital Transformation in Higher Education. – Wiesbaden: Springer, 2021.
10. Alimov, R. X., Xoliqulov, A. A. Raqamli ta'lim muhitida pedagogik texnologiyalar. – Toshkent: Fan va texnologiya, 2021.
11. Karimov, U. U. Informatika ta'limida innovatsion yondashuvlar. – T.: O'qituvchi, 2022.
12. Mirzayev, M. M. Ta'limda raqamli texnologiyalar va kompetensiyaviy yondashuv. – T.: Akademnashr, 2023.
13. Bond, M., Bedenlier, S., Marín, V. I., Händel, M. Emergency Remote Teaching in Higher Education: Mapping the First Global Online Semester. – International Journal of Educational Technology in Higher Education, 2021.
14. Zhu, Z. T., Yu, M. H., Riezebos, P. A Research Framework of Smart Education. – Smart Learning Environments, 2020.
15. Qodirov, B. S. Axborot-kommunikatsiya texnologiyalari asosida informatika fanini o'qitish metodikasi. – T., 2024.

IV. Elektron ta'lim resurslari

6. <http://natlib.uz>

7. <http://yedu.uz>
8. <http://lex.uz>
9. <http://lib.bimm.uz>
10. <http://ziyonet.uz>