

## UMUMKASBIY VA IXTISOSLIK FANLAR INTEGRATSIYASINING MOBIL' ILOVASINI ISHLAB CHIQUISH

*Orinbetov Nurilla Turdimuratovich,  
Ajiniyoz nomidagi Nukus davlat pedagogika instituti, Texnologik ta'lim kafedrası dotsenti*

*Ismailova Momogul Hamraboy qizi,  
Urganch davlat pedagogika instituti, Fizika-matematika va texnologik ta'lim kafedrası o'qituvchisi.*

*Annotatsiya: Ushbu matnda umumkasbiy va ixtisoslik fanlar o'rtasidagi chuqur o'zaro aloqani ta'minlash maqsadida veb-ilovalarni o'quv jarayoniga eng samarali integratsiyalashda qo'llanilishi mumkin bo'lgan usullar va algoritmlarni tahlil qilishga alohida e'tibor qaratilgan. Mobil ilovalarni yaratish uchun Flask-dan foydalanishning afzalliklari va kamchiliklarini tahlil qilishdir. Flask Django va FastAPI kabi boshqa veb-frameworkalar bilan ishlab chiqish qulayligi, unumdorligi, kengaytirilishi va hamjamiyatni qo'llab-quvvatlash nuqtai nazaridan taqqoslanadi. Bundan tashqari, mobil ilovani ma'lumotlar bazalari, keshlash tizimlari va bulutli platformalar kabi zamonaviy texnologiyalar bilan integratsiyalashuviga alohida e'tibor qaratiladi.*

*Kalit so'zlar: veb-frameworka, veb-ilovalar, raqamli texnologiyalar, Flask Django va FastAPI veb-frameworkalar, Flask mikrofreymvorkom, saytlararo skriptlar, shifrlashdan va seansni boshqarish, HTML, CSS va JavaScript-dan foydalanish, modulli test, integratsiya testi, optimallashtirish usullari, adaptiv veb-sahifa.*

## РАЗРАБОТКА МОБИЛЬНОГО ПРИЛОЖЕНИЯ «ИНТЕГРАЦИЯ ОБЩЕПРОФЕССИОНАЛЬНЫХ И СПЕЦИАЛИЗИРОВАННЫХ ДИСЦИПЛИН»

*Орынбетов Нурылла Турдымуратович,  
Нукусский государственный педагогический институт имени Ажинияза, доцент кафедры  
технологического обучения*

*Исмаилова Момогул Хамрабой кизи,  
Ургенчский государственный педагогический институт – преподаватель кафедры физико-  
математического и технологического обучения*

*Аннотация: в этом тексте особое внимание уделяется анализу методов и алгоритмов, которые можно использовать для наиболее эффективной интеграции веб-приложений в учебный процесс с целью обеспечения глубокого взаимодействия между общепрофессиональными и специализированными дисциплинами. Это анализ преимуществ и недостатков использования Flask для создания мобильных приложений. Flask сравнивается с другими веб-фреймворками, такими как Django и fastapi, с точки зрения простоты разработки, производительности, масштабируемости и поддержки сообщества. Кроме того, особое внимание уделяется интеграции мобильного приложения с современными технологиями, такими как базы данных, системы кэширования и облачные платформы.*

*Ключевые слова: веб-фреймворк, веб-приложения, цифровые технологии, Flask Django и fastapi веб-фреймворки, Flask microframework, межсайтовые скрипты, шифрование и управление сеансами, использование HTML, CSS и JavaScript, модульное тестирование, интеграционное тестирование, методы оптимизации, адаптивная веб-страница.*

## DEVELOPMENT OF THE MOBILE ' APPLICATION OF THE INTEGRATION OF UNIVERSAL AND SPECIALIZED SCIENCES

*Orinbetov Nurilla Turdimuratovich,  
Nukus state pedagogical institute named after Ajiniyaz,  
associate professor of the Department of technological education*

*Ismailova Momogul Hamraboykizi,  
Urganch State Pedagogical Institute is a teacher of the Department of physics, mathematics and  
technological education.*

*Abstract: in this text, special attention is paid to the analysis of methods and algorithms that can be applied in the most effective integration of web applications into the educational process, in order to ensure a deep interaction*

between Universal and specialized disciplines. It is an analysis of the advantages and disadvantages of using Flask to create mobile applications. Flask is comparable to other web frameworks such as Django and FastAPI in terms of ease of development, productivity, expansion, and community support. In addition, special attention is paid to the integration of the mobile application with modern technologies such as databases, caching systems and cloud platforms.

*Keywords:* web framework, web applications, digital technologies, Flask Django and FastAPI web frameworks, Flask microframework, cross-site scripts, encryption and Session Management, use HTML, CSS and JavaScript, modular testing, integration testing, optimization methods, adaptive web page.

Kirish. Hozirgi jadal rivojlanish va turli jarayonlarni avtomatlashtirish hamda robotlashtirish davrida dasturlashni bilish va uni o'z ish jarayonida ishlata olish texnik va pedagogik yo'nalishda ta'lim olayotgan o'quvchi-talabalar uchun juda muhim deb hisoblanadi. Bu zamonaviy mutahassilar uchun eng zaruriy talablardan biridir, desak ham bo'ladi, chunki hozirgi kunda informatika turli-tuman sohalarda muvaffaqiyatli ravishda qo'llanilishi mumkinligini hech kim ham rad eta olmaydi. Shuning uchun ham hayotimizni zamonaviy axborot texnologiyalarisiz tasavvur qilish ancha qiyin bo'lib qoldi. Internet texnologiyasi, har xil ko'rinishdagi aqlli elektron qurilmalar (shaxsiy kompyuterlar, smartfonlar, noubuklar, planshetlar va boshqalar) hayotimizning ajralmas bo'lagiga allaqachon aylanib ulgurdi.

Ta'lim jarayonida veb-ilovalardan foydalanish fanlararo aloqalarni yaratish uchun noyob imkoniyatlarni taqdim etadi, ta'lim resurslaridan foydalanishni osonlashtiradi va talabalarda tanqidiy fikrlash va mustaqil ta'limni rivojlantirishga yordam beradi. Ushbu tadqiqot doirasida umumkasbiy va ixtisoslik fanlar o'rtasidagi chuqur o'zaro aloqani ta'minlash maqsadida veb-ilovalarni o'quv jarayoniga eng samarali integratsiyalashda qo'llanilishi mumkin bo'lgan usullar va algoritmlarni tahlil qilishga alohida e'tibor qaratilgan.

Raqamli texnologiyalar hayotimizning har bir burchagiga kirib borayotgan zamonaviy dunyoda mobil ilovalar kundalik hayotning ajralmas qismiga aylanib bormoqda. Ko'proq kompaniyalar va individual ishlab chiquvchilar foydalanuvchilarning doimiy o'sib borayotgan ehtiyojlarini qondirish uchun funktsional, qulay va sezgir mobil ilovalarni yaratishga intilmoqda. Shu nuqtai nazardan, Python o'zining moslashuvchanligi va kuchli vositalar to'plami tufayli veb-ilovalarni ishlab chiqish uchun afzal qilingan dasturlash tillaridan biri sifatida ajralib turadi. Python-da yozilgan Flask veb-frameworka, veb-ilovalarni, shu jumladan mobil ilovalarni yaratish uchun engil, ammo kuchli platformani taklif etadi. Flask yordamida mobil ilovani ishlab chiqishga bag'ishlangan bo'lib, asboblar va ishlab chiqish muhitini tanlashdan boshlab asosiy funktsional talablarni amalga oshirish va ilovani sinovdan o'tkazishgacha.

Flask - bu veb-ishlab chiqish uchun mikroframework bo'lib, u ishlab chiquvchilarga o'z ilovalari uchun haqiqatan ham muhim bo'lgan narsaga e'tibor qaratish va shu bilan birga boshlash uchun zarur bo'lgan kod miqdorini kamaytirish imkonini beradi. Bu xususiyat uni mobil ilovalarni ishlab chiqish uchun ideal tanlov qiladi, bunda rivojlanish tezligi va qo'llab-quvvatlash qulayligi asosiy omillardir. Ushbu ish doirasida Flask-da mobil ilovani ishlab chiqish jarayoni batafsil ko'rib chiqiladi, jumladan, ishlab chiqish muhitini o'rnatish, dastur arxitekturasini loyihalash, foydalanuvchi interfeysi va server mantig'ini amalga oshirish va ilovaning ishlashi.

Mobil ilovalarni yaratish uchun Flask-dan foydalanishning afzalliklari va kamchiliklarini tahlil qilishdir. Flask Django va FastAPI kabi boshqa veb-frameworkalar bilan ishlab chiqish qulayligi, unumdorligi, kengaytirilishi va hamjamiyatni qo'llab-quvvatlash nuqtai nazaridan taqqoslanadi. Bundan tashqari, mobil ilovani ma'lumotlar bazalari, keshlash tizimlari va bulutli platformalar kabi zamonaviy texnologiyalar bilan integratsiyalashuviga alohida e'tibor qaratiladi.

Flask-da umumkasbiy va ixtisoslik fanlar integratsiyasining mobil ilovasini ishlab chiqish, vositalarni tanlash va ishlab chiqish muhitini sozlash bo'yicha muhokamamizni davom ettiraylik.

Avvalo, asboblar va texnologiyalarning to'g'ri to'plamini tanlash har qanday dasturiy mahsulotni ishlab chiqishda muhim qadamdir. Flask-da mobil ilovani ishlab chiqish uchun ushbu tanlov nafaqat veb-frameworkaning o'zini, balki dasturning funktsionalligini ta'minlaydigan va ish faoliyatini yaxshilaydigan qo'shimcha kutubxonalar va xizmatlarni ham o'z ichiga oladi. Ushbu bosqichda ishlab chiquvchilar ma'lumotlar bazasi, versiyalarni boshqarish tizimi, sinov va disk raskadrovka vositalari va dasturni joylashtirish platformasi haqida qaror qabul qilishlari kerak.

Rivojlanish muhitini o'rnatish Python va Flask-ni o'rnatishdan boshlanadi. Loyiha bog'liqliklarini global Python muhitidan ajratish uchun virtual muhitdan (virtualenv) foydalanish tavsiya etiladi, bu kutubxonalar o'rtasidagi ziddiyatlarni oldini oladi va qaramlikni boshqarishni soddalashtiradi. Flask va kerakli kutubxonalarni

o'rnatish pip paket menejeri yordamida amalga oshiriladi. Ma'lumotlar bazasi bilan ishlash uchun Flask kengaytmalaridan foydalanish mumkin, masalan, SQLAlchemy bilan integratsiyalashuv uchun Flask-SQLAlchemy relyatsion ma'lumotlar bazalari bilan ishlash uchun qulay interfeysni ta'minlaydi yoki MongoDB kabi NoSQL ma'lumotlar bazalari bilan ishlash uchun Flask-MongoEngine. Texnologiya to'plamini tanlagandan so'ng, keyingi qadam dastur arxitekturasini loyihalashdir. Rivojlanishga arxitektura yondashuvi asosan yakuniy mahsulotning moslashuvchanligi, kengaytirilishi va qo'llab-quvvatlash qulayligini belgilaydi. Flask mikrofreymvorkom bo'lib, qat'iy me'moriy cheklovlar qo'ymaydi, bu ishlab chiquvchilarga me'moriy uslubni tanlash erkinligini beradi. Biroq, bu aniq va samarali arxitekturani yaratish mas'uliyati ishlab chiquvchilarga tegishli ekanligini anglatadi. Ilovani mustaqil modullarga ajratishni o'z ichiga olgan modulli yondashuv, ularning har biri funktsionallikning o'ziga xos qismi uchun javob beradi, ishlab chiqish va sinovdan o'tkazishni soddalashtiradi, shuningdek, kelajakda dasturni kengaytirishni osonlashtiradi.

Xavfsizlik masalalari mobil ilovalarni ishlab chiqishda markaziy o'rinlardan birini egallaydi. Flask dastur xavfsizligini ta'minlash uchun bir nechta mexanizmlarni taqdim etadi, jumladan saytlararo so'rovlarni soxtalashtirishdan (CSRF) va saytlararo skriptlardan (XSS), ma'lumotlarni shifrlashdan va seansni boshqarishdan himoya qilish. Shaxsiy ma'lumotlarning himoyasini ta'minlash va dastur funktsiyalariga ruxsatsiz kirishni oldini olish uchun foydalanuvchi autentifikatsiyasi va avtorizatsiya tizimini diqqat bilan ko'rib chiqish muhimdir.

Foydalanuvchi interfeysi (UI) ishlab chiqish mobil ilovalarni yaratishda asosiy rol o'ynaydi, chunki foydalanuvchi dastur bilan o'zaro aloqada bo'ladi. Flask kontekstida UI ni ishlab chiqishda mobil qurilma ekranining turli o'lchamlariga moslashuvchi sezgir va intuitiv dizaynni yaratishga e'tibor qaratish lozim. HTML, CSS va JavaScript-dan foydalanish veb-interfeyslarni ishlab chiqishda standart yondashuv bo'lib, Flask Jinja2 shablon tizimi orqali ushbu texnologiyalarni server tomonidagi mantiq bilan integratsiyalashni osonlashtiradi. Jinja2 shablonlari ishlab chiquvchilarga serverda olingan ma'lumotlardan foydalangan holda dinamik HTML kontentini samarali yaratish imkonini beradi, bu esa sahifani qayta yuklamasdan interaktiv foydalanuvchi interfeyslarini yaratish imkonini beradi.

Mobil ilovalarni ishlab chiqishning muhim qismi ham ularning ishlashini ta'minlashdir. Bunga erishish uchun resurslarni minimallashtirish va siqish (CSS va JavaScript fayllari), resurslarni asinxron yuklash va keshlash kabi optimallashtirishning turli usullaridan foydalanish mumkin. Flask Flask-Assets kabi kengaytmalardan foydalanishni qo'llab-quvvatlaydi, ular statik fayllarni optimallashtirish jarayonini avtomatlashtiradi va shu bilan ilovalarni yuklash tezligini va umumiy foydalanuvchi tajribasini yaxshilaydi.

Server tomonidagi mantiqqa o'tadigan bo'lsak, Flask ishlab chiquvchilarga mijoz so'rovlarini bajarish uchun moslashuvchan marshrutizatsiya tizimini taklif qiladi. Marshrutlarni va ularga mos keladigan ishlov beruvchi funktsiyalarini aniqlash ishlab chiquvchilarga mijoz va server o'rtasidagi ma'lumotlar oqimini osongina boshqarish imkonini beradi, bu dasturda foydalanuvchining turli harakatlarini boshqarish mantiqini ta'minlaydi. Ma'lumotlar bazalari va boshqa tashqi xizmatlar bilan integratsiya Flask modullari va kengaytmalari orqali amalga oshiriladi, bu boy funktsionallikka ega murakkab veb-ilovalarni yaratishni osonlashtiradi.

Server mantig'ini amalga oshirish jarayonida dastur xavfsizligi muhim jihat bo'lib qolmoqda. Flask SQL in'ektsiyalari va CSRF hujumlari kabi keng tarqalgan tahdidlardan himoya qilish uchun o'rnatilgan mexanizmlarni taqdim etadi va Flask-Security kabi qo'shimcha kengaytmalardan foydalanish autentifikatsiya, avtorizatsiya, va foydalanuvchi sessiyasini boshqarish.

Flask yordamida mobil ilovani ishlab chiqishda test funktsionallikni rivojlantirishdan kam emas. Modulli testi, integratsiya testi va foydalanuvchi interfeysi testi ishlab chiquvchilarga xatolarni erta bosqichda aniqlash va tuzatish imkonini beradi, bu esa yuqori sifatli yakuniy mahsulotni ta'minlaydi. Flask PyTest va Flask-Testing kabi mashhur test tizimlari bilan integratsiyani qo'llab-quvvatlaydi, bu testlarni yaratish va ishga tushirish jarayonini soddalashtiradi.

Ushbu sahifada biz Flask-dagi mobil ilovaning foydalanuvchi interfeysi va server mantig'ini ishlab chiqishning asosiy jihatlarini, jumladan, ishlashni optimallashtirish, xavfsizlikni ta'minlash va sinovdan o'tkazishning ahamiyatini ko'rib chiqdik.

Ilovaning jonli efirga chiqishidan oldin uning barqarorligi va ishonchligini ta'minlashda PyTest yoki Flask-Testing kabi avtomatlashtirilgan test vositalaridan foydalanish ushbu jarayonni sezilarli darajada soddalashtirishi mumkin, bu esa ishlab chiquvchilarga alohida dastur komponentlarining ishlashini, shuningdek, ular o'rtasidagi o'zaro ta'sirlarni samarali sinab ko'rish imkonini beradi.

Sinov bosqichida foydalanuvchi interfeysining barcha elementlarining funktsionalligini, shuningdek, server tomonida amalga oshirilgan funktsionallikni tekshirishga alohida e'tibor beriladi. Modulli testlari ilovaning individual funktsiyalari va usullarining ishlashini ajratish va sinab ko'rish imkonini beradi, integratsiya testlari esa dasturning turli modullari va qatlamlari o'rtasidagi o'zaro ta'sirni baholaydi, ularning to'g'ri integratsiyasini ta'minlaydi. Ko'pincha Selenium kabi veb-UI sinovini avtomatlashtirish vositalari bilan amalga oshiriladigan UI testi ilova turli brauzerlar va qurilmalarda to'g'ri ko'rsatilishi va ishlashini ta'minlashga yordam beradi.

Funksional testdan tashqari, ishlash va xavfsizlik testlari muhim rol o'ynaydi. Ishlash testi ilovangizdagi

to'siqlarni aniqlashga yordam beradi, bu esa uni sekinlashtirishi yoki serverlar yukini oshirishi mumkin. LoadRunner yoki JMeter kabi vositalardan foydalanish ko'p sonli foydalanuvchilar tomonidan bir vaqtda kirishni taqlid qilish imkonini beradi va shu bilan ilovaning yuqori yuk sharoitida yuqori unumdorlik va barqarorlikni saqlab qolish qobiliyatini sinab ko'radi. Xavfsizlik testi tajovuzkorlar tomonidan ilova ma'lumotlari yoki funksiyalariga ruxsatsiz kirish uchun foydalanishi mumkin bo'lgan potentsial dastur zaifliklarini aniqlashga qaratilgan. OWASP ZAP yoki Burp Suite kabi vositalar ishlab chiquvchilarga dastur zaifliklari va xavfsizlik kamchiliklarini har tomonlama tekshirish imkoniyatini beradi.

Mobil ilovani ishlab chiqishning yakuniy bosqichi uni joylashtirish va qo'llab-quvvatlashdir. Joylashtirish mos hosting platformasini tanlash, server infratuzilmasini o'rnatish va ilovani nashr etishni o'z ichiga oladi. Ilovaning unumdorligi va kengaytirilishi talablariga qarab, ishlab chiquvchilar an'anaviy veb-xosting, Amazon Web Services (AWS), Google Cloud Platform (GCP) yoki Microsoft Azure kabi bulutli xizmatlar yoki xizmat sifatidagi platformalar (PaaS) o'rtasida tanlov qilishlari mumkin. Heroku yoki DigitalOcean App Platform kabi ushbu variantlarning har biri o'zining afzalliklari va kamchiliklarini, jumladan, narxdagi farqlarni, boshqaruvning qulayligi va kengaytirilishini taklif qiladi.

Ilovani ishga tushirgandan so'ng, qo'llab-quvvatlash va yangilash bosqichi boshlanadi. Ilovani muntazam yangilash, aniqlangan xatolarni tuzatish, uning funktsionalligi va xavfsizligini yaxshilash muhimdir. Foydalanuvchilarning fikr-mulohazalarini olish ushbu jarayonning asosiy jihati hisoblanadi, chunki u ilovadagi kamchiliklarni aniqlash va keyingi rivojlanish yo'nalishlarini aniqlash imkonini beradi. Jira yoki GitHub Issues kabi loyihalarni boshqarish va muammolarni kuzatish tizimlaridan foydalanish loyiha ishingizni tashkil qilishni osonlashtiradi, bu sizga vazifalarni samarali rejalashtirish, ularning bajarilishini kuzatish va ishlab chiqish guruhingiz bilan muloqot qilish imkonini beradi.

Ushbu sahifada biz Flask mobil ilovasini sinovdan o'tkazish, joylashtirish va texnik xizmat ko'rsatishning asosiy jihatlarni ko'rib chiqdik. Ushbu bosqichlar ilovalarni ishlab chiqish hayotiy tsiklining ajralmas qismi bo'lib, uning sifati, ishonchliligi va foydalanuvchi kutganlarini qondirishni ta'minlaydi.

Flaskda mobil ilovani ishlab chiqishning navbatdagi muhim jihati o'tsak, biz ilovani optimallashtirish va uni bozorda ilgari surish strategiyalariga e'tibor qaratamiz.

Ilovani optimallashtirish uning ishlashi, tezligi va ilovaning umumiy foydalanuvchi tajribasiga ta'sir qiluvchi asosiy omil hisoblanadi. Flask yordamida ishlab chiqilgan mobil ilova kontekstida optimallashtirish sahifani yuklash vaqtini yaxshilash, server samaradorligini oshirish va foydalanuvchining ilova bilan o'zaro aloqasidagi kechikishlarni minimallashtirishga qaratilgan bir qator harakatlarni o'z ichiga olishi mumkin.

Optimallashtirish usullaridan biri server tomonida va mijoz tomonida keshlashdan foydalanishdir. Keshlash tez-tez o'zgaraydigan ma'lumotlar uchun serverga so'rovlar sonini kamaytirishga imkon beradi va shu bilan serverdagi yukni kamaytiradi va foydalanuvchi uchun sahifa yuklanishini tezlashtiradi. Flask Flask-Caching kabi bir nechta keshlash kengaytmalarini taklif etadi, ular ilovangizga osongina birlashtiriladi va keshni boshqarish uchun moslashuvchan sozlamalarni ta'minlaydi.

Optimallashtirishning yana bir jihati statik fayllarni (CSS va JavaScript) kichiklashtirish va birlashtirishdir, bu sahifani yuklash uchun talab qilinadigan HTTP so'rovlari sonini kamaytiradi va uzatiladigan ma'lumotlarning umumiy hajmini kamaytiradi. Ushbu jarayonlarni avtomatlashtirish uchun Webpack yoki Gulp kabi vositalar Flask bilan birgalikda ishlatilishi mumkin.

Adaptiv veb-sahifa dizayni optimallashtirishda ham muhim rol o'ynaydi, bu dasturni turli qurilmalar va ekran o'lchamlarida to'g'ri ko'rsatishi va ishlashiga imkon beradi. Bootstrap yoki Foundation kabi CSS freymvorkdan foydalanish sezgir dizaynni ancha osonlashtirishi mumkin.

Ilovangizni optimallashtirgandan so'ng, keyingi qadam uni targ'ib qilishdir. Ilovaning sifati qanday bo'lishidan qat'i nazar, samarali marketing strategiyasi bo'lmasa, u boshqa ilovalarning olomoniga e'tibor bermay qolishi mumkin. Rag'batlantirish ijtimoiy tarmoqlardan foydalanish, kontekstli reklama, qidiruv tizimlarida ilovaning ko'rinishini oshirish uchun SEO optimallashtirishni, shuningdek, turli platformalar va forumlarda foydalanuvchi sharhlari bilan ishlashni o'z ichiga olishi mumkin.

Ilovani optimallashtirishning muhim jihatlaridan biri ma'lumotlar bazasidan samarali foydalanishdir. Bunga ma'lumotlar bazasi sxemasini to'g'ri loyihalash, ma'lumotlarni qidirishni tezlashtirish uchun indekslardan foydalanish va so'rovlarni bajarish vaqtini minimallashtirish uchun so'rovlarni optimallashtirish kiradi. Ishlab chiquvchilar qiynchiliklarni aniqlash va hal qilish uchun profillash vositalaridan foydalangan holda ma'lumotlar bazasi ishlashini muntazam ravishda tahlil qilishlari kerak. Tez-tez foydalaniladigan ma'lumotlarni keshlash ma'lumotlar bazasi qo'ng'iroqlari sonini kamaytirish orqali dastur tezligini sezilarli darajada oshirishi mumkin.

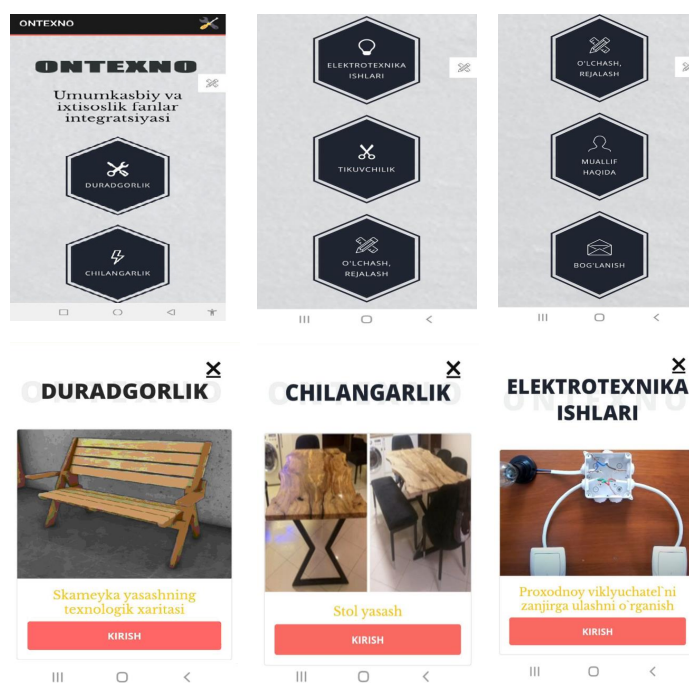
Sahifani yuklash vaqti foydalanuvchi tajribasi uchun muhim omil, ayniqsa mobil veb-ilovalarda. Uni kamaytirish uchun siz statik fayllarni (CSS va JavaScript) siqish va birlashtirish, resurslarni asinxron yuklash va tasvirlarni optimallashtirish usullaridan foydalanishingiz mumkin. WebP kabi zamonaviy tasvir formatlaridan foydalanish sifatini saqlab qolgan holda tasvir hajmini sezilarli darajada kamaytirishi va sahifalarning tezroq yuklanishini ta'minlaydi. Bundan tashqari, tasvirlar va videolar uchun Lazy Loading texnologiyasidan

foydalanish ushbu elementlarning yuklanishini foydalanuvchiga ko'ringuncha kechiktirishga imkon beradi va shu bilan sahifaning dastlabki yuklanishini tezlashtiradi.

Foydalanuvchi tomonidan ishlashni yaxshilash uchun ishlab chiquvchilar DOMni samarali yangilashni va komponentga asoslangan yondashuvdan foydalangan holda yaxshilangan foydalanuvchi tajribasini ta'minlaydigan React yoki Vue.js kabi zamonaviy JavaScript freymvorki va kutubxonalaridan foydalanishlari mumkin. Qimmatbaho operatsiyalar va xotira oqishiga yo'l qo'ymaslik uchun JavaScript kodining ishlashiga ham e'tibor berish kerak.

Ilovani turli qurilmalarda to'g'ri ko'rsatish va ulardan foydalanish qulayligini ta'minlash uchun sezgir dizayn zarur. Adaptiv tartiblar, media so'rovlar va sezgir tasvirlardan foydalanish ilova interfeysini ekran o'lchamlari va qurilma yo'nalishiga moslashtirish orqali ushbu maqsadga erishishga yordam beradi. Bu, ayniqsa, mobil ilovalar uchun juda muhimdir, bu erda foydalanuvchilar turli xil ekran o'lchamlariga ega bo'lgan keng turdagi qurilmalardan foydalanishlari mumkin.

Muntazam ishlash testlari optimallashtirish jarayonining ajralmas qismidir. Google PageSpeed Insights, Lighthouse va WebPageTest kabi vositalardan foydalanish sahifani yuklash tezligini, manba samaradorligini va ilovalar mavjudligini baholashga yordam beradi. Ushbu vositalar ishlab chiquvchilar o'z ilovalarini optimallashtirish uchun foydalanishi mumkin bo'lgan maxsus ishlash tavsiyalarini beradi.



1-rasm. ONTEXNO. Umumkasbiy va ixtisoslik fanlar integratsiyasining mobil ilovasi

Xulosa qilib aytganda, veb-ilovalarni o'quv jarayoniga integratsiyalash ta'lim sifatini oshirish, o'quvchilarning motivatsiyasi va faolligini oshirish, tanqidiy fikrlash va muammolarni hal qilish ko'nikmalarini rivojlantirish uchun kuchli vositadir. Veb-ilovalar turli ehtiyoj va imtiyozlarga ega bo'lgan talabalar uchun ochiq bo'lgan yanada moslashuvchan va inklyuziv ta'lim muhitini yaratishga yordam beradi.

#### FOYDALANILGAN ADABIYOTLAR

Ayupov Ravshan Hamdamovich. Python dasturlash tili asoslari//o'quv-uslubiy qo'llanma. Toshkent: 2020 yil. – 118 b.

Трофимов В. В. , Ильина О. П. , В. И. Кияев, Трофимова Е. В. Информационные технологии// учебник для вузов.- Москва : Издательство Юрайт, 2024. - 546 с.

3. <https://blog.skillfactory.ru>

4. <https://www.calltouch.ru>