**OʻZBEKISTON RESPUBLIKASI**

**OLIY TA’LIM, FAN VA INNOVATSIYALAR VAZIRLIGI**

**SHAROF RASHIDOV NOMIDAGI**

**SAMARQAND DAVLAT UNIVERSITETI**

**INTELLEKTUAL TIZIMLAR VA KOMPYUTER TEXNOLOGIYALAR FAKULTETI**

**“Dasturiy injiniring” kafedrasi**

**QOBULOV ASROR RAXMAT OʻGʻLI**

**MAVZU: METROLOGIYA VA QIYOSLASH MCHJNING ISH FAOLIYATINI TASHKIL ETISHNING MOBIL ILOVASI LOYIHALASH**

**«5330600 – Dasturiy injiniring» ta’lim yoʻnalishi**

**boʻyicha bakalavr darajasini olish uchun**

**BITIRUV MALAKAVIY ISHI**

Bajardi Qobulov A.

Ilmiy rahbar Nazarov U.

Ushbu ish kafedraning 2023 yil \_\_ iyundagi yigʻilishida muhokama qilindi va himoya qilishga № \_\_\_ bayonnoma bilan tavsiya etildi.

Kafedra mudiri dots. O. R. Yusupov

Fakultet dekani dots. F. M. Nazarov

Ushbu ish DAKning 2024 yil «\_\_\_» \_\_\_\_\_\_dagi yigʻilishida muhokama qilindi va №\_\_\_ bayonnoma bilan «\_\_\_\_» ballga baholandi.

DAK raisi \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Komissiya a’zolari \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Samarqand – 2024

**MUNDARIJA**

[**KIRISH** 3](#_Toc168926207)

[**I BOB. “METROLOGIYA, STANDARTLASHTIRISH VA SERTIFIKATLASHTIRISH” SOHASINING MAQSAD VA VAZIFALARI** 4](#_Toc168926208)

[1.1. Axborot texnologiyalari va kommunikasiya sohasida metrologik xizmat 4](#_Toc168926209)

[1.2. Metrologiyada sertifikatlashtirish va uning turlari 11](#_Toc168926210)

[**II BOB. MOBIL ILOVANI LOYIHALASH UCHUN JAVASCRIPTGA MO‘LJALLANGAN FREYMVORKLAR** 14](#_Toc168926211)

[2.1. Mobil platformalar va ularning turlari 14](#_Toc168926212)

[2.2. JavaScriptga mo‘ljallangan freymvorklar 17](#_Toc168926213)

[2.3. React Native framevorki va Expo komponentalaridan foydalanish 21](#_Toc168926214)

[**III BOB. DASTURIY TA’MINOTNI LOYIHALASH VA ISHLAB CHIQISH** 29](#_Toc168926215)

[3.1. Dasturiy ta’minotning mantiqiy loyihasi (jadvallar, relatsion model) API 29](#_Toc168926216)

[3.2. Mobil ilovaning dizayn qismini loyhalash 35](#_Toc168926217)

[3.3. Dasturiy ta’minotning fizik yaratilishi 37](#_Toc168926218)

[**XULOSA** 40](#_Toc168926219)

[**FOYDALANILGAN ADABIYOTLAR** 41](#_Toc168926220)

[**ILOVALAR** 42](#_Toc168926221)

# **KIRISH**

Zamonaviy axborot va kommunikatsiya texnologiyalari sohasida turli metrologik, standartlashtirish sifat va sifat boshqaruvi, sertifikatlashtirish bo‘yicha masalalar bilan shug‘ullanish, hamda me’yoriy hujjatlar va standartlar bilan ishlash borasida yo‘nalishga mos yetarli bilim, ko‘nikma va malakalarini shakllantirishdir. Shuning uchun o‘lchashlar birliligini ta’minlash, standartlashtirish va sertifikatlashtirish tizimlarini o‘rganish dolzarb masalalardan biri hisoblanadi.

So‘nggi yillarda axborot texnologiyasi kirib bormagan sohalar soni tobora kamayib bormoqda. Shu jumladan metrologiya sohasida ham bu o‘zgarishlar yaqqol ko‘zga tashlanmoqda. Metrologiya sohasi oʻlchovlar, ularning bir xilligini taʼminlash usullari va vositalari hamda talab etilgan aniqlikka erishish yoʻllari haqidagi fandir Mazkur soha tabiiy fanlarni o‘zlashtirishda, keyinchalik esa ishlab chiqarish, loyihalash va tadqiqot ishlarida kerak bo‘ladigan asosiy negiz tushunchalarni o‘rganadi. Axborot tizimlar orqali ushbu sohadagi jarayonlarni raqamlashtirishimiz, qoʻlda bajarilayotgan murakkab masalalarni avtomatlashtirishimiz mumkin.

Metrologiyada mahsulot (buyum, tovar) yoki xizmat muayyan standartga yoki texnikaviy shartlarga muvofiq kelishini tasdiqlash maqsadida oʻtkaziladigan faoliyat ham koʻrsatiladi. Ushbu faoliyat sertifikatlashtirish deb ataladi. Sertifikatlashtirish faoliyati natijasida mahsulot (buyum, tovar)ning sifati haqida iste’molchini ishontiradigan tegishli hujjat - sertifikat beriladi. Sertifikatlashtirish jarayonining qoʻlda bajarilishi murakkab boʻlib, ushbu vazifa koʻp vaqtni talab qiladi.

Ushbu bitiruv malakaviy ishda “Metrologiya va qiyoslash MCHJning ish faoliyatini tashkil etishning mobil ilovasi loyihalash” masalasi ko‘rib chiqilgan. Bitiruv malakaviy ishi natijasida MCHJning ish faoliyati o‘rganilgan hamda uning sertifikatlash tizimining murakkabliklari tahlil qilinib, tahlilllar asosida veb ilova ishlab chiqilgan.

**Ishning maqsadi.** Bitiruv malakaviy ishining maqsadi metrologiya va qiyoslash MCHJning ish faoliyatini tashkil etishning mobil ilovasini ishlab chiqishdan iborat.

**Ishning ilmiy va amaliy ahamiyati.** Bitiruv malakaviy ishining ilmiy ahamiyati metrologiya xizmati haqida koʻnikmalarga ega boʻlish, axborot tizimlarining integratsiya bosqichlari haqida tasavvur hosil qilish, maʼlumotlar xavfsizligini taʼminlash uchun amaliy ishlarni oʻrganib chiqish vazifalarini shakllantiradi.

**Bitiruv malakaviy ish strukturasi.** Ish uchta bob, xulosa, adabiyotlar roʻyxati va ilovalardan iborat.

**Birinchi bobda** “metrologiya, standartlashtirish va sertifikatlashtirish” sohasining maqsad va vazifalari tariflab berilgan. Bunda axborot komunakatisya sohasida metrologik xizmatlar va sertifikatlar turlari haqida ma’lumotlar berilgan.

**Ikkinchi bob** mobil pltformalar va ularning turlari. Mobil ilovayaratish yondashuvlari va ilovani loyihalash uchun JavaScriptga mo‘ljallangan freymvorklari haqida ma’lumotlar berilgan.

**Uchinchi bob** metrologiya sohasida sertiftikatlashtirish ma’lumotlarini tashkil qilish va ma’lumotlarni tahlil qilishning dasturiy ta’minotining mobil ilovasini ishlab chiqish jarayoni, dasturning dizayni va uni shakllantirish, dasturiy ta’minotning mantiqiy strukturasi va fizik jihatdan yaratilishi, dastur yordamida hujjatlar bilan ishlash, sertifikatlar ishlab chiqish jarayonları ko‘rsatilgan.

Xulosada dastur asosiy maqsadi, mazmuni va takliflar, adabiyotlar roʻyxatida qoʻllanilgan materiallar va internet-resurslar, ilovalarda esa, boshqaruvchi va qoʻshimcha dasturiy modullar matnlari keltirilgan.

# **I BOB. “METROLOGIYA, STANDARTLASHTIRISH VA SERTIFIKATLASHTIRISH” SOHASINING MAQSAD VA VAZIFALARI**

* 1. **Axborot texnologiyalari va kommunikasiya sohasida metrologik xizmat**

“Metrologiya, standartlashtirish va sertifikatlashtirish” fanini o‘rganishdan maqsad: talabalarda Aloqa, axborotlashtirish va telekommunikasiya texnologiyalari sohasida telekommunikasiya xizmatlarini ko‘rsatish bilan bog‘liq bo‘lgan metrologik ta'minot, sohaga ISO 9000:2000 standartlari asosida sifat menejmenti tizimini joriy qilish, standartlashtirish ishlari va sertifikatlashtirish muolajalari bo‘yicha masalalar bilan shug‘ullanish, hamda me'yoriy-texnik hujjatlar va soha standartlari bilan ishlash borasida etarli bilim va malakalarni hosil qilishdir. Asosiy vazifasi esa talabalarga uzluksiz ta'lim tizimida “Metrologiya, standartlashtirish va sertifikatlashtirish” asoslari bo‘yicha mukammal bilim berishdir [2].

Ushbu fan namunaviy o‘quv rejasining umumkasbiy fanlar blokiga kiritilgan bo‘lib, bakalavriatning barcha ta'lim yo‘nalishlari bo‘yicha ta'lim oluvchi talabalar uchun mo‘ljallangan. Fanni o‘rganish natijasida talabalar metrologiya bo‘yicha asosiy qoidalarni, talablar va me'yorlarni, standartlashtirish va sertifikatlashtirish bo‘yicha me'yoriy hujjatlar bilan ishlashni bilishi va mavjud bilimlarini, tajribalarini amaliy faoliyatda qo‘llay bilishi lozim hisoblanadi. Bu hozirgi kunda, ayniqsa, xalqaro standartlarning talablariga mos keluvchi mahsulotlarni ishlab chiqarish va uning raqobatbardoshliligini ta'minlashda dolzarb masalalardan biri sanaladi.

Standartlashtirish, metrologiya va sertifikatlashtirish bo‘yicha asosiy maqsad mahsulotlar va xizmatlar sifati, havfsizligi va raqobatbardoshliligini ta'minlashdir.

Standartlashtirishning afzallik jihatlari. Uni shunday ta'riflash mumkin, standartlashtirish iqtisodiyotni maksimallikka erishishiga qaratilgan. Standartlar jamiyatning turli jabhalarida bir qancha afzalliklar olib kiradi.

Texnologlar uchun standartlar:

Tovarlar va xizmatlarni yanada takomillashtirish uchun ilmiy tadqiqotlar va rivojlantirishning yangi bosqichlarini shakllanishida xizmat qiladi.

Standartlarning o‘ziga xos xususiyatlari.

Standart asosan uchta xususiyatga ega bo‘ladi:

Darajaviy: kompaniyada milliy va xalqaro darajalar singari

Ilmiy: oziq-ovqat, tekstil va menejment injiniringi kabi

Aspekt: qadoqlash va yorliqlashtirish, sinovdan o‘tkazish va analizlash, o‘ziga xos xususiyatlarini aniqlashtirishga o‘xshagan.

O‘zbekiston Respublikasining «Metrologiya to‘g‘risida»gi Qonuni (11-modda) tomonidan xo‘jalik yurituvchi sub'ektlarning metrologik xizmatlari o‘lchash birliligini ta'minlash va metrologik nazoratni amalga oshirish bo‘yicha ishlarni bajarish uchun zarur bo‘lgan hollarda tashkil etilishi belgilab qo‘yilgan.

Milliy Standartlar Tuzilmasi (MST). Milliy darajada standartlarni tayyorlash, ular bilan bog‘liq barcha masalalar MST orqali amalga oshiriladi. Ko‘pgina mamlakatlar MSTni tashkilotlar yoki institutlar deb yuritishadi. Dunyo bo‘yicha ko‘plab ISO tashkiloti a’zolari o‘zlarining MSTsiga ega; biroq ko‘plab mamlakatlar MSTga ega emas, ushbu mamlakatlar ISO tashkilotining standartlari va qonunlaridan foydalanadi. Hozitgi kunda dunyo bo‘yicha 148 mamlakat ISO tashkiloti a’zolari hisoblanadi.

Eng ko‘p rivojlangan mamlakatlar 1917 va 1925 yillar oralig‘ida 15 ta MST joriy etildi. Germaniya ular orasida birinchi bo‘lib o‘zining MSTsini tashkil etdi. Keyinchalik shu vaqtlar oralig‘ida Buyuk Britaniya, Amerika Qo‘shma Shtatlari, Belgiya, Kanada, Niderlandiya, Shvetsariya va Austriya ham o‘zlarining MSTni tashkil etishdi.

MST quiyidagi asosiy funksiyalarni o‘z ichiga oladi:

• Milliy standartlarni tayyorlash va chop etish;

• Sanoatda standartlarning amalga oshirilishini ta’minlash;

• Mahsulotlarni sertifikatlashtirish;

• Texnika masalalariga bog‘liq bo‘lgan masalalar va bular milliy hamda xalqaro standartlar bilan o‘zaro muvofiqligini ta’minlash haqidagi ma’lulotlarni e’lon qilish;

• Standartlar bilan aloqador forumlarda va xalqaro faoliyatlarda mamlakat ishtirokini nazorat qilish;

Hozirgi vaqtda aloqa, axborotlashtirish va telekommunikasiya texnologiyalari sohasida o‘lchashlar birliligini ta'minlash tizimida o‘lchash vositalarini qiyoslash, kalibrlash, ta'mirlash va metrologik shahodatlash huquqiga ega bo‘lgan uchta akkreditlangan metrologiya xizmatlari mavjud: O‘zbekiston aloqa va axborotlashtirish agentligi UNICON.UZ markazi qoshidagi Asos metrologiya xizmati, Radioaloqa, radioeshittirish va televidenie markazi hamda «O‘zbektelekom» AK «Shaharlararo aloqa korxonasi» filialidagi metrologiya xizmatlari.

UNICON.UZ markazi qoshidagi Asos metrologiya xizmatining qiyoslash va ta'mirlash laboratoriyasi sohaviy hisoblanadi va u sohadagi barcha xo‘jalik yurituvchi sub'ektlar uchun ta'mirlash va qiyoslash ishlarini o‘tkazish huquqiga ega. Oxirgi ikkita xizmat akkreditlash sohasiga muvofiq xo‘jalik yurituvchi sub'ektlarning ichki ehtiyojlari uchun zarur bo‘lgan o‘lchash vositalarini ta'mirlash va qiyoslash huquqiga egadirlar. Ushbu metrologik xizmatlari o‘z faoliyatlarini amalga oshirishlari uchun namunaviy o‘lchash vositalari, yordamchi uskunalar va normativtexnik hujjatlar bilan ta'minlanganlar.

Hozirgi vaqtda, metrologiya xizmatlari yo‘q bo‘lgan (o‘lchash vositalarining to‘plami kam bo‘lganligi sababli), aloqa, axborotlashtirish va telekommunikasiya texnologiyalari sohasida xo‘jalik yurituvchi sub'ektlarida o‘lchash vositalari holati, tegishli hujjatlarda belgilangan vazifa va majburiyatlari uchun javobgarlar tayinlangan.

Barcha metrologik xizmatlarga va o‘lchash vositalari holati uchun javobgar shaxslarga asosiy talablar - bu o‘lchash vositalarini, qiyoslash va ta'mirlash ishlarini o‘z vaqtida o‘tkazish.

Xo‘jalik yurituvchi sub'ektlarning metrologik xizmatlariga qiyoslash laboratoriyalari bilan bir qatorda davlat metrologiya nazorati va tekshiruviga tegishli bo‘lgan sohadan tashqaridagi o‘lchov vositalari uchun kalibrlash laboratoriyalarini tuzish maqsadga muvofiqdir. Aloqa, axborotlashtirish va telekommunikasiya texnologiyalari sohasida o‘lchash vositalarini kalibrlashning tashkiliy va texnik asoslarini joriy qilish o‘lchash vositalari holatini nazorat qilishni osonlashtirishga imkon beradi.

“O‘zbektelekom” AK, «O‘zbekiston pochtasi» DAK, uyali aloqa kompaniyalari, Internetning yirik operator va provayderlarining qayta tashkil qilinishi sababli ularning tarkibiy bo‘linmalari va filiallarini metrologik ta'minlashda muammolar tug‘ildi.

Tabiiyki, akkreditlashning qisqa sohasi bilan chegaralangan mavjud uchta metrologik xizmatlarning faoliyati yangi tuzilgan tarkiblar va bo‘linmalarni ta'mirlash va qiyoslash ishlari bilan to‘liq ta'minlay olmaydi.

Bu muammolarni hal qilish uchun va namunaviy o‘lchash vositalarini ta'mirlash va qiyoslash laboratoriyalarini yuqori texnologiya uskunalari bilan jihozlash yo‘li bilan mavjud metrologiya xizmatlarining akkreditlash sohasini izchillik bilan kengaytirish va mutaxassislarni yangi texnika hamda texnologiyalar bilan ishlashning zamonaviy uslublari va ishlash malakalarini oshirish uchun muntazam ravishda o‘qitish zarur.

UNICON.UZ markazi qoshidagi Asos metrologiya xizmati aloqa va axborotlashtirish sohasidagi xo‘jalik yurituvchi sub'ektlarning metrologik xizmatlari ishlarini muvofiqlashtiradi.

Aloqa, axborotlashtirish va telekommunikasiya texnologiyalari sohasining xo‘jalik yurituvchi sub’ektlarning metrologiya xizmatlari o‘lchashlar birliligi va talab etilgan aniqligini (o‘lchashlarni bajarish uslublarini ishlab chiqish va attestatlash, normativ hujjatlar loyihalarining metrologik ekspertizasi, loyihalash, konstruktorlik va texnologik hujjatlashtirish hamda boshqa turdagi ishlar bo‘yicha) ta’minlash sohasida muayyan faoliyatni amalga oshirishda texnik vakolatliligini tan olishga, ixtiyoriy ravishda akkreditasiyadan o‘tishi mumkin.

Metrologik ta’minotning tarkibiga O‘zbekiston aloqa va axborotlashtirish agentligi, UNICON.UZ markazi qoshidagi Asos metrologiya xizmati, Telekommunikasiyalar va pochta aloqasi sohasidagi standartlashtirish bo‘yicha Texnika qo‘mitasi, Davlat aloqa inspeksiyasi, metrologik xizmatlar va xo‘jalik yurituvchi sub’ektlari o‘lchash vositalarining holati uchun mas'ul shaxslar kiradi. Metrologik ta'minot tizimi O‘zbekiston Respublikasi Davlat o‘lchashlar birliligini ta’minlash tizimi bilan, O‘zbekiston Respublikasi standartlashtirish Davlat tizimi, standartlashtirish va sertifikatlashtirish soha tizimlari bilan o‘zaro hamkorlik qiladi.

Tizimning boshqa davlatlarning o‘lchashlar birliligini ta'minlash tizimlari va xalqaro organlar bilan o‘zaro hamkorligi O‘zbekiston Respublikasining amaldagi qonun hujjatlariga, O‘zbekiston aloqa va axborotlashtirish agentligi, «O‘zstandart» agentligining normativ hujjatlariga muvofiq tartibga solinadi.

Davlat metrologik nazorati va tekshiruvi sohasida qo‘llanilayotgan o‘lchash vositalarining Davlat Reestrini, metrologik ta'minot bo‘yicha standartlar Davlat Reestrini, akkreditlangan metrologik xizmatlar va metrologik laboratoriyalar Davlat Reestrini yuritish, Tizim bo‘yicha tashkiliyuslubiy hujjatlarning kelishuvi, boshqa davlatlarning o‘lchashlar birliligini ta'minlash Davlat tizimlari bilan o‘zaro hamkorlik qilish, shu jumladan, sinovlar natijalarini o‘zaro tan olish, tur tasdiqlanishining sertifikatlari va metrologik shahodatlash masalalari bo‘yicha, shuningdek o‘lchash vositalarini qiyoslash uslublarini, o‘lchash vositalari ustidan davlat metrologik nazorati va tekshiruvini amalga oshirish vazifalari «O‘zstandart» agentligi zimmasiga yuklatilgan.

−Metrologik xizmatlarni ko‘rsatish bo‘yicha respublika Markazi:

−o‘lchash vositalari turini tasdiqlash bo‘yicha sinovlarni o‘tkazish;

−o‘lchash vositalarini metrologik shahodatlash, o‘lchash vositalarini qiyoslash;

−metrologik xizmatlari, markazlari, laboratoriyalari o‘lchash vositalarining sinovlari va qiyoslashlarini o‘tkazish huquqini beruvchi akkreditlashni tashkil etish va o‘tkazish;

- o‘lchashlarni bajarish uslublari va metrologik faoliyatning boshqa muayyan turlarining davlat metrologik nazoratini amalga oshiradi.

Standartlashtirish, metrologiya va sertifikatlashtirish ilmiy-tadqiqot instituti belgilangan ixtisosliklar bo‘yicha kadrlar tayyorlash (qayta tayyorlash)ni ta'minlaydi va qonunlashtiruvchi metrologiya bo‘yicha ishlab chiqilayotgan soha normativ hujjatlarining kelishuvida qatnashadi, metrologik xizmatlari va sinov laboratoriyalarini akkreditlash bo‘yicha hujjatlar ekspertizasini o‘gkazadi.

Aloqa, axborotlashtirish va telekommunikasiya texnologiyalari sohasida metrologik ta'minot masalalari O‘zbekiston aloqa va axborotlashtirish agentligi tomonidan tartibga solinadi va muvofiqlashtiriladi.

O‘zbekiston Aloqa, axborotlashtirish va telekommunikasiya texnologiyalari davlat qo‘mitasining asosiy vazifalari quyidagilar hisoblanadi:

- Metrologiya qonun hujjatlari, o‘lchov vositalari turining attestasiyasi va tan olinishi masalalari bo‘yicha «O‘zstandart» agentligi bilan o‘zaro hamkorlik, Tizim tarkibini shakllantirish va uning qatnashchilarining faoliyatini koordinatlash bo‘yicha ishlarni tashkil etish;

- Aloqa sohasidagi mintaqaviy hamdo‘stlik va boshqa xalqaro tashkilotlar bilan Tizimning qoida va me'yorlarini rivojlantirish hamda uyg‘unlashtirish masalalari bo‘yicha o‘zaro hamkorlik qilish;

- umumsoha xarakteridagi masalalarni hal etish, aloqa va axborotlashtirish sohasida o‘lchashlar birliligini ta'minlashda texnik siyosatni ishlab chiqish; - soha metrologik ta'minotining qonun hujjatlari bazasini ishlab chiqish bo‘yicha ishlarni tashkil qilish.

O‘zbekiston aloqa va axborotlashtirish Agentligi qoshidagi standartlashtirish bo‘yicha Texnik qo‘mita quyidagilarni ishlab chiqadi:

- metrologiya qonun hujjatlari hamda aloqa va axborotlashtirish sohasi xo‘jalik yurituvchi sub'ektlarining metrologik ta'minoti borasidagi strategiyani;

- xalqaro standartlar talablari bilan uyg‘unlashtirilgan normativ hujjatlarni ishlab chiqish, amaldagi normativ hujjatlarga o‘zgartirishlar kiritish yoki ularni bekor qilish, chet el standartlaridan Tizimni takomillashtirish maqsadida foydalanish bo‘yicha tavsiyalarni tayyorlaydi;

- aloqa, axborotlashtirish va telekommunikasiya texnologiyalari sohasida metrologik xizmatlarning tashkiliy sxemasini takomillashtirish va shakllantirish bo‘yicha tavsiyalarni tayyorlaydi.

O‘zbekiston Respublikasi Aloqa vazirligi (hozirda O‘zbekiston aloqa va axborotlashtirish agentligi)ning 1997 yil 11 iyuldagi 225-son buyrug‘iga asosan UNICON.UZ markazi qoshida Asos metrologiya xizmati tashkil qilindi.

Asos metrologiya xizmati Aloqa, axborotlashtirish va telekommunikasiya texnologiyalari sohasining xo‘jalik yurituvchi sub'ektlarini metrologik ta'minoti vazifalarini amalga oshirish bo‘yicha ishlarga ilmiy-texnik va tashkiliyuslubiy rahbarlikni amalga oshiradi. Asos metrologiya xizmati nizomga muvofiq o‘lchash vositalarining soha reestrini yuritadi, « Aloqa, axborotlashtirish va telekommunikasiya texnologiyalari sohasida qo‘llash uchun tavsiya qilingan o‘lchash vositalari katalogi»ni har yili yangilaydi.

Aloqa, axborotlashtirish va telekommunikasiya texnologiyalari sohasidagi Radioaloqa, radioeshittirish va televidenie markazi qoshida faoliyat ko‘rsatmoqda.

Qonun hujjatlari va normativ aktlar, davlat va soha standartlari talablarini bajarish va ularga rioya qilish, lisenziya shartlari va aloqa va axborotlashtirish sohasida taqdim etilayotgan xizmatlar sifatini ta'minlash bo‘yicha Davlat nazorat organi bo‘lib Davlat aloqa inspeksiyasi hisoblanadi.

Tizim ishtirokchilarining asosiy vazifalari ular to‘g‘risidagi belgilangan tartibda kelishilgan va tasdiqlangan nizomlarga muvofiq aniqlanadi.

## **1.2. Metrologiyada sertifikatlashtirish va uning turlari**

Sertifikatlashtirish deganda, mahsulot (buyum, tovar) yoki xizmat muayyan standartga yoki texnikaviy shartlarga muvofiq kelishini tasdiqlash maqsadida o‘tkaziladigan faoliyat tushunilib, ushbu faoliyat natijasida mahsulot (buyum, tovar)ning sifati haqida iste’molchini ishontiradigan tegishli hujjat - sertifikat beriladi.

Sinash – bu o‘rnatilgan muolajaga muvofiq holda bir yoki bir nechta xarakteristikani aniqlash.

Sinash laboratoriyasi – sinashlarni o‘tkazadigan laboratoriya.

Akkreditlash tizimi – Akkreditlashni o‘tkazish uchun protseduralar va boshqaruvning o‘z qoidalariga ega bo‘lgan tizim.

Akkreditlash – Protsedura bo‘lib, uning vositasida vakolatli idora shaxs yoki idoraning muayyan ishni bajarish huquqiga ega ekanligini rasmiy tan oladi.

Akkreditlash bo‘yicha idora – Akkreditlash tizimini boshqaruvchi va akkreditlashni o‘tkazuvchi idora.

Inspektsiya tekshiruvi – sertifikatlashtirish va akkreditlashda o‘rnatilgan talablarga muvofiqligini tasdiqlash maqsadida sertifikatlashtirilgan mahsulot, sifat yoki ishlab chiqarishni boshqarish tizimlari, sertifikatlashtirish bo‘yicha idoralar, sinash laboratoriyalari (markazlari) ning faoliyatini takroriy baholash protsedurasi.

Sifat – O‘z tafsilotlari majmuining talablarga muvofiqlik darajasi.

Sifat menejmenti tizimi – Sifatga nisbatan tashkilotga rahbarlik qilish va boshqarish uchun menejment tizimi.

Metrologiyada sertifikatlar o‘lchov vositalarining aniqligi va ishonchliligini ta'minlash uchun muhim ahamiyatga ega. Ular turli maqsadlar uchun qo‘llaniladi va har xil turlarga ega. Quyida metrologiyadagi asosiy sertifikat turlari haqida ma'lumot berilgan:

1. Kalibrlash Sertifikati

Kalibrlash sertifikati o‘lchov vositasining o‘lchash natijalari standartlarga mos kelishini tasdiqlaydi. Bu sertifikat:

- Aniqlikni ta'minlash: O‘lchov vositasining standartlar va me'yorlarga mosligini tekshiradi.

- Qonuniy talablarga javob berish: Ba'zi sohalarda qonuniy yoki sanoat talablari bo‘yicha zarur bo‘ladi.

- Rejalashtirilgan texnik xizmat ko‘rsatish: O‘lchov vositalarini muntazam ravishda kalibrlash kerak bo‘lgan holatlarda qo‘llaniladi.

2. Sinov Sertifikati

Sinov sertifikati mahsulot yoki asbobning sinovlardan o‘tganligini va belgilangan me'yorlarga mos kelishini tasdiqlaydi. Bu turdagi sertifikat:

- Maxsus sinov usullariga moslik: Sinovlar maxsus usullar va standartlarga muvofiq amalga oshiriladi.

- Sifatni ta'minlash: Ishonchli va aniq sinov natijalariga ega bo‘lishni kafolatlaydi.

3. Muvofiqlik Sertifikati

Muvofiqlik sertifikati mahsulot yoki xizmatning belgilangan standartlarga va me'yorlarga muvofiqligini tasdiqlaydi. Bu sertifikat:

- Qonuniy va sanoat talablari: Muvofiqlikni isbotlaydi va bozorda ishonchni oshiradi.

- Xalqaro standartlar: ISO kabi xalqaro standartlarga muvofiqlikni tasdiqlashi mumkin.

4. Yillik Tekshirish Sertifikati

Yillik tekshirish sertifikati o‘lchov vositalarining muntazam ravishda tekshirilganligini tasdiqlaydi. Bu sertifikat:

- Doimiy nazorat: Yilda bir marta yoki belgilangan vaqt oralig‘ida o‘tkaziladi.

- Barqarorlik va ishonchlilik: O‘lchov vositasining barqaror ishlashini ta'minlaydi.

5. Kalibrlash Laboratoriyasi Akkreditatsiya Sertifikati

Bu sertifikat kalibrlash laboratoriyasining malakali va ishonchli ekanligini tasdiqlaydi. Bu turdagi sertifikat:

- Laboratoriya sifatini ta'minlash: Kalibrlash ishlarini bajarish uchun zarur bo‘lgan talab va standartlarga muvofiqlikni tekshiradi.

- Ishonchlilikni oshirish: Laboratoriya xizmatlaridan foydalanuvchilarga ishonch beradi.

6. Izlanish va Sinovlar uchun O‘lchov Vositasi Sertifikati

Izlanish va sinovlar uchun mo‘ljallangan o‘lchov vositalarining sertifikati, o‘lchov vositasining ilmiy yoki sanoat izlanishlarida foydalanish uchun qobiliyatini tasdiqlaydi. Bu sertifikat:

- Ilmiy tadqiqotlar: Tadqiqot va rivojlantirish loyihalarida ishonchli o‘lchov natijalarini ta'minlaydi.

- O‘lchov aniqligi: O‘lchov vositalarining yuqori aniqligini kafolatlaydi.

7. Kalibrlash Intervallarini Tasdiqlash Sertifikati

Bu sertifikat o‘lchov vositasining kalibrlash oralig‘ini tasdiqlaydi va qachon qayta kalibrlash kerakligini aniqlaydi. Bu sertifikat:

- Texnik xizmat ko‘rsatish rejasini belgilash: O‘lchov vositasini qachon va qanday kalibrlash kerakligini aniqlashga yordam beradi.

- Aniqlik va ishonchlilikni saqlash: O‘lchov vositalarining uzluksiz aniqligini ta'minlaydi.

Metrologiya sertifikatlari o‘lchov vositalarining ishonchliligi va aniqligini ta'minlashda muhim rol o‘ynaydi, bu esa har xil sanoat va ilmiy sohalarda sifatli va ishonchli natijalarni kafolatlaydi [2].

Masalaning kelib chiqish

# **II BOB. MOBIL ILOVANI LOYIHALASH UCHUN JAVASCRIPTGA MO‘LJALLANGAN FREYMVORKLAR**

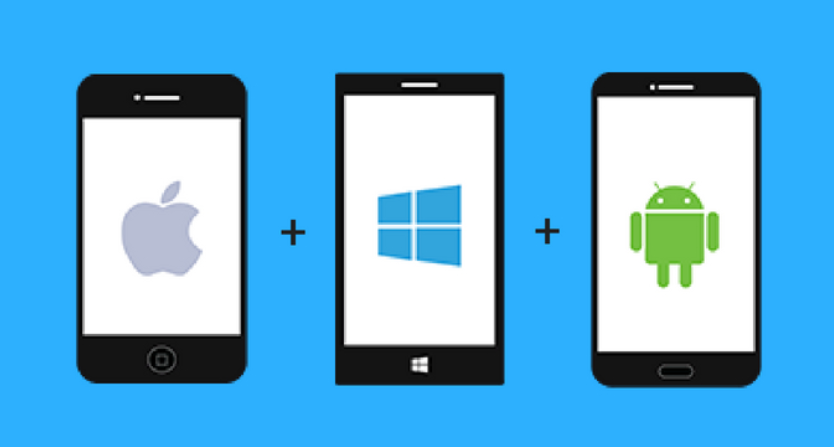
## **2.1. Mobil platformalar va ularning turlari**

*Mobil platformalar* – bu smartfon va planshetlar kabi mobil qurilmalar uchun ishlab chiqarilgan operatsion tizimlar va dasturiy muhitlardir. Mobile platformalarga quyidagilar kiradi:

1. Android – Google tomonidan ishlab chiqilgan Android Linux yadrosiga asoslangan ochiq kodli platformadir. U katta bozor ulushiga ega va ko‘plab ishlab chiqaruvchilar tomonidan qurilmalarini quvvatlantirish uchun foydalaniladi. Android yuqori darajadagi sozlashni taklif qiladi va Google Play do‘koni orqali keng ko‘lamli ilovalarni qo‘llab-quvvatlaydi [5].
2. iOS – Apple tomonidan ishlab chiqilgan iOS — iPhone va iPad kabi Apple mobil qurilmalari uchun maxsus xususiy operatsion tizim. O‘zining intuitiv foydalanuvchi interfeysi va kuchli xavfsizlik xususiyatlari bilan tanilgan iOS ilovalarning keng tanlovini taklif qiluvchi maxsus ilovalar do‘koniga ega [5].
3. Windows – Mobil qurilmalar uchun Windows, asosan, Microsoft kompaniyasining o‘z mahsulotlarida qo‘llaniladi, ish stoli Windows kabi yadroda ishlaydi, lekin sensorli ekranlar uchun optimallashtirilgan alohida foydalanuvchi interfeysiga ega edi. Biroq, Microsoft so‘nggi yillarda o‘z e'tiborini ushbu platformadan uzoqlashtirdi [5].
4. Boshqa platformalar. BlackBerry OS, Symbian va Firefox OS kabi boshqa mobil platformalar mashhurligi pasaygan yoki faol ravishda qo‘llab-quvvatlanmaydi.

Mobil platformalarning asosiylaridan bo‘lgan iOS, Windows va Android platformalar 2.1-rasmda ko‘ratilgan.

Ushbu platformalar ishlab chiquvchilar o‘z ilovalarini yaratish, sinab ko‘rish va nashr etish uchun foydalanishi mumkin bo‘lgan vositalar va xizmatlar to‘plamini taqdim etadi. Shuningdek, ular ishlab chiquvchilarga o‘z foydalanuvchilari uchun qiziqarli va interaktiv tajriba yaratishda yordam beradigan bir qator xususiyatlar va imkoniyatlarni taklif qiladi.



*2.1-rasm. Mobil platformalar turlari.*

Mobil platformalarga dasturiy ta’minot ishlab chiqish ikki turga bo‘linadi. Bular cross-platform, native mobile va hybrid developmentdir (2.2-rasm).

Native mobile development – bu iOS yoki Android kabi yagona platforma uchun maxsus ishlab chiqilgan mobil ilovalarni yaratishni o‘z ichiga oladi. Ushbu ilovalar iOS uchun Swift yoki Objective-C va Android uchun esa Java yoki Kotlin kabi platforma tomonidan qo‘llab-quvvatlanadigan dasturlash tillari va frameworklarda yoziladi. Native mobile developmentning afzalliklari shundaki, ilovalar odatda yaxshi ishlashi va foydalanuvchi tajribasini taklif qiladi, chunki ular muayyan platforma uchun optimallashtirilgan. Shuningdek, ular qurilmaning barcha xususiyatlari va imkoniyatlaridan to‘liq foydalanishlari mumkin. Kamchilliklariga kelsak bir nechta platformalar uchun ilovalarni ishlab chiqish alohida kod bazalarini talab qiladi, bu ko‘p vaqt talab qiladi va qimmatga tushishi mumkin. Bu, shuningdek, odatda, maxsus platformaning dasturlash tillari va asboblar to‘plami bo‘yicha tajribaga ega ishlab chiquvchilarni talab qiladi [6].

Cross-platform – bu iOS, Android va ba'zan hatto Windows kabi bir nechta platformalarda ishlay oladigan mobil ilovalarni yaratish imkonini beradi. Bunga ko‘pincha React Native, Flutter, Xamarin yoki Unity kabi platformalararo ishlab chiqish fremworklari va vositalari yordamida erishiladi. Cross-platformaning afzalliklar shundaki dasturni ishlab chiqish tejamkorroq va samaraliroq bo‘lishi mumkin, chunki u ishlab chiquvchilarga bir marta kod yozish va uni bir nechta platformalarda joylashtirish imkonini beradi. Shuningdek, u ilovaga texnik xizmat ko‘rsatish va yangilanishlarni soddalashtirishi mumkin. Kamchilliklariga kelsak,

ilovalari har bir platformaning o‘ziga xos xususiyatlaridan to‘liq foydalana olmasligi mumkin, bu esa foydalanuvchi tajribasidagi farqlarga olib kelishi mumkin. Ishlash, ayniqsa, yuqori darajadagi grafik yoki hisoblash intensivligini talab qiladigan ilovalar uchun ham tashvish tug‘dirishi mumkin [6].



*2.2-rasm. Dasturiy ta’minot ishlab chiqish turlari.*

Framework - bu dasturiy ta'minotni ishlab chiqish jarayonini soddalashtiradigan va tezlashtiradigan dasturiy ta'minot yoki kod strukturasidir. U tayyor kodlar, kutubxonalar, asboblar va usullar to‘plamini o‘z ichiga oladi, bu esa dasturchilarga ma'lum bir platforma yoki dastur turini yaratishda yordam beradi. Framework'lar dasturchilarga umumiy muammolarni hal qilish va takrorlanadigan vazifalarni bajarish uchun standart usullarni taklif etadi. U loyiha talablariga javob berish uchun ishlatilishi va hatto o‘zgartirilishi mumkin bo‘lgan shablon kabi ishlaydi [7].

## **2.2. JavaScriptga mo‘ljallangan freymvorklar**

JavaScript - koʻp paradigmali dasturlash tili bo‘lib, ob'ektga yo‘naltirilgan, imperativ va funktsional uslublarni qo‘llab-quvvatlaydi. Bu ECMAScript spetsifikatsiyasining standartilarini amalga oshiradi.

JavaScript odatda dastur ob'ektlariga dasturiy kirish uchun o‘rnatilgan til sifatida ishlatilgan va u dastlab veb-sahifalarga interaktivlikni qo‘shish uchun skript tili sifatida brauzerlarda keng qo‘llanilgan [8].

Asosiy arxitektura xususiyatlari: tiplarni dinamik aniqlash, avtomatik xotirani boshqarish, prototip dasturlash, birinchi darajali ob'ektlar sifatida funktsiyalar.

Frameworklar malum bir dasturlash tillariga asoslangandir. React, Flutter, Angular, Vue, PyTorch, Spring Boot, React Native, Apache Spark, Ionic va boshqalar eng ko‘p qo‘llaniladigan mashhur framworklardir. Bu framworklar ishlab chiquvchilarga mustahkam va boy funksiyalarga ega dasturiy ta'minotni yaratishga imkon beradi.

Agar frameworkdan fondalanilmasa dastur kodni toza, yaxshi sinovdan o‘tgan va xatolarsiz yozish kerak kerak. Boshqa ishlab chiquvchilar uchun kodni tushunish va u ustida ishlash qiyin bo‘ladi. Shunday qilib, sizning talablaringizga javob beradigan framworklar bilan ishlash yaxshiroqdir. Ular ishlab chiqish jarayonini kamroq xatolar bilan osonlashtiradi. Bu umumiy shablon bo‘lib, undan foydalanish va talabga muvofiq o‘zgartirish mumkin. Boshqalar sizning kodingizni tushunishlari oson bo‘ladi, chunki ular ham framworklar bilan yaxshi tanish bo‘lishadi.

Frameworklardan foydanalish afzalliklari quyidagilardir:

1. Kodni sinab ko‘rilgan va xatoliklar tuzatilgan bo‘ladi.
2. Toza kodni tushunish va ishlash juda oson.
3. Loyihada kodning ortiqchaligini kamaytiradi.
4. Kengaytirilgan dastur bilan loyihaning vaqtini va narxini pasaytiradi
5. Framework taqdim etilgan xususiyatlar va funksiyalar o‘zgartirilishi va kengaytirilishi mumkin.

Ishlab chiqish jarayonini boshlashdan oldin, qaysi dasturlash tilini loyihangizga mos kelishini tanlashingiz kerak. Misol uchun, agar JavaScript-ni tanlansa, veb dastur uchun Angular, React, Next.js, Vue.js kabi JavaScript frameworklardan birini tanlab ishlatish mumkin (2.3-rasm).

Javascript dasturlash tilga mo‘ljallangan mobil dastur yaratish uchun frameworklar:

1. React Native - bu JavaScript va React yordamida mobil ilovalarni yaratish uchun ochiq manbali framework. U Facebook tomonidan ishlab chiqilgan va ishlab chiquvchilar tomonidan mobil ilovalarni ishlab chiqishda eng ko‘p foydalaniladiganlardan frameworklardan biri hisoblanadi. Bu ishlab chiquvchilarga bir xil kod bazasiga ega iOS va Android-da ishlatilishi mumkin bo‘lgan mobil ilovalarni yaratish imkonini beradi. React native keng qo‘llanilgani uchun u ishlab chiquvchilarga xatoliklar va muammolarini hal qilishda yordam beradigan katta hamjamiyat yordamiga ega. React native boshqa texnologiyalarni mobil ilovaga integratsiyalash uchun ko‘plab plaginlarni taqdim etadi [9].

2. PhoneGap - bu Adobe tomonidan ishlab chiqilgan ochiq manbali framework. U Apache Software Foundation tomonidan qo‘llab-quvvatlanadi. U HTML, CSS va JavaScript kabi asosiy veb-dastur yaratish uchun muo‘ljallangan tillaridan foydalangan holda platformalararo mobil ilovalarni yaratish uchun ishlatiladi. Shuningdek, u platformalararo ishlab chiqishni qo‘llab-quvvatlaydi, ya'ni ilova bir xil kod bazasi bilan Android, iOS, Blackberry va boshqalarda ishlatilishi mumkin. Kod bazasi qayta ishlatilishi mumkinligi sababli, bu uni tejamkor qiladi. PhoneGap-da resurslarni baham ko‘radigan va bir-birini qo‘llab-quvvatlaydigan dasturchilar jamoasi mavjud. PhoneGap React Native dasturidan tezroq, chunki u asosiy veb-ishlab chiqish texnologiyalaridan foydalanadi [8].

3. NativeScript - bu Progress tomonidan ishlab chiqilgan ochiq manbali JavaScript frameworki. U NativeScript hamjamiyati tomonidan boshqariladi. U JavaScript, TypeScript va AngularJs yordamida mobil ilovalar yaratish uchun ishlatiladi. NativeScript, shuningdek, o‘zaro faoliyat platformalarni ishlab chiqishni qo‘llab-quvvatlaydi, bu esa ishlab chiquvchilar tomonidan foydalanish uchun yanada mos keladi va vaqt va kuchni tejaydi. U mobil ilovaga boshqa texnologiyalarni integratsiyalash uchun turli xil plaginlarni qo‘llab-quvvatlaydi. NativeScript resurslarni va qo‘llab-quvvatlashni ta'minlash uchun hamjamiyat yordamiga ega [8].

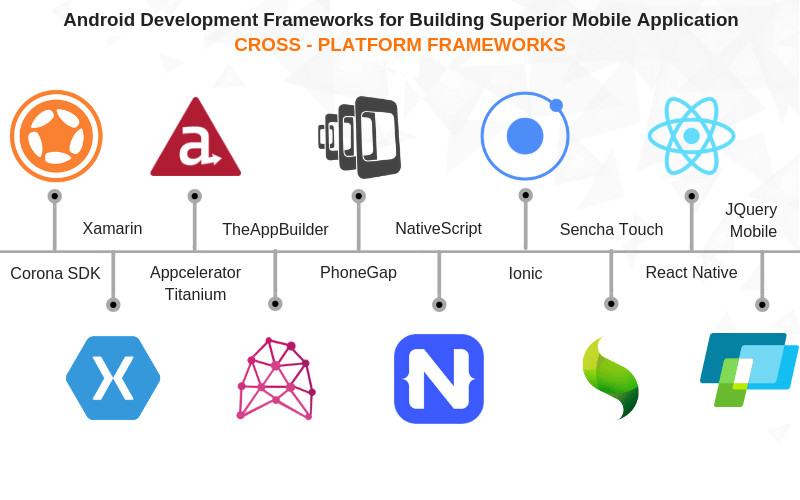
4. Ionic - bu MIT litsenziyasiga ega Drifty.co tomonidan ishlab chiqilgan va ion hamjamiyati tomonidan qo‘llab-quvvatlanadigan ochiq manbali JavaScript frameworki. Bu shuningdek, HTML, CSS va JavaScript kabi veb-ishlab chiqish texnologiyalaridan foydalangan holda mobil ilovalarni ishlab chiqishning eng oddiy frameworklaridan biridir. Ionic veb-texnologiyalar yordamida jozibali UI taklif qiladi. Ionic sezgir va foydalanuvchilarga qulay mobil ilovalarni ishlab chiqadi. Bu ishlab chiquvchilarga o‘z ilovalarida mahalliy funksionallik va xususiyatlardan foydalanish imkonini beruvchi mahalliy API-larga kirishni ta'minlaydi. Ionic - platformalararo mobil ilovalarni ishlab chiqish uchun kuchli frameworkdir [8].

5. Framework7 - HTML, CSS va JavaScript kabi veb-texnologiyalardan foydalangan holda mobil ilovalar, veb-ilovalar va ish stoli ilovalarini yaratish uchun mutlaqo bepul va ochiq manbali framework. MIT tomonidan litsenziyalangan. U Vladimir Xarlampidi tomonidan ishlab chiqilgan va hozirda Framework7 hamjamiyati tomonidan qo‘llab-quvvatlanadi. Bu ishlab chiquvchilarga android, iOS, windows va boshqalar kabi turli platformalarda ishlaydigan ilovalar yaratish imkonini beradi.U HTML, CSS va JS-dan foydalansa-da, bular bilan cheklanmagan, ishlab chiquvchilar ishlab chiqish uchun React.js va Vue.js-dan foydalanishlari mumkin. Unda foydalanishga tayyor bo‘lgan ro‘yxatlar, shakl elementlari va boshqalar kabi turli komponentlar mavjud bo‘lib, ular ishlab chiquvchilar uchun qulaylik yaratadi. U turli xil API-lar bilan osongina birlashtirilishi mumkin [8].

6. Onsen - UI HTML, CSS va JavaScript kabi veb-texnologiyalar yordamida mobil va veb-ilovalarni yaratish uchun ochiq manbali UI asosidir. Bu mobil va ish stoli veb-ilovalari uchun jozibali foydalanuvchi tajribasini taqdim etadi. U shakl komponentlari, ro‘yxatlar, modallar, dialoglar va boshqalar kabi bir qator o‘rnatilgan komponentlar bilan birga keladi, bu esa ishlab chiquvchilarga ilovalarni ishlab chiqishni osonlashtiradi. U barcha ekran o‘lchamlariga mos keladigan sezgir dizaynni taqdim etadi. U qayta ishlatilishi mumkin bo‘lgan va tejamkor bo‘lgan platformalararo rivojlanishni qo‘llab-quvvatlaydi. U uchinchi tomon ilovalari va API bilan osongina birlashtirilishi mumkin. U yangi ishlab chiquvchilarni qo‘llab-quvvatlaydigan keng jamoatchilikka ega [7].

7. Meteor - bu mobil va kompyuter uchun dasturlarini ishlab chiqish uchun ishlatiladigan JavaScript frameworkdir. Meteor birinchi marta veb-ilovalarni yaratish uchun mo‘ljallangan, u mobil ilovalarni ishlab chiqish uchun ham ishlatilishi mumkin. U veb va mobil ilovalar uchun yagona kod bazasidan foydalanadi, bu esa ishlab chiquvchilarga ilovalarni yaratish va ularga xizmat ko‘rsatishni osonlashtiradi.

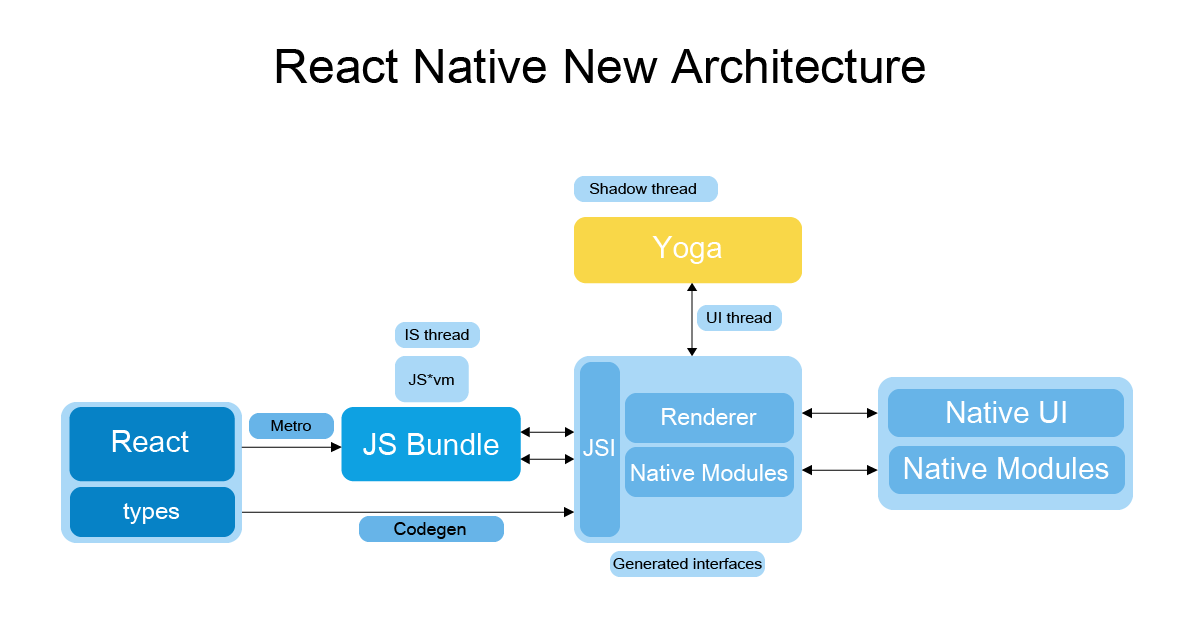
Mobil ilovalarni ishlab chiqish uchun Meteor-dan foydalanishning afzalliklaridan biri shundaki, u ishlab chiquvchilarga iOS va Android platformalari uchun bir xil kod bazasidan foydalanish imkonini beradi. Bu vaqt va mablag‘ni sezilarli darajada tejash imkonini beradi, chunki ishlab chiquvchilar har bir platforma uchun alohida kod yozishlari shart emas. U asosan veb-ilovalarni ishlab chiqish uchun mo‘ljallanganligi sababli, uni juda murakkab mobil ilovalarni ishlab chiqish uchun ishlatib bo‘lmaydi [4].



*2.3-rasm. Mobil dasturlash uchun frameworklar.*

## **2.3. React Native framevorki va Expo komponentalaridan foydalanish**

React Native Javascript dasturlash tiliga asoslangan framework bo‘lib, Facebook (hozirgi Meta) tomonidan ishlab chiqilgan ochiq kodli (open source) framework. Bu framework orqali Android, Android TV, iOS, macOS, tvOS uchun ilovalarni ishlab chiqish uchun ishlatiladi. React Native javascriptning mobil platformarlar uchun frameworklari ichida eng mashhuridir. Uning ishlash prinsipi 2.4-rasmda keltirilgan.

****

*2.4-rasm. React Native arvitekturasi*

React Native mobil dasturchiga ilovalarni yaratishga imkon beradigan barcha kerakli API-larga ega asboblar to‘plamini taqdim qiladi. Bu asboblar orqali qurilmalarning fizik qismlaridan foydalanish munkin.

React Native orqali qo‘shimcha frameworklarsiz ham mobil ilova yaratish mumkin. Lekin ko‘pchillik mobil ilova ishlab chiquvchilari Expo frameworkidan foydalanishi aniqlangan. Expo faylga asoslangan marshrutlash, yuqori sifatli universal kutubxonalar va mahalliy fayllarni boshqarmasdan mahalliy kodni o‘zgartiradigan plaginlarni yozish qobiliyati kabi xususiyatlarni taqdim etadi. Frameworklardan foydalanishning yana bir ustun tarafi shundaki, agar frameworklardan foydalanilmasa har bir komponentani, masalan button (tugma) komponentasini o‘zingiz yaratib chiqishingiz kerak bo‘ladi. Bu esa vaqt va boshqa resurslarni ko‘proq sarflashga olib keladi. Frameworkda esa bunda komponentalar avvaldan yaratib quyilgan va har bir kod sinov test tekshiruvlaridan o‘tgan bo‘ladi. Bunda oldindan yaratilgan komponentada xatoliklar kamroq uchraydi.

React Native hamjamiyati ko‘p yillar davomida navigatsiya, mahalliy API-larga kirish, mahalliy bog‘liqliklar bilan ishlash va boshqalarga yondashuvlarni takomillashtiradi. Aksariyat ilovalar ushbu asosiy xususiyatlarga muhtoj. React Native framework ularni ilovangiz boshidanoq beradi.

Agar ilovada Framework tomonidan yaxshi xizmat ko‘rsatmaydigan noodatiy cheklovlar bo‘lsa yoki ushbu muammolarni hal qilishni ma’qul ko‘rilsa, Android Studio, Xcode’dan foydalanib, React Native ilovasini Frameworksiz yaratish mumkin. Bunda React Native jamoasi muhitingizni qanday sozlashni va qanday qilib frameworksiz dastur yaratish bo‘yicha qo‘llanma yaratgan [5].

Expo - bu React Native yordamida mobil ilovalar yaratish uchun ishlab chiqilgan ochiq manba platformasi. U ishlab chiqish jarayonini soddalashtiradigan va ishlab chiquvchilarga iOS, Android va veb-platformalar uchun yuqori sifatli ilovalar yaratishda yordam beruvchi vositalar va xizmatlar to‘plamini taqdim etadi. Expo quyidagi asosiy xususiyatlarga ega:

1. Ishlab chiqish muhiti: Expo daturchilarga o‘zlarining React Native ilovalarini osongina yaratish, oldindan ko‘rish va sinab ko‘rish imkonini beruvchi ishlab chiqish muhitini taqdim etadi. U loyiha aktivlarini boshqarish, disk raskadrovka kodini va jismoniy qurilmalar yoki simulyatorlarda real vaqtda ilovani oldindan ko‘rish vositalarini o‘z ichiga oladi.

2. Birlashtirilgan API: Expo platformaga xos tafovutlar olib tashlaydigan yagona API taqdim etadi, bu esa ishlab chiquvchilarga bir nechta platformalarda ishlaydigan kod yozishni osonlashtiradi. Bu dasturchilarga platformaga xos tafsilotlar haqida qayg‘urmasdan, ilovaning xususiyatlarini yaratishga e’tibor qaratish imkonini beradi.

3. O‘rnatilgan komponentlar va kutubxonalar: Expo oldindan o‘rnatilgan UI komponentlari, kutubxonalari va modullarini taqdim etadi, ulardan ishlab chiquvchilar o‘z ilovalariga kameraga kirish, bildirishnomalar yuborish va joylashuv kabi umumiy xususiyatlarni qo‘shish uchun foydalanishi mumkin. Bu rivojlanishni tezlashtirishga yordam beradi.

4. Yangilanishlar: Expo dasturchilarga ilovalar do‘konidan yangi versiyani yuklab olishni talab qilmasdan, dasturchilarga havola orqali o‘z ilovalariga yangilanishlarni yuborish imkonini beradi. Bu ilovalar do‘konini ko‘rib chiqish jarayonidan o‘tmasdan xatolarni tezda tuzatish yoki yangi xususiyatlarni o‘rnatish uchun foydali bo‘ladi.

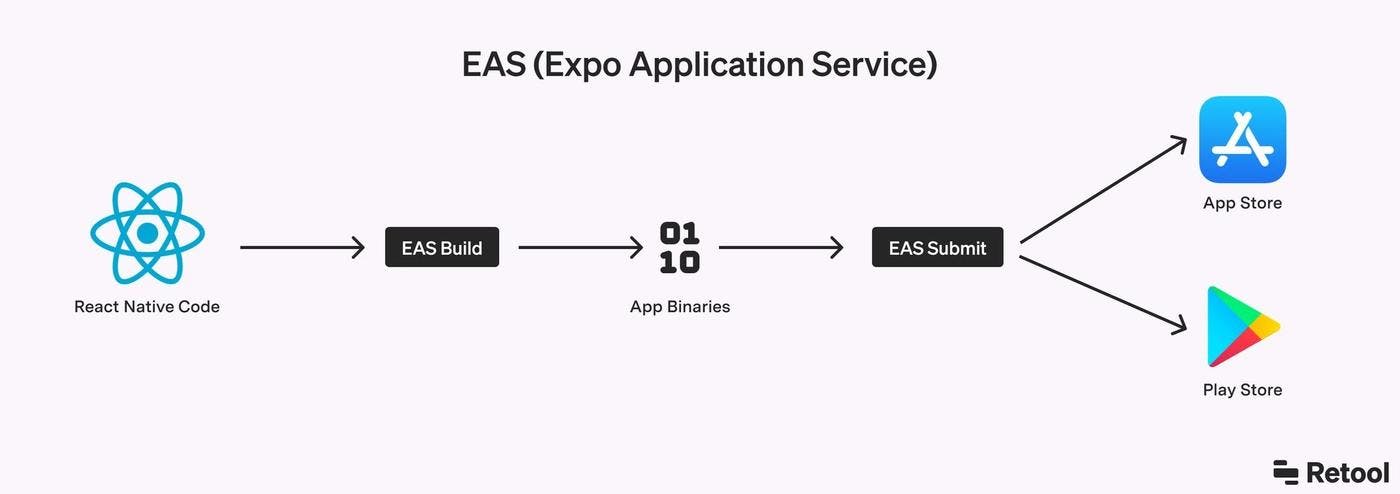
5. Nashr qilish va tarqatish: Expo ilova do‘konlariga ilovalarni tayyorlash va nashr etish, shuningdek, sinov uchun beta-versiyalarni tarqatishni boshqarish vositalarini taqdim etadi. Bu joylashtirish jarayonini soddalashtiradi va ilovalarning foydalanuvchilarga samarali yetkazib berilishini ta'minlaydi.

Expo React Native bilan platformalararo mobil ilovalarni yaratish uchun ishlab chiqish jarayonini soddalashtiradigan kuchli platformadir. U React Native frameworkining afzalliklaridan foydalangan holda ilovalarni tez va samarali yaratmoqchi bo‘lganlar tomonidan keng qo‘llaniladi.

React Native va Expodan foydalanilgan holdan ilova yaratish uchun quyidagi ko‘nikmalarga ega bo‘lish kerak:

* HTML va CSS
* Javascript
* ReactJS
* NodeJS va npm paketlaridan foydalanish.
* EAS (Expo Application Services)

EAS ning ishlash sxemasi 2.5-rasmda keltirilgan.



*2.5-rasm. EASning ishlash prinsipi.*

Ilova yaratish uchun quyidagi qadamlarni bajarish kerak:

1. Nodejs paket menejerini tizimga o‘rnatib olish kerak. Bunda <https://nodejs.org>/ sayti orqali yuklab olish va tizimga o‘rnatish kerak.
2. Loyhaning yaratish uchun quyidagi kodni terminalda ishga tushirish kerak:

* npx create-expo-app@latest

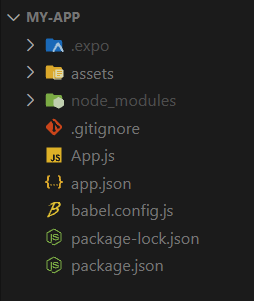
1. Keyingi qadamda ilova nomi va kerakli paketlarni belgilash kerak.
2. Ilova paketlarni yuklangandan so‘ng, terminal orqali ilova joylashga papkaga o‘tish kerak.

* cd app\_name

1. Ilovani ishga tushirish uchun quyidagi kod ishlatiladi:

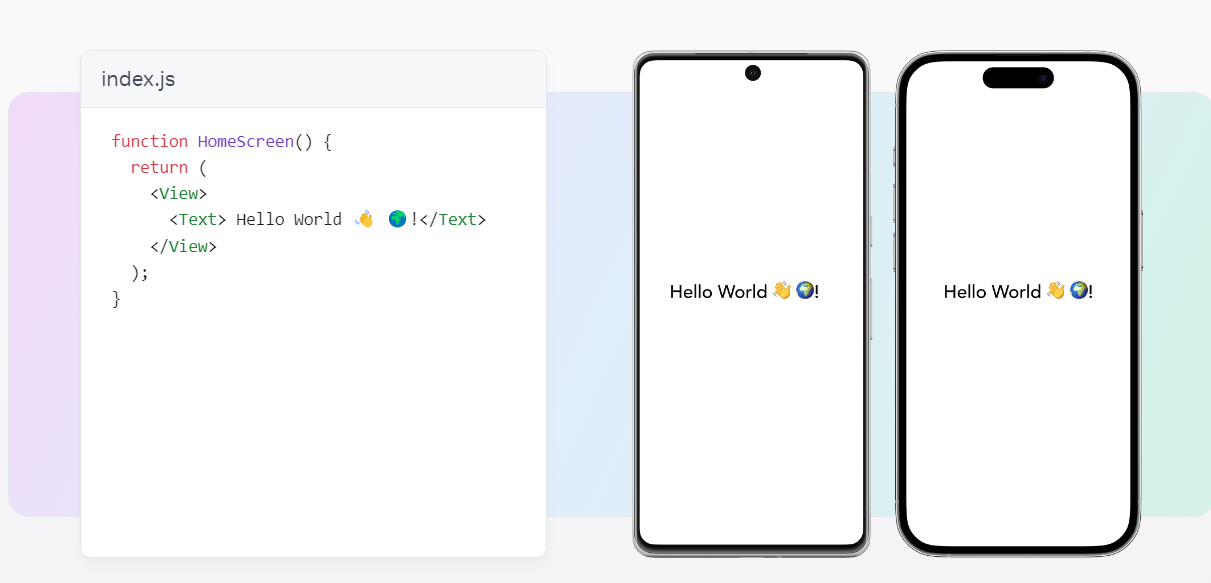
* npx expo start

1. Loyihaning fayl strukturasi quyidagicha bo‘ladi (2.6-rasm).

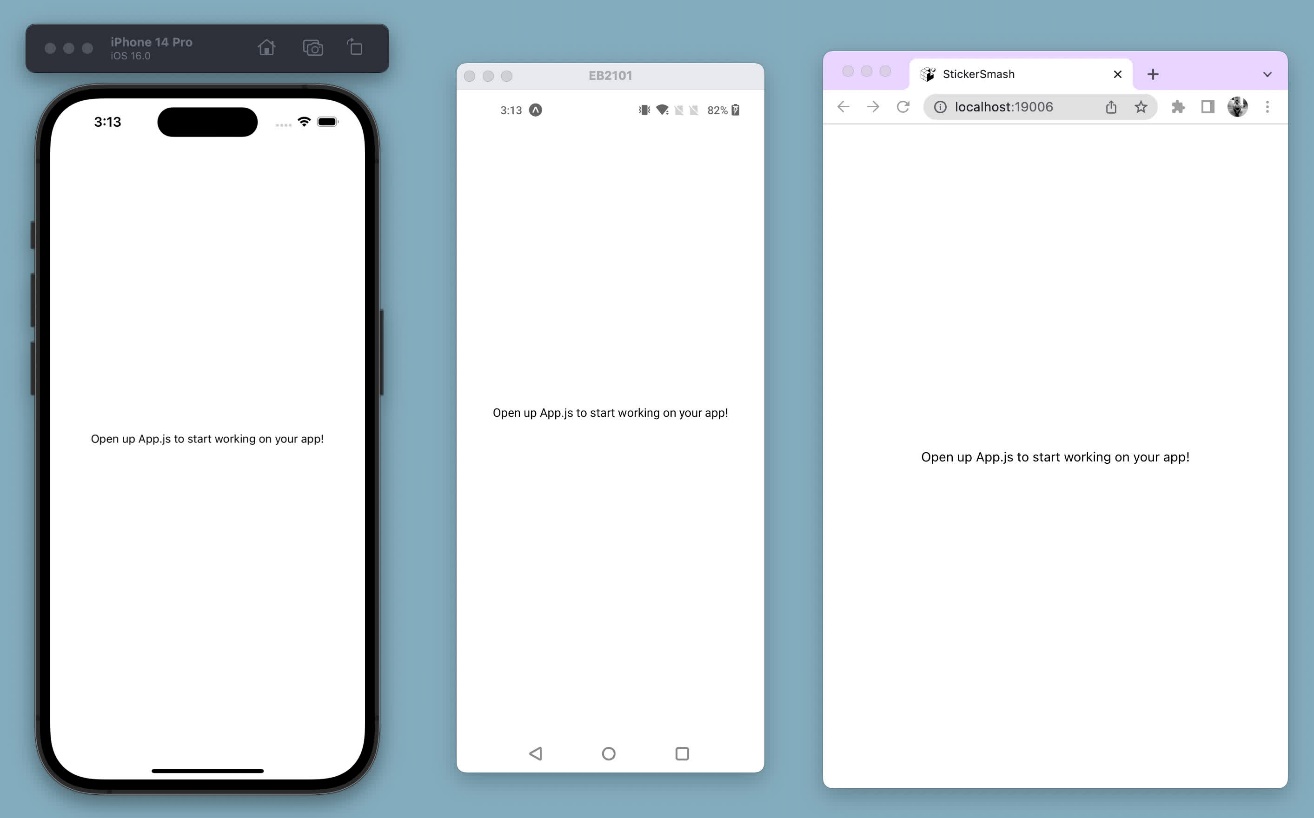


*2.6-rasm. React Native fayl strukturasi*

1. Bunda asosiy va boshlang‘ich yuklanuvchi kod App.js fayliga yozilgan bo‘ladi. Uni o‘zgatirish orqali dasturni o‘zgatirish munkin. (2.7-2.8- rasmlar)



*2.7-rasm. Dastur bosh fayli*



*2.8-rasm. Daturni turli platformalarda ishlashi.*

Ilovani qurish jarayonida oldindan ko‘rish uchun ishlatishda Android uchun Android Studioning emulyatori yoki dastur ishlab turgan tizim bilan bitta tarmoqda turgan istalgan android smartfon bilan ishlatish munkin. Bunda yaratilayotgan dastur qurilmadagi Expo Go dasturini ichida ishga tushadi va buning uchun qurilmaga Expo Go dasturini yuklab olish va o‘rnatish kerak. Agar ilovani iOS platformasi uchun ishlab chiqish maqsad qilingan bo‘lsa, tizimga iOS simulyator yoki ios qurilma orqali ishlatish munkin.

Ilovani Expo Go dasturida ishga tushirish uchun dastur ishlatilganda berilgan maxsus QR kodni scanner qilishishi kerak. Agar datur yaratuvchi tizim va ishga tushiruvchi qurilma yagona tarmoqda bo‘lsada dastur yuklanishda qatolik chiqadi. Dasturni qurish davomida tizimlar tarmoqdan chiqib ketmasligi lozim. Aks holda dasturni qaytadan ishlatish kerak bo‘ladi.

Dasturni qayta yuklash uchun dastur ishlatuvchi termilanga ctrl + C klavshlari kombinatsiyasi yordamida tugatilib qaytadan ishga tushirilishi kerak.

|  |  |
| --- | --- |
|  |  |

*2.9-rasm. Ilovani ishga tushirish*

React Native frameworki va Expo platfomasinig asosiy komponentalari:

1. View - UI yaratish uchun eng asosiy komponent bo‘lib, boshqaruvlari bilan tartibni qo‘llab-quvvatlaydigan konteynerdir.
2. Text – React Native dasturda matnni ko‘rsatish uchun komponentdir. Matn joylashtirish uslublari va barcha matin xususiyatlarini bilan ishlashni qo‘llab-quvvatlaydi.
3. Button - Har qanday platformada yaxshi ko‘rsatilishi kerak bo‘lgan asosiy tugma komponenti. Moslashtirishning minimal darajasini qo‘llab-quvvatlaydi.
4. ScrollViews - bir nechata komponentani joylashtirganda ekranga sig‘may qolidanga ishlatilivchi komponenta.
5. Image - Har xil turdagi tasvirlarni, jumladan, tarmoqdan, statik resurslardan, vaqtinchalik mahalliy tasvirlar va mahalliy diskdagi tasvirlarni ko‘rsatish uchun React Native komponenti.
6. ImageBackground – komponetaning orqa foniga tasvir quyish uchun ishlatilivchi kompanenta.
7. TextInput - Klaviatura orqali ilovaga matn kiritish uchun asosiy komponent. Avtomatik bosh harflar bilan yozish, belgilar va raqamli klaviatura kabi turli xil klaviatura turlari kabi bir nechta funksiyalar uchun konfiguratsiyani taʼminlaydi.
8. FlatList – ekranga ro‘yxat elementlarini chiqarish uchun qullaniluvchi kompanenta.
9. TouchableOpacity – tugma komponentasining boshqa uslubdagi varyanti bo‘lib, tugma sohasi bosilganda shaffoflik kamayadi va animatsiyani boshqarish xususiyatlatiga ega.
10. Modal – mobil ilovada modal ya’ni qalqib chiquvchi oyna vazifasini bajaruvchi kompanenta.
11. Switch – bu ba’zi qiymatlarni o‘chirib yoqish vazifasini bajarish uchun kompanenta bo‘lib, asosian sozlanmalar bo‘limida bajariladi.

# **III BOB. DASTURIY TA’MINOTNI LOYIHALASH VA ISHLAB CHIQISH**

## **3.1. Dasturiy ta’minotning mantiqiy loyihasi (jadvallar, relatsion model) API**

Dasturni loyhalashda bir necha bosqichlardan o‘tish kerak. Bu bosqichlar quyidagilardir:

1. Loyihani rejalashtirish: loyha maqsadi va vazifalarini aniqlashtirib olish.
2. Texnik topshiriqni tayyorlash: loyha dasturini tayyotlash uchun dasturlash tilini, platformalarni va ishlatiladigan texnalogiyalarni tanlash. Bunda tashqari texnik tuzulishini, xavfsizlik talablarini va boshqa xusisiyatlarini aniqlash.
3. Dasturni dizayn qismini tayyorlash va ma’lumotlar bazasini tuzilishini tayyorlash: dasturning texnik vazifalariga qarab, veb va mobil platformalar uchun dastur UI (user interface) ni tayyorlash va dasturning ma’lumotlar bazasini shakillantirish.
4. Dastur kodini yozish: tanlanga texnalogiyalardan foydalanib dasturning veb va mobil qismini tayyorlash. Mobil dasturni veb dasturga API (application programming interface) yordamida bog‘lash.
5. Dasturni testlash: tayyorlangan dasturni sinovdan o‘tkazish va xatoliklarni bartaraf qilish.
6. Dasturni tarqatish va doimiy qo‘llab-quvvatlash: tayyorlangan dasturni qurish va ilovalar do‘koniga joylash. Keyinchalik dasturni rivojlantirish uchun yangi qushimchalar kiritish va yangilab borish.

Qo‘yilgan masala yuzasidan ma’lumotlar bazasida jadvallar hosil qilindi. Ular orasidagi bog‘lanishlar mos keluvchi (ustma-ust tushuvchi) maydonlari yordamida bog‘langan. Ma’lumotlar bazasidagi jadvallar orasidagi munosabatlar quyidagi to‘rtta turda bo‘lishi mavjud: birga-bir, birga-ko‘p, ko‘pga-bir, ko‘pga-ko‘p. Masala yuzasidan ma’lumotlar bazasini loyilahashda birga-ko‘p va ko‘pga-ko‘p bog‘lanish munosabatlaridan foydalanildi. Birga-ko‘p bog‘lanish munosabati ma’lumotlar bazasini loyihalashda ko‘p uchraydi va uning vazifasi takrorlanmaydigan qiymat oluvchi (kalit) maydonga ega bo‘lgan jadvalning har bir yozuvi qiymatiga ushbu maydon qiymatlari orqali bog‘langan jadvaldagi ko‘p sondagi yozuvlarini mos qo‘yishdir.

Ma’lumotlar bazasidagi jadvallar va uning tarkibiy qismlari – maydonlari haqidagi ma’lumotlarni keltiramiz:

“action” – tizimdagi barcha harakatlarni saqlovchi jadval

|  |  |  |
| --- | --- | --- |
| **Maydon nomi** | **Tipi** | **Izoh** |
| idaction (Primary) | int(11) | Kalit maydon |
| module | varchar(40) | Harakat bajarilgan module nomi |
| description | varchar(80) | Harakat izohi |
| date\_time | datetime | Harakat bajarilgan vaqt |
| user\_iduser (Primary) | int(11) | Harakat bajaruvchisi |
| user\_role\_id (Primary) | int(11) | Bajaruvchi roli |

“certificate” – yaratilgan sertifikatlarning ma’lumotlarini saqlovchi jadval

|  |  |  |
| --- | --- | --- |
| **Maydon nomi** | **Tipi** | **Izoh** |
| id (Primary) | int(11) | Kalit maydon |
| orginal\_number | int(11) | Hujjatning raqami |
| type | int(11) | Sertifikat turi |
| lang | varchar(15) | Sertifikat tili |
| status | tinyint(4) | Sertifikat holati (tasdiqlanganligi) |
| date | date | Sanasi |
| time | time | Vaqti |
| link | varchar(120) | Sertifikat yuklab olish manzili |
| url | varchar(120) | Sertifikat asl manzili |
| data | blob | Ma’lumotlari |
| comply\_with | varchar(80) | Amal qilish muddati |
| compare | varchar(80) | Solishtirma |
| organization | varchar(80) | Tashkilot nomi |
| user\_iduser | int(11) | Yaratgan foydalanuvchi idsi |

“”user” – foydalanuvchilarni saqlovchi jadval

|  |  |  |
| --- | --- | --- |
| **Maydon nomi** | **Tipi** | **Izoh** |
| iduser (Primary) | int(11) | Kalit maydon |
| name | varchar(50) | Ismi |
| phone | varchar(35) | Telefon raqami |
| email | varchar(45) | Email manzili |
| password | varchar(250) | Paroli |
| role\_id (Primary) | int(11) | roli |

“role” – foydalanuvchilarning huquqlarini saqlovchi jadval

|  |  |  |
| --- | --- | --- |
| **Maydon nomi** | **Tipi** | **Izoh** |
| id (Primary) | int(11) | Kalit maydon |
| name | varchar(45) | Role nomi |

“task” – vazifalarni saqlovchi jadval

|  |  |  |
| --- | --- | --- |
| **Maydon nomi** | **Tipi** | **Izoh** |
| id (Primary) | int(11) | Kalit maydon |
| name | varchar(50) | Vazifa nomi |
| path | varchar(65) | Vazifa yo‘li |
| description | varchar(450) | Vazifa tasnifi |

“role\_has\_task” –vazidalar va role jadvalining bog‘lanishidan hosil bo‘lgan jadval

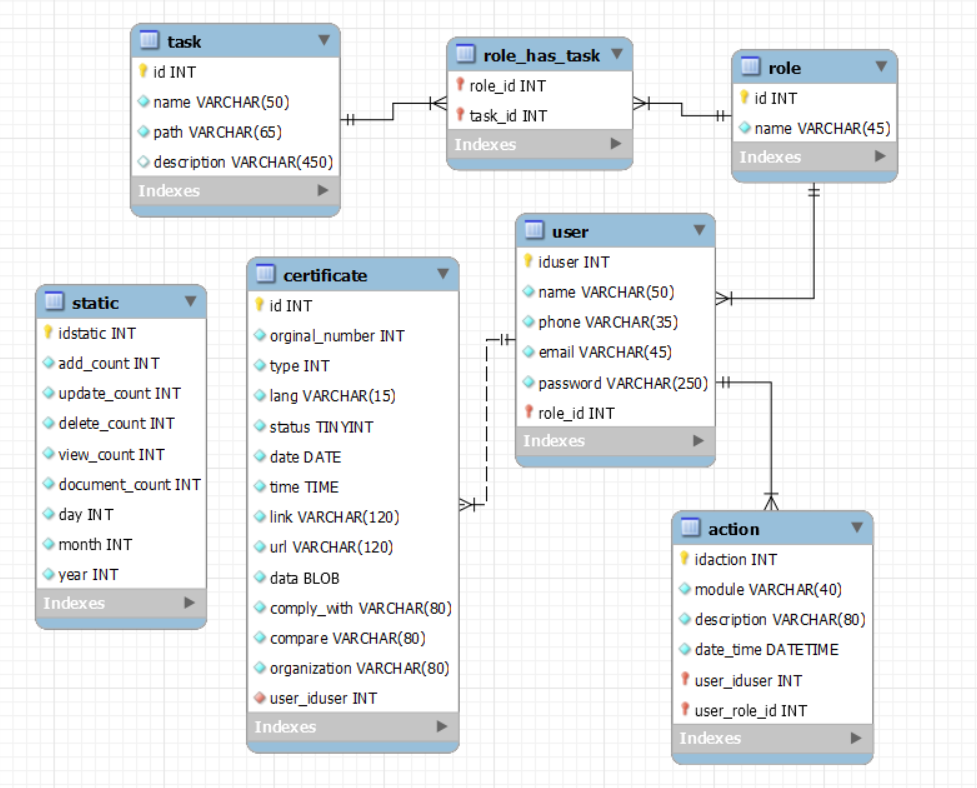
|  |  |  |
| --- | --- | --- |
| **Maydon nomi** | **Tipi** | **Izoh** |
| role\_id (Primary) | int(11) | Rol id qiymati |
| task\_id (Primary) | int(11) | Vazifa id qiymati |

“Static“ – tizimgagi ma’lumotlar statistikasini saqlovchi jadval

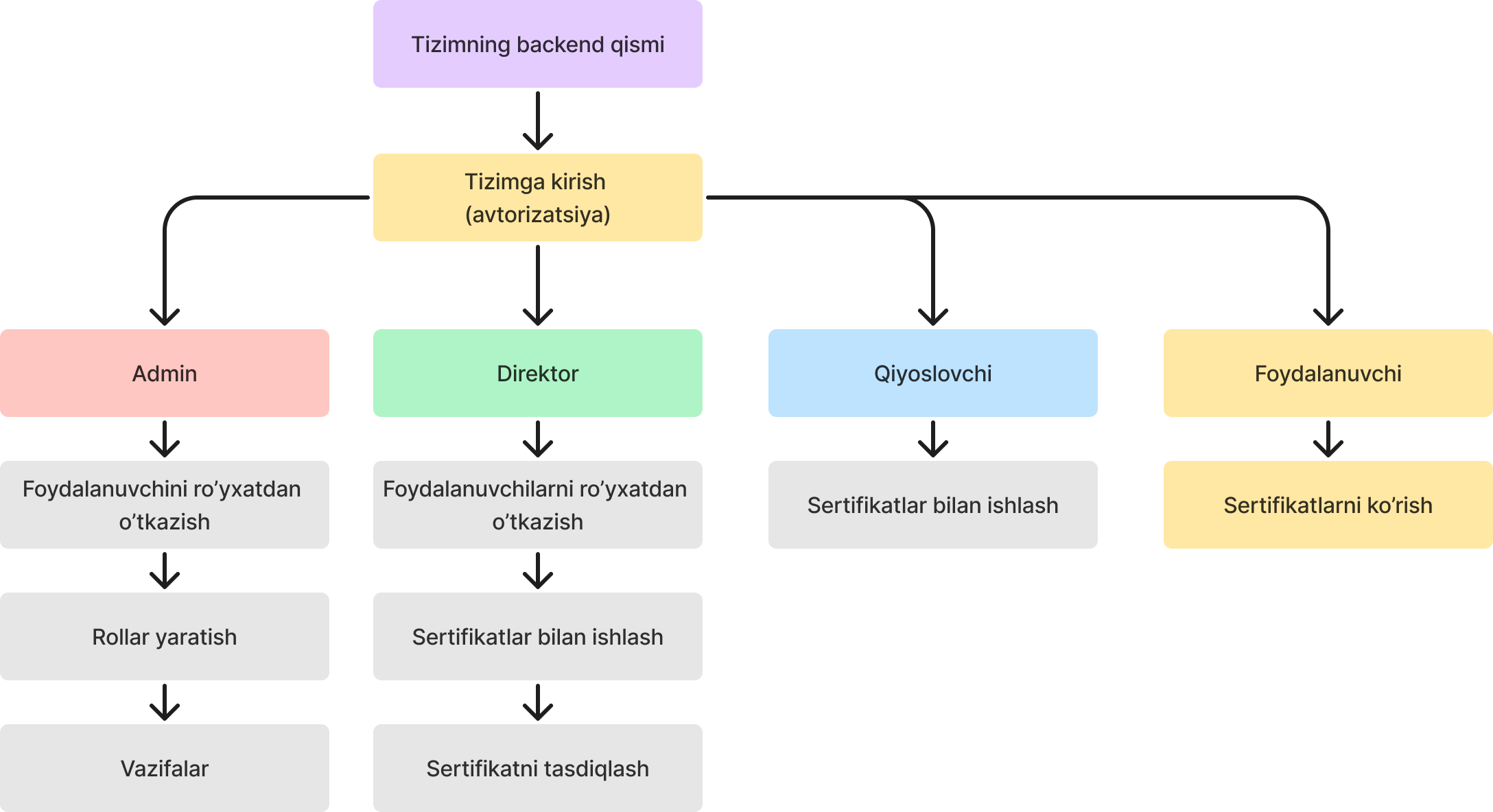
|  |  |  |
| --- | --- | --- |
| **Maydon nomi** | **Tipi** | **Izoh** |
| idstatic (Primary) | int(11) | Kalit maydon |
| add\_count | int(11) | Yaratilgan hujjatlar soni |
| update\_count | int(11) | Yangilangan hujjatlar soni |
| delete\_count | int(11) | O‘chirilgan hujjatlar soni |
| view\_count | int(11) | Ko‘rilgan hujjatlar soni |
| document\_count | int(11) | Barcha hujjatlar soni |
| day | int(11) | Yangilangan kun |
| month | int(11) | Yangilangan oy |
| year | int(11) | Yangilangan yil |

Dasturiy ta’minotning ma’lumotlar bazasi modelini aks ettirish uchun relatsion modelning umumiy koʻrinishi quyida ko‘rsatilgan (3.1-rasm).

Ilovaning mantiqiy strukturasi dasturiy ta’minotda mavjud boʻlgan boʻlimlar, qismlar va vazifalarni oʻzida aks ettiradi (3.2-rasm).



*3.1-rasm. Ma’lumotlar bazasining relatsion modeli.*



*3.2-rasm. Dasturiy ta’minotning mantiqiy strukturasi.*

Mobil ilovada ma’lumotlarni dasturning backend qismidan olish uchun dasturlar orasida bog‘lanishni va ma’lumot almashinishni taminlovchi bog‘lanish turi kerak bo‘ladi. Bunda bizga API yordamga keladi.

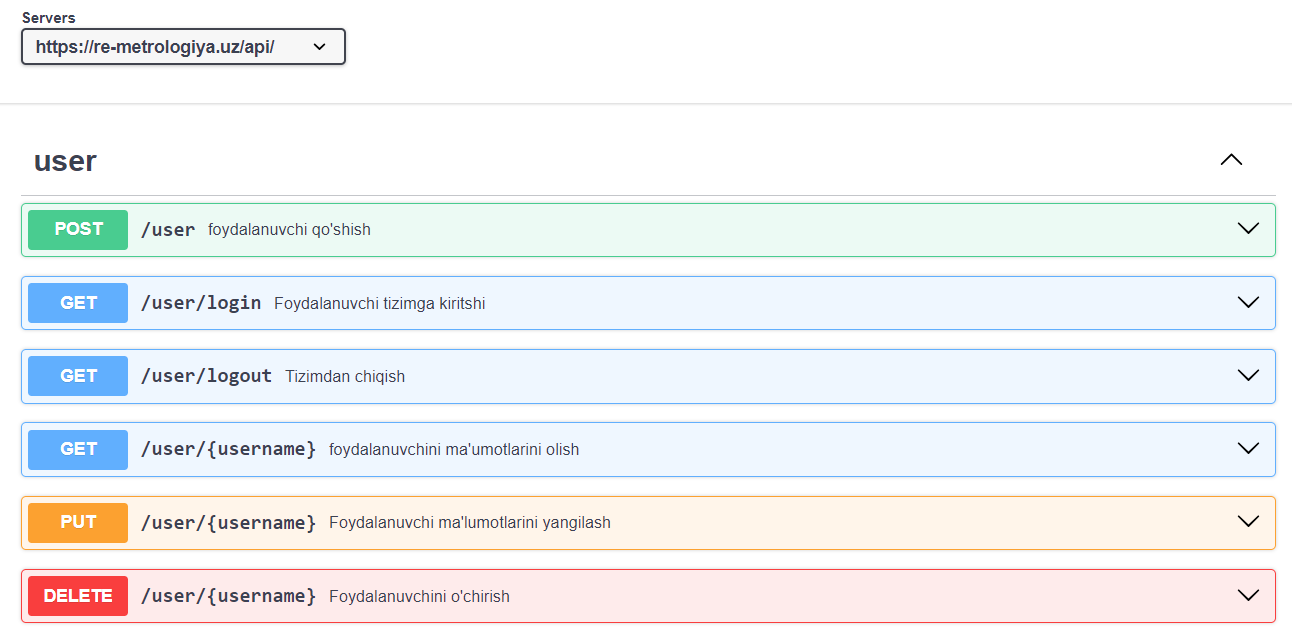
Application Programming Interface (API) — foydalanuvchi interfeysidan farqli ravishda, ikki yoki undan ortiq kompyuter dasturlari oʻzaro bogʻlanishi va muloqot qila olishi uchun ishlatilinadi. Bu turdagi aloqa yoki intefeysni ishlab chiqish, yoxud foydalanishni tavsiflovchi hujjat API spesifikatsiyasi (inglizcha: API specification) deb nomlanadi [9].

Kompyuter va insonni bogʻlovchi fodalanuvchi interfeysidan farqli ravishda, API kompyuterlar yoki dasturiy taʼminot qismlarini bir-biri bilan bogʻlashda ishlatiladi. API oddiy foydalanuvchilar tomonidan emas, balki, dasturchilar tomonidan foydalanish uchun moʻljallangan.

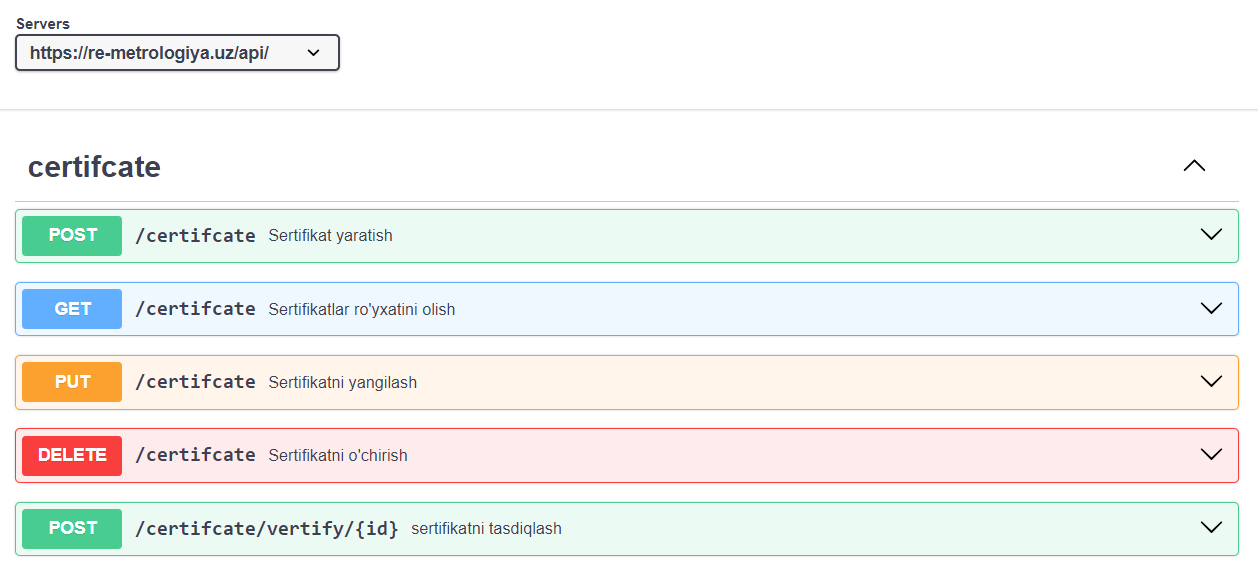
APIning asosiy maqsadlaridan biri tizimning ichki detallarini yashirib, dasturchi uchun kerak boʻladigan qismlarini ochiqlash va ichki detallar oʻzgargan taqdirda ham ularni izchil saqlashdan iboratdir. API maʼlum bir tizim juftligi uchun moslashtirilgan, yoki, koʻplab tizimlar oʻrtasida oʻzaro ishlash imkonini beruvchi umumiy standart tarzida ishlab chiqilgan boʻlishi mumkin.

Loyhaning backend dasturdan mobil dasturiga bog‘lash va ma’lumotlarni almashinish uchun quyidagicha APIlar tuziladi:

1. Tizimda ro‘yxatda o‘tish, kirish va foydalanuvchilar bilan ishlash oynalari uchun API larni qurish (3.3-rasm).
2. Sertifikatlar bilan ishlash uchun mo‘ljallangan oyna uchun backend dasturda quyidagcha API manzillari yaratish kerak (3.4-rasm):
3. Foydalanuvchilar uchun bajariladigan vazifalarni taqsimlash, olish, o‘zgatirish, yaratish va o‘chirish uchun API manzili.
4. Foydalanuvchilar rollari bilan ishlash uchun API manzillarini shakillantirish



*3.3-rasm. Foydalanuvchilar bilan ishlash bo‘limi.*

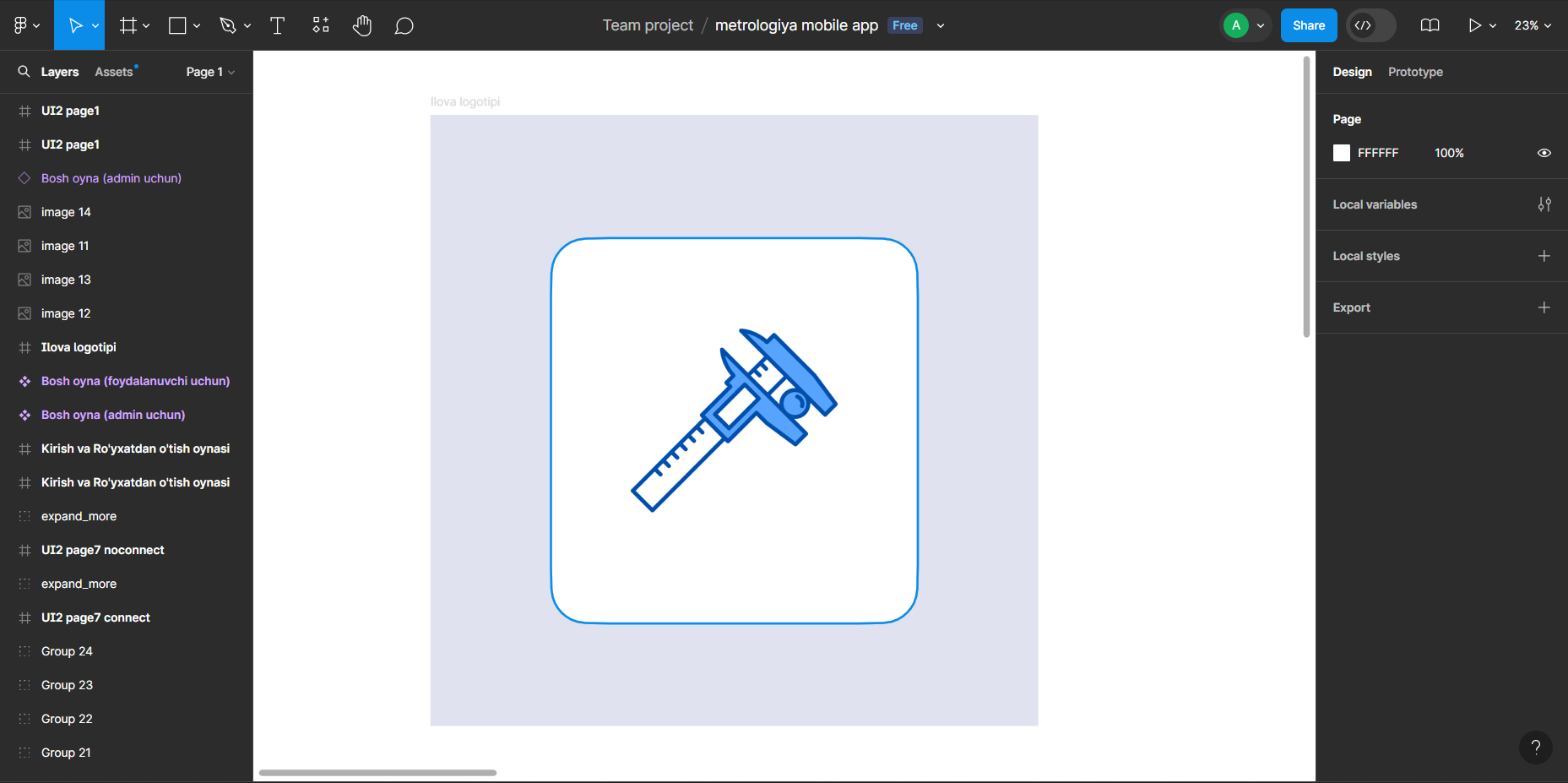


*3.5-rasm. Sertifikatlar bilan ishlash bo‘limi.*

## **3.2. Mobil ilovaning dizayn qismini loyhalash**

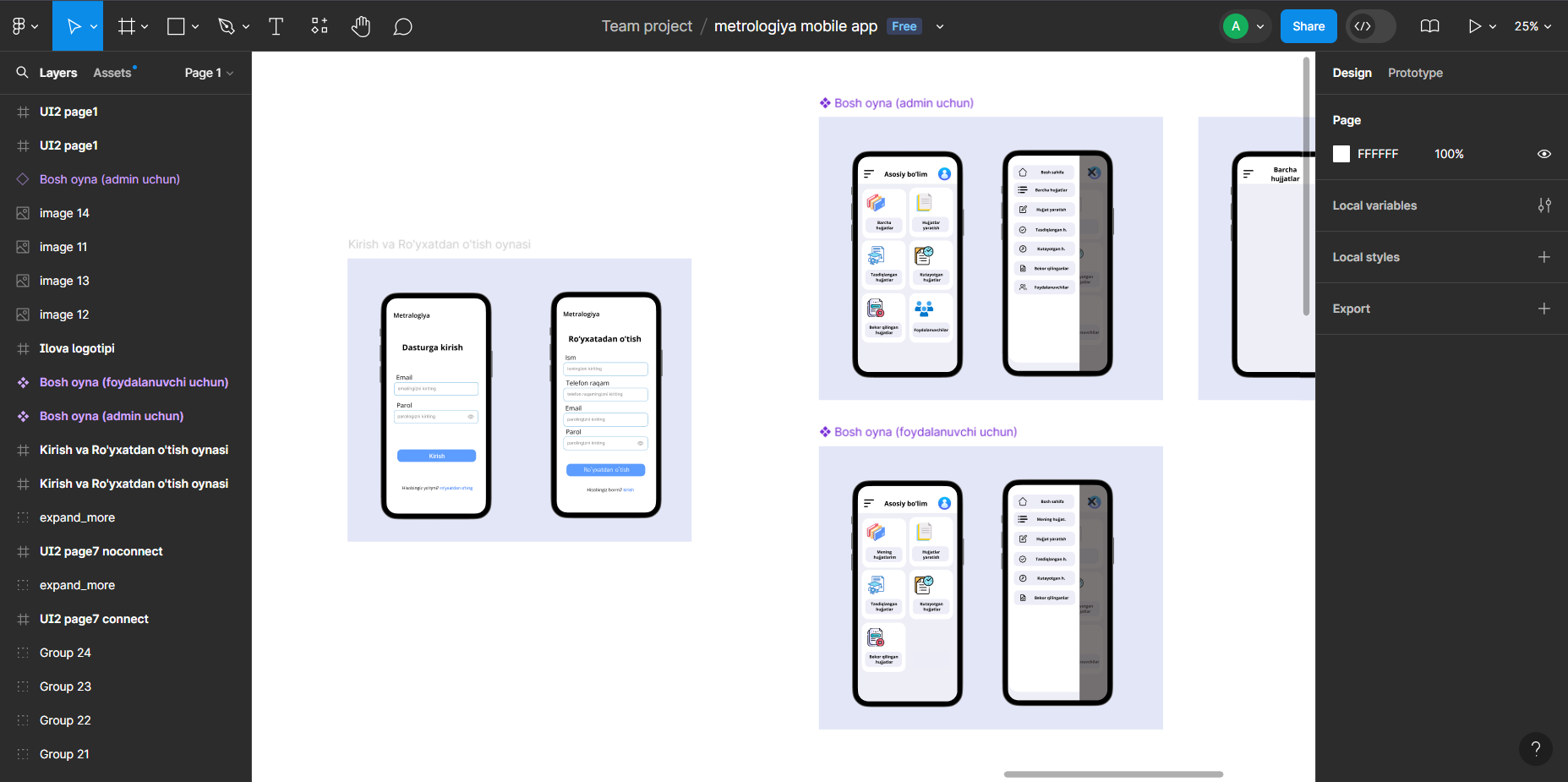
Foydalanuvchilarga qulayroq va sodda bo‘lishi uchun dizaynini tayyorlash muhimdir. Bu jarayonda, mobil dastur dizayni yaratishda figma dasturidan foydalanildi va bunda quyidagi bosqichlardan o‘tildi:

1. Dastur mobil ilovasining belgisini yaratish. Bunda bir qancha jihatlarga e’tibot berish kerak: dasturni boshqa ilovalardan ajralib turishi, tez tanib olish uchun aniq bo‘lishi muhimdir.



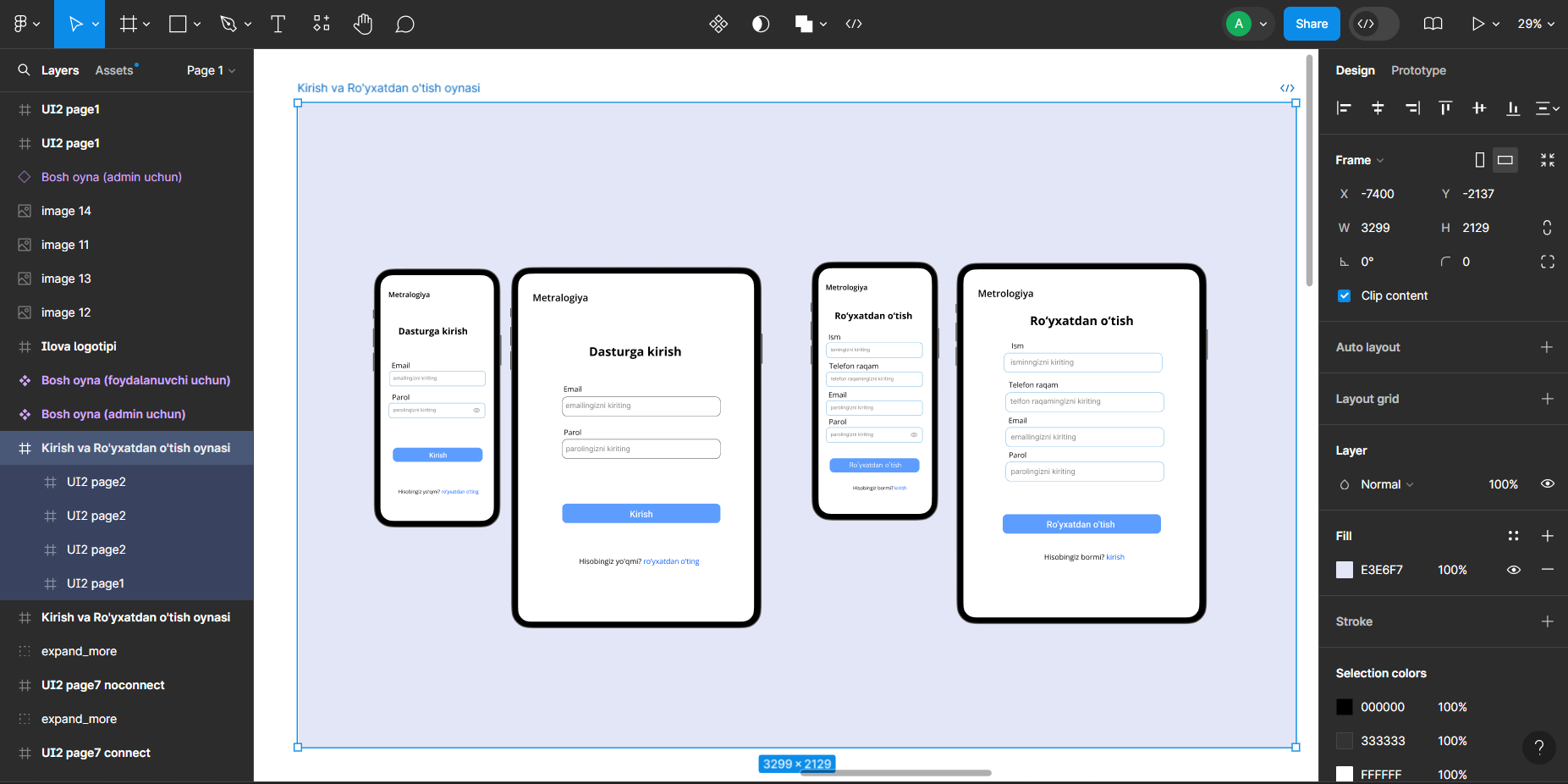
*3.6-rasm. Ilovaning logotipi*

1. Ilovaning har bir qismi uchun alohida dizaynlarini yaratish. Bunda mobil dasturni qismlarga ajratib olish kerak: kirish oynasi, asosiy oyna, sertifikatlar oynasi va boshqa oynalar (3.7-rasm). Oynlalarni shakillantirishda har bir rol uchun alohida xususiyatlarga ega bo‘lishini lozim.



*3.7-rasm. Dastur qismlarining dezaynini yaratish*

1. Dasturni boshqa qurilmalari uchun ham moslash. Dastur barcha mobil va planshet qurilmalarda ekran o‘lchamidan qat’iy nazar to‘g‘ri va belgilangandek ishlashi uchun ilova dezaynini moslashuvchan qilib yaratish kerak (3.8-rasm).

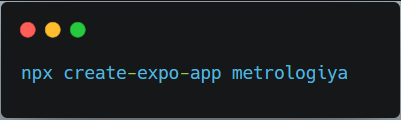


*3.8-rasm. Dasturni planshet qurilimasiga moslash*

* 1. **Dasturiy ta’minotning fizik yaratilishi**

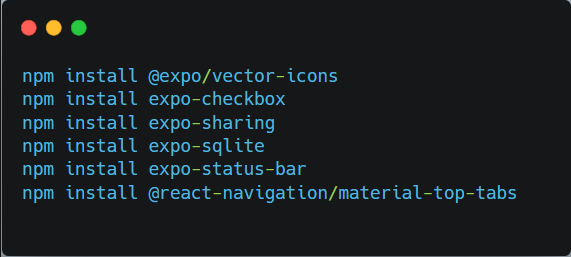
Mobil ilovani yaratish uchun quyidagi qadamlar bajariladi:

1. Dastur tuzilmasini yaratish: dastlab dasturning boshlang‘ich tuzilishini yaratish uchun expo cli dan foydalanamiz. Terminal orqali quyidagi kodni ishga tushirish kerak bo‘ladi (3.9-rasm):



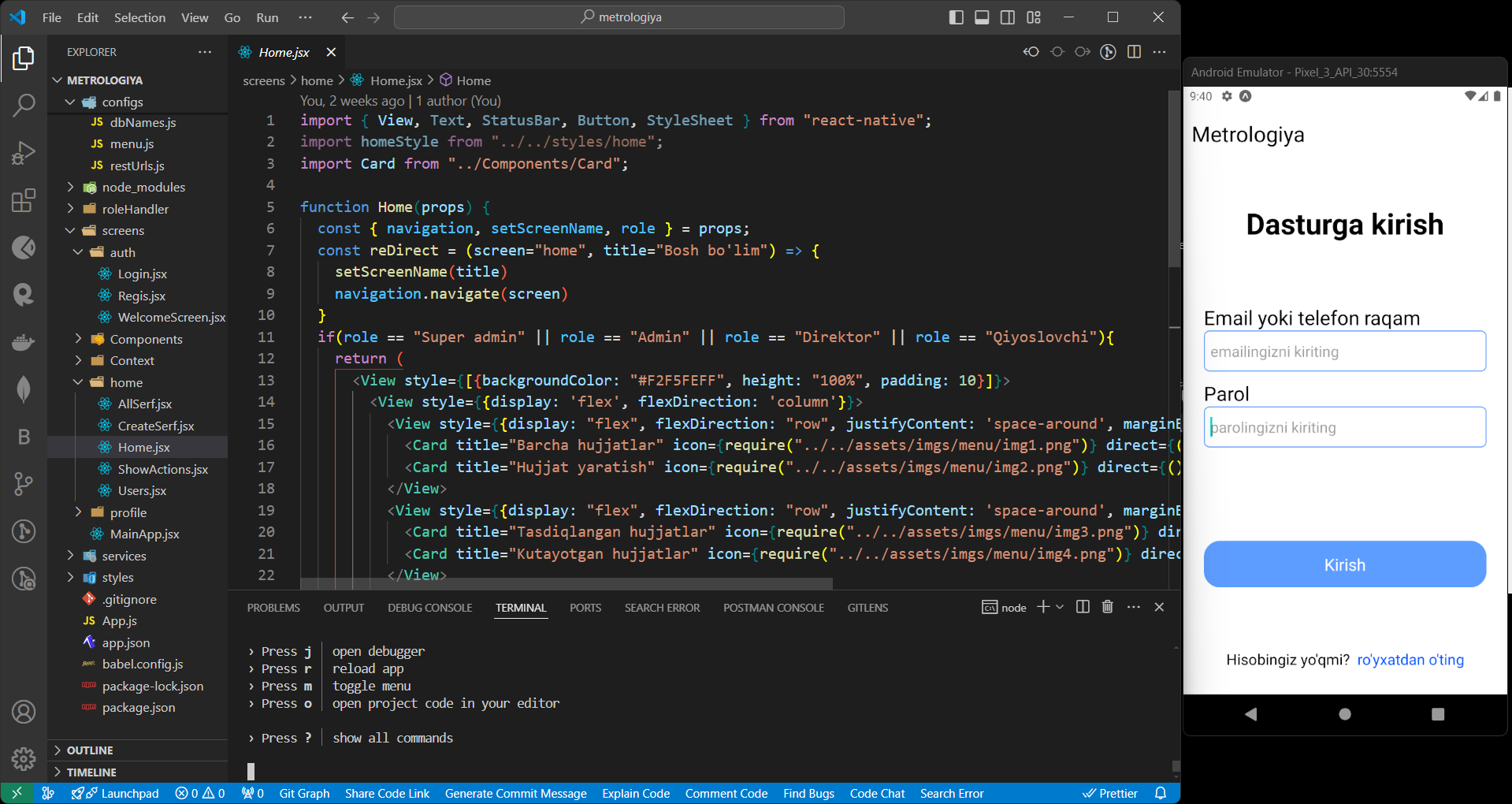
*3.9-rasm. Dastur yaratish komandasi.*

1. Kerakli kutubxonalarni yublab olish: dastur boshlang‘ich tuzilishi yartilganidan so‘ng, kerakli kutubxonalarni yuklab olinamiz (3.10-rasm).



*3.10-rasm. Kerakli kutubxonalarni yuklash*

1. Dasturni emulyator orqali ishga tushirish va oyalarni yaratish (3.11-rasm):



*3.10-rasm. Dastur yaratish*

4. Dasturning qolgan oynalarining yaratilishi:

* Tizimda avtorizatsiyani boshqarish uchun ro‘yxatdan o‘tish va tizimga kirish oynasini yaratildi va dasturning backend qismi bilan bog‘lanish (3.11-rasm).
* Dasturdagi bosh oyna bilan ishlash: har bir roldagi foydalanuvchi uchun alohida dastur xususiyatlarini taqdim qilish (3.12-rasm).
* Dasturda sertifikatlar bilan ishlovchi qisim: tizimga kirgan foydalanuvchi roliga va huquqlariga qarab, sertifilatlar bilan ishlash jarayonini tashkil qilish (3.13- rasmlar).

|  |  |
| --- | --- |
| D:\Main programm files\Desktop\Screenshot_1715521174.png | D:\Main programm files\Desktop\Screenshot_1715521178.png |

*3.11-rasm. Dastur kirish va ro‘yxatdan o‘tish oynalari*

|  |  |
| --- | --- |
| D:\Main programm files\Desktop\Screenshot_1715521189.png | D:\Main programm files\Desktop\Screenshot_1715521193.png |

*3.12-rasm. Dastur bosh oynasi va menusi*

|  |  |
| --- | --- |
| D:\Main programm files\Desktop\Screenshot_1715521233.png | D:\Main programm files\Desktop\Screenshot_1715521243.png |

*3.13-rasm. Dastur hujjat yaratish va hujjatlarni ko‘rish oynasi*

# **Xulosa**

Bitiruv malakaviy ishi metrologiya sohasida sertifikatlashtirish jarayonini avtomatlashtirishdan iborat. Bunda bizga ma’lum dasturlash tilining tuzilmalarini o‘rganish, ma’lumotlar bazasi bilan bog‘lash, frameworklar bilan ishlash va mobil dasturlar ishlab chiqish kabi ustuvor vazifalar qo‘yildi. O‘rganish davomida ko‘plab yangi texnalogiyalar bo‘yicha tushunchalarga ega bo‘lindi va ishlab chiqilgan dasturning amaliyotga tadbiqi o‘rganildi.

Ushbu ishlab chiqiladigan dasturiy ilovaning afzalliklari shundan iboratki, oldin sertifikatlashtirish jarayonida qo‘lda bajarilgan. Bunda sertifikatlar yaratish ko‘p vaqtni talab qilgan va inson omili sababli ba’zida xatoliklarga yo‘l quyilgan. Yaratilgan dastur bilan esa tizim sertifikatni o‘zi generatsiya qilgani uchun sertifikatni yaratish vaqti sezilarli miqdorda tezlashgan va hujjat yaratishda davomida xatoliklar qilinishi ehtimoli pasaytirilgan.

Foydalanuvchilarga veb dasturdan yanada foydalanishni yanada qulaylashtirish uchun mobil ilovalar ishlab chiqish va veb dasturning backend qismi bilan REST API orqali integratsiya qilindi. Bu orqali har qanday foydalanuvchi, jumladan, admindan tortib qiyoslovchigacha barcha rol uchun usha rol xususiyatlarini taqdim etish orqali foydalanuvchilar huquqlarini chegaralash yoki kengaytirish imkoniyati amalga oshirilgan.

Mobil qurilmani ishlatish kompyuter qurilmani ishlatishdan ancha qulay bo‘lishi hisobiga dasturning mobil ilovasidan foydalanadiganlar soni ko‘p va bu istalgan holatda ishlata olishni ta’minlaydi.

# **FOYDALANILGAN ADABIYOTLAR**

1. Burdun G. D., Markov B. N., Osnovi metrologii, M., 1972; Dushin Ye. M., Osnovi metrologii i elektricheskiye izmereniya, M., 1987.
2. I.Kh. Siddikov, Kh. A. Sattarov, O. I. Siddikov, X.E. Khujamatov, G.N. Suleymanova, D. T. Khasanov, Sh. B. Olimova, METROLOGY, STANDARDIZATION AND CERTIFICATION, Toshkent, 2018
3. A. Qurbanov A. Axmedovich, “Metrologiya, standartlashtirish va sertifikatlashtirish” 2007.
4. С.В. Пономарев, Г.В. Шишкина, Г.В. Мозгова, “МЕТРОЛОГИЯ, СТАНДАРТИЗАЦИЯ, СЕРТИФИКАЦИЯ” 2010.
5. Tejinder S. Randhawa, “Mobile Applications Design, Development and Optimization” 2022.
6. Shaun Lewis, Mike Dunn, “Native Mobile Development: A Cross-Reference for iOS and Android” 2019.
7. Tobias Baumgärtner, “A Software Framework for Mobile Apps in the Museum Application Domain (Gabler Theses)” 2024.
8. Chris Griffith, “Mobile App Development with Ionic, Revised Edition: Cross-Platform Apps with Ionic, Angular, and Cordova” 2017.
9. Vlad Filippov, “Building Your Own JavaScript Framework: Architect extensible and reusable framework systems” 2023.

**FOYDALANILGAN INTERNET MANBALARI**

1. <https://www.turing.com/blog/cross-platform-vs-native-mobile-development/>
2. <https://ru.wikipedia.org/wiki/JavaScript>
3. <https://www.geeksforgeeks.org/top-javascript-frameworks-for-mobile-apps-development/>
4. <https://reactnative.dev/docs/getting-started-without-a-framework>

**ILOVALAR**

1. **App.js:** Mobil ilovaning asosiy ishga tushuvchi faylibo‘lib, qolgan bo‘limlar shu oyna orqali yaratiladi va boshqariladi.

import Login from "./screens/auth/Login";

import Regis from "./screens/auth/Regis";

import MainApp from "./screens/MainApp";

import WelcomeScreen from "./screens/auth/WelcomeScreen";

import { useFonts } from "expo-font";

import { SafeAreaProvider } from "react-native-safe-area-context";

import { NavigationContainer } from "@react-navigation/native";

import { View } from "react-native";

import { getToken, createTokenTable } from "./services/tokenServer";

import {tableDataNames} from "./configs/dbNames";

import { useEffect, useState } from "react";

export default function App() {

  const [fontsLoaded] = useFonts({

    openSans: require("./assets/fonts/openSans.ttf"),

    Roboto\_Medium: require("./assets/fonts/Roboto-Medium.ttf"),

    Roboto\_Regular: require("./assets/fonts/Roboto-Regular.ttf"),

  });

  const [token, setToken] = useState('');

  const [role, setRole] = useState('user');

  const [initialized, setInitialized] = useState(false);

  useEffect(()=>{

    if (!initialized) {

      createTokenTable();

      setInitialized(true);

    }

    if (!token) {

      getToken(tableDataNames.token)

        .then((data) => {

          if(data){

            setToken(data);

            //get role using getToken()

            getToken(tableDataNames.role).then((data1)=>{

              if(data1){

                setRole(data1);

              }else console.log(data1);

            })

          }else{

            createTokenTable()

          }

        })

        .catch(err=>console.log(err))

    }

  }, [initialized])

  return (

      <SafeAreaProvider>

        <NavigationContainer>

          {

            (!token)?

            <WelcomeScreen setToken={setToken} setRole={setRole} />:

            <MainApp

              setToken={setToken}

              token={token}

              setRole={setRole}

              role={role}

            />

          }

        </NavigationContainer>

      </SafeAreaProvider>

  );

}

1. **MainApp.jsx:** bosh oynadan keyingi asosiy oyna bo‘lib, dastur xususiyatlari bo‘limlarga yo‘naltirish shu oynadan bajariladi.

import { Image, View, StatusBar, TouchableOpacity, Text } from "react-native";

import { createNativeStackNavigator } from "@react-navigation/native-stack";

import { useState } from "react";

import mainStyle from "../styles/mainStyle";

import { Ionicons } from "@expo/vector-icons";

import { deleteToken } from "../services/tokenServer";

//screens

import Home from "./home/Home";

import AllSerf from "./home/AllSerf";

import CreateSerf from "./home/CreateSerf";

import Users from "./home/Users";

import { defaultStyle } from "../styles/defaultStyle";

import Menu from "./Components/Menu";

import MenuNames from "../configs/menu";

import Profile from "./profile/Profile";

import ViewDoc from "./Components/ViewDoc";

import ShowActions from "./home/ShowActions";

import { tableDataNames } from "../configs/dbNames";

const Stack = createNativeStackNavigator();

export default function MainApp(props) {

  const { setToken, token, setRole, role } = props;

  const [screenName, setScreenName] = useState("Bosh bo‘lim");

  const [menuState, setMenuState] = useState(false);

  const [profileState, setProfileState] = useState(false);

  const [viewDoc, setViewDoc] = useState(false);

  const changeScreenName = (name) => {

    setTimeout(() => {

      setScreenName(name);

    }, 1000);

  };

  const destroyUserData = () => {

    //destroy token

    deleteToken(tableDataNames.token);

    setToken("");

    //destroy role

    deleteToken(tableDataNames.role);

    setRole("");

  };

  return (

    <View>

      <StatusBar barStyle="dark-content" backgroundColor={"#fff"} />

      <View

        style={[

          {

            position: "absolute",

            top: 0,

            left: 0,

            backgroundColor: "#0005",

            width: "100%",

            height: "100%",

            zIndex: 100,

          },

          defaultStyle.row,

          { display: menuState ? "flex" : "none" },

        ]}

      >

        <Menu setMenuState={setMenuState} />

        <TouchableOpacity

          style={{ width: "30%", height: "100%", zIndex: 101 }}

          onPress={() => setMenuState(false)}

        >

          <Ionicons

            name="close-sharp"

            size={50}

            color="#000"

            style={{ margin: 5 }}

          />

        </TouchableOpacity>

      </View>

      <View

        style={[

          {

            position: "absolute",

            top: 0,

            left: 0,

            backgroundColor: "#fff",

            width: "100%",

            height: "100%",

            zIndex: 100,

          },

          defaultStyle.row,

          { display: profileState ? "flex" : "none" },

        ]}

      >

        <Profile

          setProfileState={setProfileState}

          destroyUserData={destroyUserData}

        />

      </View>

      <View

        style={[

          {

            position: "absolute",

            top: 0,

            left: 0,

            backgroundColor: "#fff",

            width: "100%",

            height: "100%",

            zIndex: 100,

          },

          defaultStyle.row,

          { display: viewDoc ? "flex" : "none" },

        ]}

      >

        <ViewDoc setViewDoc={setViewDoc} />

      </View>

      <View style={mainStyle.header}>

        <View

          style={[

            defaultStyle.row,

            defaultStyle.between,

            { paddingHorizontal: 5 },

          ]}

        >

          <TouchableOpacity

            style={{ width: 40 }}

            onPress={() => setMenuState(true)}

          >

            <Image

              source={require("../assets/imgs/menu.png")}

              style={{ width: 40, height: 40 }}

            />

          </TouchableOpacity>

          <Text style={{ fontSize: 25 }}>{screenName}</Text>

          <TouchableOpacity

            style={{ width: 45 }}

            onPress={() => setProfileState(true)}

          >

            <Image

              source={require("../assets/imgs/profile.png")}

              style={{ width: 40, height: 40 }}

            />

          </TouchableOpacity>

        </View>

      </View>

      <View style={{ width: "100%", height: "95%" }}>

        <Stack.Navigator

          screenOptions={{ headerShown: false }}

          initialRouteName="home"

          screenListeners={({ route }) => {

            changeScreenName(MenuNames[route.name]);

          }}

        >

          <Stack.Screen name="home">

            {(props) => <Home {...props} setScreenName={setScreenName} role={role} />}

          </Stack.Screen>

          <Stack.Screen name="all">

            {(props) => (

              <AllSerf {...props} winType={"all"} setViewDoc={setViewDoc} token={token} />

            )}

          </Stack.Screen>

          {(role == "Super admin" || role == "Admin" || role == "Direktor" || role == "Qiyoslovchi") && (

            <Stack.Screen name="create">

              {(props) => <CreateSerf {...props} />}

            </Stack.Screen>

          )}

          {(role == "Super admin" || role == "Admin" || role == "Direktor") && (

            <Stack.Screen name="verify">

              {(props) => <AllSerf {...props} winType={"verify"} />}

            </Stack.Screen>

          )}

          {(role == "Super admin" || role == "Admin" || role == "Direktor") && (

            <Stack.Screen name="pending">

              {(props) => <AllSerf {...props} winType={"pending"} />}

            </Stack.Screen>

          )}

          {(role == "Super admin" || role == "Admin" || role == "Direktor") && (

            <Stack.Screen name="actions">

              {(props) => <ShowActions {...props} />}

            </Stack.Screen>

          )}

          {(role == "Super admin" || role == "Admin" || role == "Direktor") && (

            <Stack.Screen name="users">

              {(props) => <Users {...props} />}

            </Stack.Screen>

          )}

        </Stack.Navigator>

      </View>

    </View>

  );

}