**MUNDARIJA**

[**KIRISH 6**](#_Toc137439219)

[**I – BOB. TEXNOLOGIYLAR VA TIZIMLARNI TAHLILI. 9**](#_Toc137439220)

[**1.1-§ Yetkazib berish xizmatini elektronlashtirishda 9**](#_Toc137439221)

[**maʻlumotlar tahlili 9**](#_Toc137439222)

[**1.2-§ Veb ilovalarni ishlab chiqish muhitlari haqida 16**](#_Toc137439223)

[**1.3-§ Masalaning qoʻyilishi 20**](#_Toc137439224)

[**II-BOB. ASOSIY QISM. 21**](#_Toc137439225)

[**2.1-§ Axborot tizimini ishlab chiqishda tanlangan texnologiyalar 21**](#_Toc137439226)

[**2.2-§ Dastur interfeyslarining tavsifi 32**](#_Toc137439227)

[**2.3-§ Foydalanuvchiga yoʻriqnoma 45**](#_Toc137439228)

[**XULOSA 46**](#_Toc137439229)

[**FOYDALANILGAN ADABIYOTLAR ROʻYHATI 47**](#_Toc137439230)

# KIRISH

**Mavzuning dolzarbligi.** Buyurtmalarni shakillantirish va mahulotlarni tashish va yuklarni yetkazib berish jarayonlari turli xil axborotlarni tizimli ravishda yuritishni talab qiladi. Tizimli axborotlarni yuritish insonlarning hayotini yengillashtirish, vaqtdan va mablagʻdan yutish uchun juda katta yordam beradi. Bu odamlarga oson va aniq maʻlumotlar berilishi yordamida amalga oshiriladi.

Tizimli axborotlar, yuklarni yetkazib berish sohasida bir necha xil tizimlarda yuritilishi mumkin. Bu tizimlar, mahsulotlar, yuklar, mijozlar va eltuvchilarni faoliyatiga oid boshqa maʻlumotlarni kuzatish, toʻplash va boshqarish uchun foydalaniladi.

Tizimli axborotlarni yuritishning bir necha foydalari mavjud. Birinchidan, bu xatoliklar va kamchiliklarni yuzaga kelishini kamaytiradi va xatoliklarni tez-tez aniqlash va bartaraf qilish imkoniyatini beradi. Ikkinchidan, tizimli axborotlar, mijozlar soʻrovilari bilan ishlash faoliyatini yanada samarali va sezilarli qiladi, chunki tizim va operatorlar bilan avtomatik ravishda bogʻliq maʻlumot almashish imkonini beradi. Uchinchi, tizimli axborotlar, mijozlar va boshqa tashuvchi taraflarga xizmat koʻrsatishni yanada samarali qiladi, chunki mijozlar oʻz soʻrovlariga tez va aniq javob olishlari mumkin.

Axborot tizimlari, buyurtmalar va yetkazib berish jarayonini avtomatlashtirish va boshqarish yordamida xatolikni kamaytirishga yordam beradi. Bu buyurtmalarni avtomatik ravishda qabul qilish, yetkazib berishni tizimlashtirish va mijozlar bilan aloqa qilishga imkon beradi. Axborot tizimlari, buyurtmalar va yetkazib berish jarayonini kuzatish uchun qoʻllaniladigan vositalarni oʻz ichiga oladi. Bu mijozlarga toʻgʻri vaqtda xizmat koʻrsatish, yetkazib berish jarayonini optimallashtirish va mijozlar bilan aloqani yaxshilashga imkon beradi. Axborot tizimlari, buyurtmalar va yetkazib berish jarayonini barcha taraflar orasida bir-biriga bogʻliq axborotlar almashishga imkon beradi. Mijozlarga toʻgʻri vaqtda xizmat koʻrsatish, buyurtmalarni avtomatlashtrish va yetkazib berish jarayonini optimallashtirishga yordam beradi. Axborot tizimlari, yetkazib berish jarayonini kuzatish uchun bosqichlarini oʻz ichiga oladi. Bu mijozlarga toʻgʻri vaqtda xizmat koʻrsatish, yetkazib berish jarayonini optimallashtirish va mijozlar bilan aloqani yaxshilashga imkon beradi. Axborot tizimlari, mijozlar va yetkazib berish xizmatlari orasida aniq axborot almashishga imkon beradi. Bu mijozlar uchun xizmatni sifatini yaxshilash, buyurtmalar va yetkazib berish jarayonini optimallashtirish va mijozlar bilan aloqni yaxshilashga imkon beradi. tizimlari, buyurtmalar va mijozlar haqida maʻlumotlar analizini amalga oshirishga imkon beradi. Bu mijozlar uchun yaxshi xizmat koʻrsatishni va buyurtmalar va yetkazib berish jarayonini optimallashtirishni osonlashtiradi.

Yetkazib berish jarayonini optimallashtirish uchun, GPS va loyihalash tizimlaridan foydalanish mumkin. Bu tizimlar, buyurtmalar va yetkazib berish xizmati tomonidan ishlatiladigan transport vositalarini chuqur kuzatishga yordam beradi. Bu, yetkazib berish jarayonini optimallashtirish uchun kerakli maʻlumotlarni yigʻishga, harakatni kuzatishga va buyurtmalar va yetkazib berish jarayonini barcha taraflar orasida koʻrsatishga imkon beradi. Mijozlar bilan aloqa tizimlaridan foydalanish, mijozlar bilan yaxshi aloqa taʻminlash, ularning talablari va shikoyatlari bilan qiziqtirgan vaqtda yordam berishga imkon beradi. Bu tizimlar, mijozlar bilan ovozli va matnli ravishda aloqa oʻrnatishga imkon beradi, shuningdek buyurtmalar va yetkazib berish jarayonini kuzatishga yordam beradi, mijozlarga real vaqtda xizmat koʻrsatish va yetkazib berish jarayonini optimallashtirish uchun kerakli maʻlumotlarni yigʻadi.

**Bitiruv malakaviy ishining obyekti va predmeti:** Diplom ishining obyekti buyurtmalarni shakillantirish va yetkazib berilishini nazorat qilish axborot tizimi sohasidagi axborot tizimlardir. Bu tizimlar mijoz yuklari va ularga bogʻliq tashuvchi taraflar tomonidan mahsulotlar, yuklar va boshqa maʻlumotlar jamlanishini, saqlanishini, nazorat qilishini va boshqarishini osonlashtirish uchun ishlab chiqiladi.

Diplom ishining predmeti esa yuklarni eltuvchi xizmatning hisob-kitobi va axborot tizimlarining yaratilishi, rivojlanishi va ularning yuritilishi haqida tushunchalar va koʻnikmalar hisoblanadi. Bu tushunchalar va koʻnikmalar yuklarni eltuvchi xizmat sohasidagi axborot tizimlarining yaratilishiga, rivojlanishiga va ularga oid hisob-kitob jarayonlarini yuritishga oid, shuningdek, bu tizimlarning mijozlar uchun qulayliklarini taʻminlash va mijozlar bilan ish faoliyatini yanada samarali qilishga oid.

**Bitiruv ishining maqsadi:** Bu diplom ishining maqsadi buyurtmalarni shakillantirish va yetkazib berish axborot tizimini yaratish, rivojlantirish va ularga oid hisob-kitob jarayonlarini yuritishga oid tushunchalar va koʻnikmalar yaratishdir. Bu ishning maqsadi, mijozlarning yklarini yetkazib berish sohasidagi axborot tizimlarining muhimligini va ulardagi muammolarni bartaraf qilish yoʻllarini oʻrganish, mijozlar uchun qulayliklarni taʻminlash va ishonchli ish faoliyatini yanada samarali qilishga oid axborot tizimlarini ishlab chiqishni oʻrganishdir.

**Bitiruv malakaviy ishining tuzilmaviy tasnifi.** Bitiruv malakaviy ishining tuzilishiga koʻra: kirish, ikkita bob, xulosa, foydalanilgan adabiyotlar roʻyxatidan iborat.

Ishning birinchi bobida Buyurtmalarni shakillantirish va yetkazib berishni nazorat qilish axborot tizimi berishi mumkin boʻlgan foydali tomonlari haqida aytib oʻtamiz va bu Veb-ilovaning qanday yaratilishi, umuman oʻzi Veb-ilovalar qanday yaratilishi, qaysi Veb texnologiyalar, Veb-ilovalarni ishlab chiqishda samarali va foydali boʻlishi haqida aytib oʻtamiz.

Ishning ikkinchi bobida Veb-ilovalarni yaratish texnologiyalari, frontend uchun ajoyib framework Angular frameworki yordamida Veb-ilovaning frontend qismini ishlab chiqamiz. Frontend qismidan keyin ilovaning backend qismini yozish va ishlash muhiti, ilova koʻrinishlari joy olgan.

Ishning xulosa qismida oʻrganilgan maʻlumotlar asosida xulosalar qilingan va foydalanilgan adabiyotlar va Veb-saytlar keltirib oʻtilgan.

# I – BOB. TEXNOLOGIYLAR VA TIZIMLARNI TAHLILI.

# 1.1-§ Yetkazib berish xizmatini elektronlashtirishda

# maʻlumotlar tahlili

Buyurtmalarni shakillantirish va yetkazib berishni nazorat qilish axborot tizimini yaratishda biz mijoz va firmaning qulaylik va funksianallik inkoniyatlaiga eʻtibor qaratdik. Bunda tizimda har bir foydalanuvchi uchun alohida profil ochilib admin, mijoz, eltuvchi alohida oʻz sahifalariga kirib biriktirilgan vazifani bajarishlari mumkin.

Tizim funksiyalariga quyidagilar kiradi:

1. Buyurtmalar qabul qilish: Buyurtma xizmati, mijozlar tomonidan berilgan buyurtmalarni qabul qiladi. Mijozlarga toʻgʻri vaqtda xizmat koʻrsatish uchun kerakli transport vositalarini tizimlashtirish, yetkazib berish jarayonini optimallashtirish va xatolikni kamaytirishga yordam beradi.
2. Buyurtma xizmati, mijozlardan toʻlovni qabul qiladi. Toʻlovlar, mijozlar uchun qulay toʻlov usullari orqali amalga oshiriladi, masalan, onlayn toʻlov, avtomatik toʻlov, kredit kartalari yoki pul koʻchirish orqali.
3. Buyurtmalarni shakillantirish: Buyurtma xizmati, mijozlar tomonidan berilgan buyurtmalarni shakillantiradi. Bu shakllantirish jarayoni, mijozlar tomonidan berilgan buyurtmalarni tuzatish, yetkazib berish uchun kerakli maʻlumotlarni yigʻish va yetkazib berish xizmatiga tayyorlashni oʻz ichiga oladi.
4. Yetkazib berish jarayonini tizimlashtirish: Buyurtma xizmati, yetkazib berish jarayonini tizimlashtirish uchun bir necha vositalardan foydalanadi. Bu, mijozlarga toʻgʻri vaqtda xizmat koʻrsatish uchun kerakli transport vositalarini tizimlashtirish, yetkazib berish jarayonini optimallashtirish va xatolikni kamaytirishga yordam beradi.
5. Buyurtmalarni joylashtirish: Buyurtma xizmati, mijozlar tomonidan berilgan buyurtmalarni joylashtiradi. Bu, buyurtmalar uchun kerakli transport vositalarini tayyorlash, yetkazib berish jarayonini tizimlashtirish va mijozlar bilan aloqni oʻz ichiga oladi.
6. Yetkazib berish xizmatini amalga oshirish: Buyurtma xizmati, mijozlar tomonidan berilgan buyurtmalarni yetkazib berish xizmatiga oʻtkazadi. Yetkazib berish xizmati, mijozlarga toʻgʻri vaqtda va buyurtmalariga mos ravishda xizmat koʻrsatish uchun kerakli transport vositalarini tayyorlash, yetkazib berish jarayonini tizimlashtirish va mijozlar bilan aloqni oʻz ichiga oladi.
7. Mijozlar bilan aloq: Buyurtma xizmati, mijozlar bilan aloqni taʻminlash uchun bir necha vositalardan foydalanadi. Bu, mijozlarga xizmatni sifatli ravishda koʻrsatish, ularning shikoyatlarini yoki talablari bilan shugʻullanish va mijozlar bilan yaxshi aloqni taʻminlashga yordam beradi.
8. Buyurtmalarni kuzatish: Buyurtma xizmati, buyurtmalarni kuzatish uchun bir necha vositalardan foydalanadi. Bu, mijozlarga toʻgʻri vaqtda xizmat koʻrsatish, yetkazib berish jarayonini optimallashtirish va mijozlar bilan aloqni oʻz ichiga oladi.

Bu erda asosiy talablar yuk oqimining uzluksizligi va yuk oqimining miqyosidagi iqtisodlarga erishishdir. Yuk oqimining uzluksizligi deganda, marshrutning alohida uchastkalariga bir necha kishi yoki asbob-uskunalar xizmat koʻrsatishdan koʻra, bitta yuk koʻtaruvchi yoki yuk ortish moslamasi yukni belgilangan joyga koʻchirish yaxshi boʻlishini anglatadi. Yukni qoʻldan qoʻlga oʻtkazish yoki bir forkliftdan ikkinchisiga qayta yuklash nafaqat vaqtni yoʻqotishga olib keladi, balki yukning shikastlanish xavfini ham oshiradi. Shunday qilib, odatda, ombor ishlarida uzoqroq va kamroq tez-tez yoʻnalishlarga afzallik beriladi.

Yuk tashishda miqyosning tejamkorligi shuni anglatadiki, har bir operatsiya imkon qadar koʻproq yukni tashish kerak: bitta paketda emas, balki paketlar toʻplamida - palletlar yoki konteynerlar. Ushbu toʻplam bir vaqtning oʻzida turli xil mahsulotlar yoki bir nechta buyurtmalarning tarkibiy qismlarining harakatlanishiga olib kelishi mumkin. Albatta, bu qoʻshimcha qiyinchiliklarni keltirib chiqaradi, ammo shunga qaramay, operatsiyalar sonini va shunga mos ravishda xarajatlarni kamaytirishga harakat qilish kerak.

Saqlash maydonining joylashuvi. Uchinchi tamoyilga koʻra, omborni loyihalashda saqlanadigan tovarlarning jismoniy xususiyatlarini - birinchi navbatda, hajmi, vazni va saqlash sharoitlarini hisobga olish kerak.

U yoki bu rejalashtirish yechimini tanlashni belgilovchi asosiy omil yukning hajmi (hajmi) hisoblanadi. Katta yuk tashish yoki tranzit yuklarni eng qisqa yuklash va tushirish yoʻllari yaqinida, yaʻni asosiy yoʻlaklar yaqinida va tokchalarning pastki javonlarida saqlash kerak. Bu sayohat masofasini qisqartiradi. Aksincha, kichik hajmli yuklarni asosiy yoʻlaklardan uzoqda va raftlarning yuqori javonlarida joylashtirish mumkin.

Xuddi shunday, saqlash joylarini rejalashtirishda ogʻirlik va saqlash sharoitlari kabi yuk xususiyatlarini hisobga olish kerak. Nisbatan ogʻirroq yuklarni koʻtarish paytida shikastlanish xavfini va buning uchun zarur boʻlgan harakatlarni kamaytirish uchun imkon qadar pastroq joylashtirilishi kerak. Ommaviy yoki past zichlikdagi yuklarni joylashtirish juda koʻp boʻsh joyni talab qiladi, shuning uchun ular ochiq joy yoki yuqori tomonlari boʻlgan tokchalar bilan taʻminlanishi kerak. Boshqa tomondan, kichik oʻlchamdagi tovarlarni saqlash uchun kichik hujayrali tokchalar talab qilinadi. Shuning uchun omborning tartibi saqlanadigan mahsulotlarning barcha turlarining xususiyatlarini aks ettirishi kerak.

Biroq, baʻzi hollarda bu tamoyillar bir-biriga zid keladi. Masalan, texnologik sabablarga koʻra tovarlarni saqlash joylaridan yigʻish va buyurtmalarni yigʻish uchun bitta uskunadan foydalanish, ikkinchisini esa ularni tirkamaga yuklash foydaliroq boʻlishi mumkin. Bu yuklarni bir forkliftdan boshqasiga oʻtkazish kerakligini anglatadi, bu esa qoʻshimcha vaqtni oladi. Biroq, maʻlum boʻlishicha, ushbu operatsiyalarning har birida maxsus jihozlardan foydalanish umumiy xarajatlarni kamaytiradi.

Yongʻinning oldini olish choralari, ayniqsa, yoqilgʻi-moylash materiallari va yonuvchan moddalar omborda saqlangan boʻlsa, muhim jihatdir. Biroq, bularning barchasi umumiy tamoyillarning rolini kamaytirmaydi.

**Jarayonlarni tashkil etishning logistik konsepsiyasi -** faqatgina logistika sohasida mutaxassislar tomonidangina amalga oshirib hoʻlmaydi, Jarayonlarni tizimli tashkil etish falsafasi biznesning tayanch falsafasi boiishi lozim. Tijoratchilar, iqtisodchilar, turli larmoq menejerlari logistika konsepsiyasini tushunishlari va qabul qilishlari, jarayonlarni logistik tashkil etishning asosiy usullarini bilishlari hamda erishiladigan samarani koʻzda tutadigan boʻlishlari kerak. Shuning uchun ham bugungi kunda logistika sohasidagi keng qamrovdagi oliy taʻlim vazifasi nihoyatda muhim hisoblanadi.

Oʻzbekiston Respublikasi Birinchi Prezidenti I.A. Karimov Oʻzbekiston Respublikasi Oliy Majlisi Qonunchilik palatasi va Senatining qoʻshma majlisidagi maʻruzasida Oʻzbekiston Respublikasi iqtisodiyotini yanada rivojlantirish borasida toʻxtalib: «... bizning yaqin istiqboldagi eng muhim vazifamiz boshlagan ishlarimizni izchil davom ettirish — isteʻmol talabini kengaytirish maqsadida sotsial sohani rivojlantirish, mehnatga haq toʻlashni yanada oshirish, xizmat koʻrsatish sektorini, infratuzilma obyektlarini rivojlantrishga, transport va kommunikatsiya loyihalarining amalga oshirilishiga alohida eʻtibor bcrishdir»1 deb taʻkidlab oʻtdilar. Shuningdek, I.A. Karimov Oʻzbekiston Respublikasi Vazirlar Mahkamasining 2011-yilning asosiy yakunlari va 2012-yilda Oʻzbekistonni ijtimoiy-iqtisodiy rivojlantirishning eng muhim ustuvor yoʻnalishlariga bagʻislilangan maʻruzasida Oʻzbekiston iqtisodiyoti oldida turgan asosiy vazifalar toʻgʻrisida toʻxtalib, «... bugungi. keskin raqobat sharoitida mahsulotlarimizning jahon va mintaqaviybozorlarda xaridorgir boiishi va mustahkam oʻrin egaliashi uchun bu boradagi ishlarni yana bir bor tanqidiy koʻrib chiqish lozim.

Hunda tasliki bo/orda xaridorbop, yuqori likvidli mahsulotlar ishlab chiqarishni kengaytirish uchun eksportga mahsulot chiqaradigan korxonalarni ragʻbatlantirishni yanada kuchaytirish, ularga yangi imiiyoz va pieferensiyalar berish boʻyicha qoʻshimcha choraladbirlar tayyorlash zarur»ʻ deb taʻkidladilar.

Maʻlumki, gʻarb logistikasi modellari mamlakatimiz xoʻjalik amaliyotiga har doim mos kelavermaydi. Logistik tizimlar — bu atrof-m uhit bilan uzviy aloqasiz qarab boʻlmaydigan jonli tizimlardir. (iʻarb tizimlaridan nusxa olish strategiyasi — bu egalik strategiyasi boʻlib, biroq yetakchilik strategiyasi emas.

Mamlakatimiz iqtisodiyotiga, birinchi navbatda, bizning sharoitimizga toʻgʻri keladigan, jarayonlar samaradorligining keskin оʻsis hi ga imkon beruvchi logistika falsafasini tadbiq etish lozim. Yangi ilmiy va oʻquv «logistika» fanini oʻrganish obyekti moddiy va ular bilan bogʻliq boʻlgan axborot oqimlari hisoblanadi. Fanning dolzarbligi va uni oʻrganishga boʻlgan qiziqishning keskin ortishi logistik yondoshuvdan foydalanish orqali moddiy tizimlar faoliyati samaradorligini oshirishning potensial imkoniyatlari bilan bogiiqdir. Logistika xomashyo va yarimtayyor mahsulotlar olib kelish bilan isteʻmolchiga tayyor mahsulot yetkazib berish oʻrtasidagi vaqt oraligʻini keskin qisqartirish imkonini beradi.

Yuklarni saqlash va tashish xarajatlarini keskin qisqartirishga imkoniyat yaratadi. Logistikaning qoʻllanishi m aʻlumot olish jarayonini tezlashtiradi, serviz xizmati darajasini oshiradi. Logistika sohasidagi faoliyat koʻpqirralidir. U transport, ombor xoʻjaligi, zaxiralar, xodimlar boshqaruvi, axborot tizimlari, tijorat faoliyati va boshqalarni tashkil etishni qamrab oladi. Sanalgan vazifalarning har biri tegislili soha fanida chuqur oʻrganilgan va bayon etilgan. Logistik yondashuvning vangiligi yuqorida sanalgan faoliyat sohalarining maromli tashkil etilgan yengil boshqaruvini va yuqori samarali material oʻtkazuvchi tizimlarini yaratish maqsadidagi uzviy bogʻliqligini taʻminlash hisoblanadi.

Logistik tarzda tashkil etilgan tizimlar anʻanaviy material oʻtkazuvchi tizimlardan xuddi zamonaviy avtomobil XX asr boshlaridagi avtomobildan farq qilgani kabi farq qiladi. Logistikaning nazariy fan sifatidagi maqsadi xuddi shunday tizimlarning paydo hoʻiishi va amal qilish qonuniyatlarini oʻrganishdir. Logistikaning amaliy faoliyat taridagi maqsadi esa ushbu tizimlarni yaratish va ularning amal qilishini taʻminlash hisoblanadi. Moddiy oqimlarni boshqarish har doim xoialik faoliyatining muhim tomoni hisoblanib kelingan, biroq nisbatan yaqin vaqtlarda u iqtisodiy faoiiyatning eng muhim vazifalaridan biri tarzidagi mavqega ega boldi. Asosiy sabab -isteʻmolchining tez oʻzgaruvchi hoxish-istaklariga tez moslashuvchan boʻlgan ishlab chiqarish va savdo tizimlari zamriyatini tugʻdirgan sotuvchi bozoridan isteʻmol bozoriga oʻtishidir.

Iqtisodiyotda tarkibiy islohotlar va iqtisodiyotni modernizatsiyalashni yanada chuqurlashtirish, uning koʻlamini kengaytirish bilan bogliq masalalarning nazariy va amaliy tomonlarini oʻrganishda «Logistika» fani alohida oʻrin tutadi. Jahon tajribasi shuni koʻrsatadiki, bugungi kunda raqobat bozorida kimki logistika sohasida layoqatli (kompetentli) boʻIsa, yetakchilik mavqeyiga ega boʻladi. Qʻquv qoVllanmani tayyorlashda mualliflar koʻp yil davomida olib borishgan izlanishlar natijasi, osqigan maʻruzalari va oʻquv adabivotlardan keng foydalangallar.

Logistika murakkab ishlab chiqarish-sotish komplekslarini oʻrganadi, bu komplekslar taʻminot, asosiy ishlab chiqarish, transport va sotuv sohalarida tashkiliy-iqtisodiy faoliyatni amalga oshiradi. Bu komplekslarning oʻziga xos xususiyatlari quyidagilardan iborat: asosiy vositalar va ishchi guruhlarning katta maydonga tarqalganligi; texnik vositalar asosiy qismining yuqori fond sigʻimi; logistik tizim elementlari baʻzilarining mobilligi va boshqalarining maʻlum jugʻrofik mintaqaga qattiq bogliqligi; katta miqdordagi aralash tizimostilar (yuk joʻnatuvchilar, yuk qabul qiluvchilar, resurslarni yetkazib beruvchilar va hokazo) ishining natijasiga bogliq. Koʻrib chiqilgan xususiyatlar logistik tizimda moddiy va axborot. oqimlarining oʻtishiga va hajmiga taʻsir qiladi hamda material oʻtkazuvchi zanjiming alohida boʻgʻinlari oʻrtasida yuzaga keladigan taslikiliy va iqtisodiy munosabatlar tabiatini aniqlaydi. Fan sifatida logistikaning predmeti boʻlib tovar harakati sohasida xarid, ishlab chiqarish va mahsulotni sotish bosqichlaridagi tashkiliy-iqtisodiy munosabatlar xizmat qiladi. Logistika fan sifatida logistik tizimlarni tashkil etishning umumiy xususiyatlari ni, qonun va qorruniyatlarmi oʻrganadi.

Logistikaning fan sifatida tuzilmasi boʻlib tovar harakati liiiayoniga tegishli boʻlgan qonuniyatlar va aloqalarni oʻrnatish hisoblanadi. Bu moddiy va axborot oqimlarini boshqarishning samarali shakl va uslublarini amaliyotda aniqlash va qoʻllash maqsadida amalga oshiriladi. Logistika fanining umumiy uslubi boMib tadqiqotning dialektik nsliibi hisoblanadi. Aniq masalalar yechimini topish uchun fanda ii/imli yondashuv, amallarni tadqiqot qilish usiublari va tizimlar na/ariyasidan foydalaniladi. Atama tarixi. Logistika tushunchasi oʻzinining tarixiga ega. Oadimgi greklar logistika deb hisob-kitoblarni amalga oshirish sanʻatini tushunishardi. Maxsus davlat nazoratchilarini logistlar deb atashardi. Qadimgi Rimda esa logistika deb mahsulotlar taqsimoti tushunilardi. Keyinchalik logistika atamasi harbiy qoʻshinlarning joyiashishi va harakatlanish amaliyotini tasniflashda ishlatila boshladi. I ramizning IX—X asrlarida yashagan Vizantiya imperatori Leon V, oʻzining harbiy ish boʻyicha kitobida «logistika» atamasini «front orti, qoʻshinlarning taʻminoti» maʻnosida ishlatadi.

Moddiy oqim. Moddiy oqim tushunchasi logistikada muhim oʻrin tutadi. Moddiy oqimlar tashish, omborga joylashtirish va xomashyo, yarimtayyor mahsulotlar hamda tayyor mahsulotlar bilan boshqa moddiy amallarni bajarish natijasida yuzaga keladi. Moddiy oqim - bu vaqt oraligʻiga keltirilgan va har xil logistik amaliarni joriy etish jarayonida коʻrib chiqilayotgan tovar — moddiy boyliklarning yigʻindisi. Ishlab chiqarishning aniq manbayidan to isteʻmolgacha boʻlgan oraliqda joylashgan bir turdagi resurslar yigʻindisi oddiy moddiy oqimni yuzaga keltiradi. Korxonada shakllanadigan bir nechta oddiy oqimlar korxonaning meʻyoriy faoliyatini taʻminlab bemvchi integrallashgan moddiy oqimni tashkil etadi. Moddiy oqimlarning harakatlanish sxemasi quyidagi rasmda keltirilgan.

Ombordagi barcha rejalashtirilgan operatsiyalar diqqat bilan oʻrganilishi va ish koʻrsatmalari shaklida tavsiflanishi kerak, bunda ombordagi tovarlar bilan barcha operatsiyalar hujjatlarda va axborot tizimida aks ettirilishi kerak. Ichki hujjatlarning standart (kompaniya uchun) shakllaridan foydalanish tavsiya etiladi, takroriy operatsiyalarni istisno qilish kerak; jarayon/operatsiya uchun masʻul xodimni tayinlash zarur. Boshqaruv qarorlarini qabul qilish, omborning ishlashini nazorat qilish uchun zarur boʻlgan maʻlumotlarni roʻyxatdan oʻtkazish va toʻplash kerak - omborni avtomatlashtirish tizimining funktsional imkoniyatlaridan maksimal darajada foydalanish kerak.

# 1.2-§ Veb ilovalarni ishlab chiqish muhitlari haqida

Internetning oʻzining taʻlimi esa 1980-yillarda boshlandi. 1989-yil Barmers-Li (Britaniya)da Tix Berners-Li, hypertext tarmoqlari asosida ishlovchi protokol HTTP va uning uchun xizmat koʻrsatadigan brauzer Mosaic tuzishadi. Bu hypertext tarmoqlari asosida ishlovchi protokol HTTP (Hyper Text Transfer Protocol) va brauzerlar, internetdagi eng mashhur tizimlardan biri hisoblanadi.

Veb saytlari yaratishining boshlangʻichi esa 1990-yillarga toʻgʻri keladi. Bu yillarda, HTML (Hyper Text Markup Language) dasturiy tillari ishlatilishi boshlandi. Bu tillar saytlar uchun maʻlumotlar yaratishda va ularga koʻrinish berishda yordam berdi. Bu davrda Veb-saytlarining koʻrinishi juda oddiy edi va ularning koʻrinishi koʻp farkli edi.

Bundan soʻng Veb-saytlarning yaratilishi va ularning koʻrinishi, tuzilishi va ishlash usullari kengaydi. Veb-saytlarining yaratilishida bir necha tillar, protokollar va vositalar ishlatildi. Bu davrda Veb-saytlarining koʻrinishi va dizayni ham koʻp oʻzgarib, ularning maqsadi va foydalanuvchilarga qulaylik koʻrsatishga qaratildi.

2000-yillarda, Veb 2.0 konseptsiya yuzaga kelishi bilan, Veb-saytlarining tuzilish va ishlash usullari koʻp oʻzgarishga uchradilar. Bu davrda, foydalanuvchilar saytlar ichida oʻzlarining maʻlumotlarini ham joylashtirishi mumkin boʻldi. JavaScript, AJAX va boshqa technologiyalar yordamida, Veb-saytlarining interaktivligi va dinamikligi oshirildi.

Veb ilovalarni yaratish uchun quyidagi texnologiyalardan foydalanish mumkin:

1. **HTML**: Asosiy veb dasturlash tili boʻlib, HTML (HyperText Markup Language) veb sahifalarni yaratish uchun ishlatiladi. Bu tilda veb sahifalarning strukturasi, matni, rasmlari, videolarni va boshqa tarkibiy elementlarni koʻrsatiladi.
2. **CSS**: CSS (Cascading Style Sheets) veb sahifalarni koʻrinishini oʻzgartirish uchun ishlatiladi. Bu tilda veb sahifalarga rang, qatorlar, qoralama, oʻlchamlar va boshqa tarkibiy elementlarga koʻrinish berish mumkin.
3. **JavaScript**: JavaScript veb sahifalarni dinamik qilish uchun ishlatiladi. Bu tilda veb sahifalarni oʻzgartirish, animatsiyalar yaratish, foydalanuvchilar bilan interaktivlik yaratish va boshqa amallar uchun ishlatiladi.
4. **Angular**: Bu Google tomonidan yaratilgan va TypeScript asosida ishlaydigan bir veb ilovalarining platformasi hisoblanadi. Bu platforma, bir nechta interfeys elementlarini yaratishga yordam beradi va dinamik veb ilovalarni yaratish uchun imkoniyatlar taqdim etadi.
5. **React**: Bu Facebook tomonidan yaratilgan va JavaScript asosida ishlaydigan bir veb ilovalarining platformasi hisoblanadi. Bu platforma, veb ilovalarining qisqa vaqt ichida ishga tushirilishini taʻminlaydi va bir nechta elementlarini birlashtirish imkonini taqdim etadi.
6. **Vue**.js: Bu JavaScript asosida ishlaydigan veb ilovalarining platformasi hisoblanadi. Bu platforma, veb ilovalarini qisqa vaqt ichida ishga tushirishga yordam beradi va odatda, yagona sahifalarni yaratishni osonlashtiradi.
7. **Ruby**: Bu Ruby asosida ishlaydigan bir veb ilovalarining platformasi hisoblanadi. Bu platforma, veb ilovalarini tezkor yaratish va ishga tushirishni osonlashtiradi.
8. **Django**: Bu Python asosida ishlaydigan bir veb ilovalarining platformasi hisoblanadi. Bu platforma, veb ilovalarini yaratishni osonlashtiradi va ishga tushirishni tezkorlashtiradi.

Bu texnologiyalardan foydalanib, siz veb ilovalarini yaratish va ishga tushirishni osonlashtirishingiz mumkin. Har bir platforma oʻziga xos imkoniyatlar taqdim etadi va sizning loyihangizga qarab, siz oʻzgaruvchilarni tanlashda vaqt yoki pullarni tejam etishda qoʻngʻiroq qilishi mumkin.

Shu kungacha, Veb-saytlarining yaratilish usullari va vositalari koʻp oʻzgardi. Bugungi kunda, Veb-saytlarini yaratish uchun bir qancha CMS (Content Management System) va platformalar mavjud. Bu tizimlar sizning saytingizni oson va qulay yaratishga yordam beradi. Shu bilan birga, saytlarning dizayni va optimallashtirilishi, foydalanuvchilarga qulaylik va saytingizni qidiruv tizimlari uchun optimallashtirishda katta muhim ahamiyatga ega.

Veb-saytlarining yaratilish tarixi internetning oʻzining tarixi bilan bogʻliqdir. Ular, internetning boshlangʻich yillarida oddiy va qulay emas edi, lekin ularning yaratilish usullari va vositalari kengayib, oʻzgarib, yaxshi koʻrinish va foydalanuvchilarga qulaylik koʻrsatishda katta oʻzgarishlar qildi. Bugungi kunda, Veb-saytlarining yaratilishi uchun juda koʻp vositalar, tillar va CMS tizimlari mavjud. Saytlarning dizayni, foydalanishini oson va qulay qilish, maqsadini aniqlash, maʻlumotlar tuzilishi, SEO va test etish, narxlash va boshqa jarayonlar saytingizning yaxshi koʻrinishda ishlashini taʻminlaydi. Joriy vaqtning Veb-saytlarini yaratishda juda koʻp yirik kompaniyalar, tashkilotlar va xususiy shaxslar qatnashishadi. Shuningdek, Veb-saytlarining yaratilishi va ularga foydalanish korxonalarning yaxshi tashkil etilishiga va oʻzlarining brendlarini rivojlantirishiga yordam beradi.

Dasturchi veb-saytning Front-End qismida JavaScriptdan foydalanmasligi ham mumkin. Lekin bu holat deyarli kuzatilmaydi sababi JavaScript-siz veb-sahifa bamisoli “oʻlik” veb-saytga aylanadi. Shunig uchun JS(JavaScript qisqartmasi) dan voz kechish toʻgʻri hisoblanmaydi. CSS va HTML esa almashtirish mumkin boʻlmagan tillar hisoblanadi. Bu tillarsiz veb-sayt tayyorlab boʻlmaydi. Shunday boʻlsada soʻnggi yillarda rivojlanib borayotgan SCSS ni CSS oʻrniga ishlatish ham mumkin. Tillar, afzallik va kamchiliklari, qay biri yaxshiroqligi haqida kelasi maqolalarda yanada batafsil maʻlumot berib oʻtaman.

Hozir oʻqiganlaringizdan hulosa qilsangiz boʻladiki yuqoridagi tillardan boshqa tillarni oʻrganib veb-sayt tayyorlashning imkoni yoʻq. Shuning uchun kompaniyalar Front-End dasturlovchiga qoʻyadigan talablar orasida yuqoridagilarni mukammal bilish asosiysi hisoblanadi.

Siz biror loyiha ustida ishlayotganingizda sizda turli hil kutubxonalar, tillar, frameworklar bilan ishlashga ham ehtiyoj paydo boʻladi. Shuning uchun korxonalar Front-End dasturlovchi olayotganda baʻzi qoʻshimchalarni bilishni ham talab qilishadi. Lekin, umumiy olganda yuqoridagi 3 ta til bilan han Front-End dasturchisiz.

NestJS Freamworki haqida qisqacha yozib oʻtamiz. NestJS, MVC (Model-View-Controller) arxitekturasini ishlatadi va barcha logikani komponentlarga boʻlib, tizimni modullarga ajratadi. Buning natijasida, dasturlar modullar yordamida tizimga qoʻshiladi, va bu tizimli xususiyatni taʻminlaydi. NestJS dasturlarini yaratish oson va natijaviy testlarni yozish osondir. Testlar, dasturni toʻgʻri ishlashini tekshirish uchun muhimdir va NestJS bu jarayonni qulaylashtiradi. NestJS tizim xususiyatlarini oʻz ichiga oladi va ularni tizimning har xil qismlarida ishlatish mumkin. Bu xususiyatlar, tizimning har xil qismlari uchun ishlatiladigan toʻplamlar, loggerlar, interseptorlar, middleware va boshqa xususiyatlar hisoblanadi. NestJS yordamida RESTful API yaratish oson boʻladi va bu API lar, hatto katta tizimlarda ham qoʻllaniladi. NestJS yordamida VebSocket imkoniyatlari boʻlib, real-time kommunikatsiyani osonlashtiradi. NestJS kengaytirish imkoniyatlarini taqdim etadi va koʻp qoʻllaniladigan tizimlarda ham foydalaniladi. NestJS TypeScript tilini ishlatadi va bu til, kodning xatolarni aniqlashni osonlashtiradi va dasturlarni yozish va oʻqishni osonlashtiradi.

Avvalo “frontend” va “backend” soʻzlarining oʻzbek tilidagi maʻnolari bilan tanishib chiqaylik. “Frontend” (front end, front-end) oʻzbek tilida “Oldindagi (ustki) tomon” degan maʻnoni anglatadi. Biz bu soʻz bilan u yoki bu narsaning tashqi, hammaga koʻrinib turadigan qismini nazarda tutamiz. Backend (back end, back-end) esa Frontendning aksi hisoblanib, qaysidir narsaning ichki, yoki orqa tomoni degan maʻnoni anglatadi. Bu bilan biz qaysidir narsaning hammaga koʻrinmaydigan, yashiring, ichki yoki orqa tomonini nazarda tutgan boʻlamiz.

Bugungi kunda shaxsiy veb sahifaga ega boʻlish oʻziga xos elektron vizitka, passport vazifasini bajaribgina qolmay, balki sayt orqali koʻplab foydali ishlarni amalga oshirish mumkin boʻlib qoldi. Masalan jurnalist, shoir, yozuvchi oʻz ijodi va izlanishlarinini veb sahifasi orqali yoritib borishi mumkin. Tadbirkor oʻz sahifasida mahsulot va xizmatlarini keng targʻib qilishi va daromadini oshirishi mumkin.

# 1.3-§ Masalaning qoʻyilishi

Buyurtmalarni shakillantirish va yetkazib berilishini nazorat qilish axborot tizimini ishlab chiqish shu sohada ish yuritayotgan birqancha firmalarga kata yordam bergan boʻladi. Xususan bu saytda reklama qismi, mijoz qismi, eltuvchi qismi va admin qismlari mavjud boʻlib ular ishlatishga qulay va funksionaldir.

Boshqa veb-saytlardek oʻxshamagan holda sayt koʻrinishi ixcham va foydalanuvchiga qulaylik yaratishi, veb-saytdan foydalanganda qiyinchiliklarga uchramasliklari muhim. Shuning uchun veb-saytni yaratish davomida juda koʻp boʻlmagan matnlardan foydalanish va koʻproq grafik maʻlumotlar qoʻshish maqsadga muvofiq boʻladi.

Veb-ilovada faqatgina maʻlumotlarni kiritish chiqarish va ularni boʻlimlarga boʻlishdan tashqari, ularning birliklari ham bor. Shuningdek undagi limitlar orqali tovarlarning qoldiqlari va qanday holatda(qizil, sariq, yashil) ekanligi tekshirib boriladi.

Veb-saytni yaratish davomida omborxona haqida koʻplab maʻlumotlar toʻplandi va aniq maʻlumotlarga asoslanib veb-saytga maʻlumotlar kiritildi. Veb-saytni yaratish uchun quyidagi qadamlar qoʻyildi:

* Shu kungacha yuklarni eltish sohasida qanday texnologiyalar ishlatilganligi oʻrganish va tadbiq qilish;
* Frontend qismida sahifalarni yaratish va CSS yordamida dizaynini yaratish;
* Veb-ilovaning backend qismini yozish, maʻlumotlar bazasi bilan ulash va frontend qismiga maʻlumotlarni uzatish;
* Frontend qismida maʻlumotlarni qabul qilish kodlarini yozish va maʻlumotlarni sahifalarga chiqarish;
* Veb-ilova toʻgʻri ishlayotganini tekshirish va uni hostingga joylash;

- Ilovaga boshlangʻich maʻlumotlarni kirgizish orqali uni tahlil qilish va foydalanuvchiga topshirish;

# II-BOB. ASOSIY QISM.

# 2.1-§ Axborot tizimini ishlab chiqishda tanlangan texnologiyalar

Foydalanuvchi interfeysi yaʻni veb-saytning server tomonini ishlab chiqish texnik tajriba, ijodkorlik va tafsilotlarga eʻtiborni talab qiladi. Eng yaxshi amaliyotlar va sanoat standartlariga rioya qilish va toʻgʻri vositalar va usullardan foydalanish orqali siz xavfsiz, kengaytiriladigan va samarali boʻlgan server tomoni kod bazasini yaratishingiz mumkin.

Veb-Framework - bu veb-ilovalarni yaratish uchun vositalar, kutubxonalar va xizmatlar toʻplamini taʻminlovchi dasturiy platforma. Veb-frameworklar ishlab chiquvchilarga standartlashtirilgan arxitektura va kodni tashkil qilish va tizimlashtirish uchun konventsiyalar toʻplamini taqdim etish orqali veb-ilovalarni yaratishni osonlashtiradi. Bu ishlab chiquvchilarga vaqtni tejash, xatolarni kamaytirish va samaradorlikni oshirishga yordam beradi. Ular, veb ilovalarning asosiy komponentlarini, protokollarni va kod tashkilini taʻminlaydi. Quyidagi muhim faktorlarni hisobga olgan holda, veb frameworklar veb ilovalar yaratishdagi oʻrinini taʻriflash mumkin:

* Qayta ishlatish imkoniyati: Veb frameworklar, asosiy funktsiyalarni (masalan, yoʻnlash, maʻlumotlar bazasiga kirish, sessiyalar boshqarish) oʻz ichiga olgan tayyor komponentlar va kutubxonalarni taʻminlaydi. Bu dasturchilarning bu komponentlarni qayta ishlatib, veb ilovalarni tezroq va samarali tarzda yaratishlarini taʻminlaydi.
* MVC (Model-View-Controller) arxitekturasi: Koʻpchilik veb frameworklar MVC arxitekturasiga asoslanadi. Bu arxitektura ilovalarning maʻlumot modelini (Model), foydalanuvchi interfeysini (View) va ishlov boshqaruvini (Controller) bir-biridan alohida tartibda tashkil etishga imkon beradi. Bu kodni yuqori tartibda saqlash va saqlashni osonlashtiradi.
* Xavfsizlik: Veb frameworklar xavfsizlik choralarni osonlashtiradi. Masalan, foydalanuvchi tasdiqlash, ruxsat berish va xavfsiz maʻlumotlar bazasi amallarini oʻz ichiga oladigan komponentlarni taʻminlaydi. Bu veb ilovalarning xavfsizligini oshiradi va xavfsizlik xatorlarini kamaytiradi.
* Maʻlumotlar bazasiga integratsiya: Veb frameworklar maʻlumotlar bazalariga kirishni osonlashtiradi. Ular odatda ORM (Object-Relational Mapping) vositalari bilan integratsiya qilinadi, bu esa maʻlumotlar bazasi amallarini obyektoriyentlangan tarzda bajarishga imkon beradi. Bu maʻlumotlar bazasi operatsiyalarini oson, tez va xavfsiz tarzda bajarishga yordam beradi.
* URL yoʻnlash: Veb frameworklar URL yoʻnalashish uchun oson va moslashtirilgan struktura taʻminlaydi. Bu mijozlarni toʻgʻri sahifalarga yoʻnaltirish, dynamik URLʻlar boshqarish va sahifalar orqali yoʻnaltirishni osonlashtirishga imkon beradi.
* Shablon oʻrnaklari: Veb frameworklar, sahifalarni dynamik tuzish uchun shablon oʻrnaklari taʻminlaydi. Bu server tomonidagi HTML tarkibini dynamik tarzda yaratish, maʻlumotlarni shablon ichiga joylash va foydalanuvchi interfeysini boshqarishga imkon beradi.

Veb frameworklar, veb ilovalarni oson va samarali tarzda ishlab chiqishda katta ahamiyatga ega boʻladi va dasturchilarga kuchli imkoniyatlar beradi.

Biz oʻz loyihamizni ishlashda Angular, NestJS va MySQL texnologiyalarinda foydalandik va bular haqida qisqacha maʻlumot berib oʻtaman.

Angular veb ilovalarni ishlab chiqishda foydalaniladigan katta va murakkab veb frameworkidir. Angular, bir qancha kuchli imkoniyatlarni va vositalarni taʻminlaydi. Ushbu framework komponent asosli arxitekturani, maʻlumotlarni bogʻlashni, yoʻnlarni boshqarishni, formalar bilan ishlashni, animatsiyalar va boshqalarini oʻz ichiga oladi. Angular, dasturchilarga ish jarayonini tezlashtirish va ish samaradorligini oshirish uchun bir qator vositalar taʻminlaydi. Angular CLI (Command Line Interface) misol uchun loyihani yaratish, komponent yaratish, testni bajarish va boshqalarini avtomatlashtiradi. Angular, maʻlumot bogʻlash imkoniyatlari bilan tanishadi. Ikki tomonlama maʻlumot bogʻlash, komponentlar va shablonlar orasida maʻlumotlar sinxronizatsiyasini osonlashtiradi. Bu maʻlumot oʻzgarishlarining tezroq koʻrsatilishini va foydalanuvchi interfeysining yangilanishini taʻminlaydi.

Angular modulli tuzilishga ega. Komponentlar, xizmatlar, direktivlar va boshqa qurilmalar modulyar tarzda yaratilishi mumkin va qayta ishlatishga moʻljallangan. Bu kodni tizilgan va boshqarilishga imkon beradi. Shuningdek, Angular kengaytirish imkonini beruvchi strukturasiga ega, bu esa ilovani maxsus talablarga moslashtirishni osonlashtiradi. Angular katta va faol jamoaga ega. Ushbu jamoa, resurslar, misollar, kutubxonalarga va yoʻriqnoma taqdim etish bilan yordam berishi mumkin. Shuningdek, mavjud boʻlgan rasmiy Angular hujjatlari ham doimiy ravishda yangilanib turadi.

Angular katta va oʻlchamli veb ilovalarni ishlab chiqishni isteyen dasturchilar uchun ideal bir tanlovdir. Lekin kichik oʻlchamdagi yoki oddiy veb loyihalar uchun yengil va oson frameworklarni tanlash mumkin. Angular murakkab va katta oʻlchamdagi Veb ilovalarni ishlab chiqishda foydalaniladi. Ushbu framework, komponent asosli arxitektura, modullar, xizmatlar va qoʻshimcha vositalarni taʻminlaydi, shuningdek, ikki tomonlama maʻlumot bogʻlash, maʻlumotlar bazasiga integratsiya, animatsiyalar va koʻp qanchalik boshqa imkoniyatlarga ega.

Angular juda katta interfeyslar yaratish uchun qulay imkoniyatlarni taʻminlaydi. Bunda komponentlar va shablonlar oʻrtasida maʻlumotlar sinxronizatsiyasi va avtomatik yangilanish mavjud. Bu foydalanuvchining veb ilovaga kirish bilan bogʻliq maʻlumotlarni toʻgʻri sahifalarga yoʻnaltirishni osonlashtiradi. Angular modullar yordamida ilovani modulyar tuzishga imkon beradi. Bu kodni qulay boshqarish va qayta ishlatishni osonlashtiradi. Modullar bir nechta funksiyonalni bir araya kelishi mumkin va ilovani tuzilishini aniq va qisqa boʻladi.

Angular test qilish va xato tanqisligi uchun imkoniyatlarni taʻminlaydi. Bunda tashkil qilinadigan testlar yordamida ilovadagi funktsiyalar toʻgʻri ishlayotganligini tekshirish mumkin. Angular toʻgʻridan-toʻgʻri kengaytirish va oʻzgartirishlarni qabul qilish imkoniyatini taʻminlaydi. Bu ilovada yangi talablar va funksiyalarni qoʻshishni osonlashtiradi.

Veb saytning server tomonini ishlab chiqishda eʻtiborga olish kerak boʻlgan baʻzi qadamlar:

1. Dasturlash tili va frameworkni tanlang: Dasturlash tili va veb-frameworkni tanlash veb saytingizning oʻziga xos ehtiyojlariga bogʻliq. Masalan, Ruby on Rails maʻlumotlar bazasi backendiga ega dinamik veb saytlarni yaratish uchun mashhur veb-frameworkdir, Node.js esa real vaqt rejimida, voqealarga asoslangan veb-ilovalarni yaratish uchun mashhur tanlovdir. Siz tanlagan framework marshrutlash, oʻrta dastur va maʻlumotlar bazasi integratsiyasi kabi xususiyatlarni taʻminlashi kerak.
2. Maʻlumotlar bazasi sxemasini loyihalash: Maʻlumotlar bazasi sxemasi veb sayt maʻlumotlari qanday saqlanishining rejasidir. Sxemani diqqat bilan loyihalash, uni masshtablash, samarali va oson saqlashga ishonch hosil qilish muhimdir. Bu jadvallar, ustunlar va jadvallar orasidagi munosabatlarni aniqlash va asosiy kalitlar, tashqi kalitlar va indekslar kabi cheklovlarni oʻrnatishni oʻz ichiga oladi.
3. Server tomoni kodini ishlab chiqish: Server tomonidagi kod veb saytning mijoz tomonidagi soʻrovlarni koʻrib chiqish, maʻlumotlarni qayta ishlash va javoblarni yaratish uchun javobgardir. Bunga API yaratish, EJS va Rulda kabi mashhur shablon dvigatellaridan foydalangan holda server tomoni shablonlarini yaratish va looplar, shartlar va xatolarni qayta ishlash kabi dasturlash konstruksiyalaridan foydalangan holda biznes mantiqini amalga oshirish kiradi.
4. Xavfsizlik choralarini amalga oshirish: Veb saytning server tomonini ishlab chiqishda xavfsizlik birinchi oʻrinda turishi kerak. Bunga kirishni tekshirish, autentifikatsiya va avtorizatsiya kabi xavfsizlik choralarini amalga oshirish kiradi. Kirish tekshiruvi foydalanuvchi kiritgan maʻlumotlar xavfsiz va kutilgan formatlarga mos kelishini taʻminlaydi, autentifikatsiya va avtorizatsiya esa faqat avtorizatsiya qilingan foydalanuvchilar veb saytning ayrim qismlariga kirishi yoki muayyan harakatlarni amalga oshirishi mumkinligini taʻminlaydi. Boshqa xavfsizlik choralari orasida HTTPS dan foydalanish, xizmat koʻrsatishni rad etish hujumlarining oldini olish uchun tezlikni cheklashni amalga oshirish va maxfiy maʻlumotlarni shifrlash kiradi.
5. Ishlash uchun optimallashtirish: Ishlash veb saytning server tomoni uchun juda muhim, chunki u foydalanuvchi tajribasiga bevosita taʻsir qiladi. Maʻlumotlar bazasi soʻrovlarini minimallashtirish, tez-tez foydalaniladigan maʻlumotlarni keshlash va kerak boʻlganda server tomonidagi renderlash orqali ishlash uchun server tomoni kodini optimallashtirish muhimdir. Ishlashni optimallashtirishning boshqa usullari maʻlumotlarni siqish, statik aktivlarga xizmat koʻrsatish uchun CDN-lardan foydalanish va maʻlumotlar bazasi soʻrovlarini optimallashtirishni oʻz ichiga olishi mumkin.
6. Sinov va disk raskadrovka: Sinov server tomonini ishlab chiqishning muhim qismidir, chunki u kodning moʻljallanganidek ishlashini va hech qanday xato yoki xavfsizlik zaifliklari yoʻqligini taʻminlaydi. Sinov birlik testini, integratsiya testini va oxirigacha sinovni oʻz ichiga olishi mumkin va Jest, Mocha va Selenium kabi vositalar yordamida amalga oshirilishi mumkin. Xatolar yoki muammolar aniqlanganda, ularni disk raskadrovka vositalari va usullaridan foydalangan holda tizimli ravishda tuzatish muhimdir.
7. Ishlab chiqarishga joylashtirish: Server tomonidagi kodni ishlab chiqarish muhitiga joylashtirish puxta rejalashtirish va bajarishni talab qiladi. Bu Git yoki Docker kabi vositalar yordamida bulutli serverni oʻrnatish, server muhitini sozlash va kodni joylashtirishni oʻz ichiga olishi mumkin. Ishlab chiqarish muhiti unumdorlik, kengayish va xavfsizlik uchun optimallashtirilganligini hamda joylashtirish jarayoni avtomatlashtirilgan va takrorlanishini taʻminlash muhim.
8. Monitoring va texnik xizmat koʻrsatish: Veb saytning server tomonini kuzatib borish uning muammosiz va samarali ishlashini taʻminlash uchun zarurdir. Bu real vaqt rejimida muammolarni aniqlash va bartaraf etish uchun server jurnallari, ishlash koʻrsatkichlari va xatolik stavkalarini kuzatish hamda New Relic yoki Datadog kabi vositalardan foydalanishni oʻz ichiga olishi mumkin. Server tomoni kodiga muntazam texnik xizmat koʻrsatish bogʻliqliklarni yangilashni, xatolarni tuzatishni va veb sayt funksionalligi va foydalanuvchi tajribasini yaxshilash uchun yangi xususiyatlarni joriy qilishni oʻz ichiga oladi.

Umuman olganda veb saytning server tomonini ishlab chiqish texnik tajriba, ijodkorlik va tafsilotlarga eʻtiborni talab qiladi. Eng yaxshi amaliyotlar va sanoat standartlariga rioya qilish va toʻgʻri vositalar va usullardan foydalanish orqali siz xavfsiz, kengaytiriladigan va samarali boʻlgan server tomoni kod bazasini yaratishingiz mumkin.

Veb-framework - bu veb-ilovalarni yaratish uchun vositalar, kutubxonalar va xizmatlar toʻplamini taʻminlovchi dasturiy platforma. Veb-frameworklar ishlab chiquvchilarga standartlashtirilgan arxitektura va kodni tashkil qilish va tizimlashtirish uchun konventsiyalar toʻplamini taqdim etish orqali veb-ilovalarni yaratishni osonlashtiradi. Bu ishlab chiquvchilarga vaqtni tejash, xatolarni kamaytirish va samaradorlikni oshirishga yordam beradi.

Veb-frameworklar odatda quyidagi xususiyatlar va funksiyalarni taqdim etadi:

1. Marshrutlash: Veb-ramkalar URL-manzillarni dasturning muayyan qismlariga, masalan, kontrollerlar yoki koʻrinishlarga solishtirish uchun marshrutlash mexanizmlarini taqdim etadi. Bu ishlab chiquvchilarga oʻz ilovalari uchun toza va tartibli URL manzillarini yaratishga imkon beradi.
2. Shablonlashtirish: Veb-ramkalar ilova maʻlumotlari asosida dinamik ravishda HTML kodini yaratish uchun shablonlash mexanizmlarini taqdim etadi. Shablonlash mexanizmlari ishlab chiquvchilarga taqdimot qatlamini biznes mantiqiy qatlamidan ajratish imkonini beradi.
3. Maʻlumotlar bazasi integratsiyasi: Veb-ramkalar obʻektga aloqador xaritalash (ORM) yoki maʻlumotlarga kirish qatlamlari kabi maʻlumotlar bazalari bilan integratsiya qilish mexanizmlarini taqdim etadi. Bu ishlab chiquvchilarga ilovadagi maʻlumotlarni saqlash, olish va boshqarishni osonlashtiradi.
4. Xavfsizlik: Veb-ramkalar kirishni tekshirish, autentifikatsiya va avtorizatsiya kabi xavfsizlik xususiyatlarini taʻminlaydi. Bu dasturni SQL inʻektsiya hujumlari yoki saytlararo skript (XSS) hujumlari kabi xavfsizlik zaifliklaridan himoya qilishga yordam beradi.
5. Sinov: Veb-ramkalar dastur kodini sinab koʻrish uchun test tizimlarini taqdim etadi, jumladan birlik testi, integratsiya testi va funktsional test. Bu ilovaning moʻljallanganidek ishlashini va hech qanday xato yoki muammo yoʻqligini taʻminlashga yordam beradi.
6. Masshtablilik: Veb-ramkalar dasturni masshtablash mexanizmlarini taqdim etadi, masalan yuklarni muvozanatlash, keshlash va klasterlash. Bu ilovaga oʻsib borayotgan trafik miqdorini boshqarish imkonini beradi.

Koʻplab veb-frameworklar mavjud, ularning har biri oʻzining kuchli va zaif tomonlariga ega. Baʻzi mashhur veb-ramkalarga quyidagilar kiradi:

1. Ruby on Rails veb-ramka boʻlib, konfiguratsiyadan koʻra konventsiyani taʻkidlaydi. U maʻlumotlar bazasiga asoslangan veb-ilovalarni yaratish uchun bir qator vositalar va konventsiyalarni taqdim etadi, bu esa uni startaplar va kichik biznes uchun mashhur tanlovga aylantiradi.
2. Django yuqori unumli veb-ilovalarni yaratish uchun veb-ramkadir. U soddaligi, moslashuvchanligi va foydalanish qulayligini taʻkidlaydi va umumiy veb-ishlab chiqish vazifalari uchun bir qator oʻrnatilgan xususiyatlarni taqdim etadi.
3. Laravel PHP veb-ramka boʻlib, u kengaytiriladigan va xizmat koʻrsatish mumkin boʻlgan veb-ilovalarni yaratish uchun vositalar va konvensiyalar toʻplamini taqdim etadi. U kodning oʻqilishi va barqarorligini taʻkidlaydi va sinov, autentifikatsiya va maʻlumotlar bazasi integratsiyasi uchun bir qator xususiyatlarni taqdim etadi.

U qayta foydalanish mumkin boʻlgan komponentlarni yaratish uchun asboblar va kutubxonalar toʻplamini taqdim etadi, bu esa uni kengaytiriladigan va xizmat koʻrsatish mumkin boʻlgan veb-ilovalarni yaratish uchun mashhur tanlovga aylantiradi.

Umuman olganda, veb-ramkalar ishlab chiquvchilarga standartlashtirilgan arxitektura va kodni tashkil qilish va tizimlashtirish uchun konventsiyalar toʻplamini taqdim etish orqali veb-ilovalarni yanada samarali va samarali yaratishga yordam beradi. Loyiha uchun toʻgʻri veb-ramkani tanlash va uning xususiyatlari va funktsiyalaridan foydalanish orqali ishlab chiquvchilar oʻz foydalanuvchilari va manfaatdor tomonlarning ehtiyojlarini qondiradigan yuqori sifatli, kengaytiriladigan va texnik xizmat koʻrsatadigan veb-ilovalarni yaratishi mumkin.

Biz loyihamizda Angular JS dan foydalanganmiz, shuning uchun shu haqida toʻliq maʻlumot berib oʻtamiz.

Biz ilovamizning server qismini NestJS frameworkidan foydalanib amalga oshirdik. NestJS Node.js uchun TypeScript asosli bir server-side frameworkidir.

NestJS ilovani modullar boʻyicha boʻlimlarga ajratish imkoniyatini taʻminlaydi. Bu ilovadagi kodni tuzilgan va tuzilmagan qismga boʻlib taqsimlash va qayta ishlatishni osonlashtiradi. Modullar baʻzi funksiyonalni bir araya kelish uchun yoki kodni katta oʻlchamdagi ilovalarni qisqartirish uchun ishlatiladi. NestJS dependance injection prinsiplariga asoslangan bir arxitektura bilan ishlaydi. Bu ilovada foydalanuvchi oʻzgartirishlari bilan shu oʻzgartirishlarga boʻlgan baʻzi qismni avtomatik ravishda birlashtirishga imkon beradi. Bağımlilik enjeksiyoni, ilovadagi obyektlar yaratilganda va ularga qoʻshimcha tahrirlar qilganda yordam beradi.

NestJS HTTP endpointlarni yaratish va boshqarish uchun qulay imkoniyatlarni taʻminlaydi. Bu RESTful API lar yaratish uchun yaxshi ishlaydi. Endpointlar HTTP metodi va URL boʻyicha tuziladi va ularga tahlili middleware va metodsizlar bilan toʻlov amalga oshirish mumkin. NestJS middleware lar yordamida HTTP soʻrovlarni ichki qismiga oʻtkazishni yoki javobni oʻzgartirishni taʻminlaydi. Middleware lar, soʻrovlarni ushbu qismiga yetkazish tartibini belgilashga imkon beradi. Bu bloglash, avtorizatsiya, keshlash va boshqalar kabi bir nechta amallarni bajarish uchun foydalaniladi.

NestJS interseptorlar yordamida HTTP soʻrovlarni boshqarish jarayonini oʻzgartirishga imkon beradi. Interseptorlar, soʻrovni oldin va keyin ushlab turish, soʻrovni rad etish, javobni oʻzgartirish va boshqalar kabi amallarni bajarish uchun ishlatiladi. NestJS yuqori darajadagi tuzilish, kod qayta ishlash imkoniyatlari va bogʻliqliklar orasida oʻzaro aloqalarni osonlashtirish uchun moʻljallangan. Bu framework, server-side ilovalarni yaratishda yaxshi ishlaydi va korporativ tizimlarni qurish uchun muhim variantdir.

NestJS TypeScriptʻi qoʻllab-quvvatlashi va koʻp modulli arxitekturasini oʻz ichiga olganligi bilan tanlanib qolgan. TypeScript, JavaScript tilining bir uzluksizlashmasidir va statik tip berish imkonini qoʻllab-quvvatlayadi. Bu esa kodni aniq va xato yuzaga kelmagan qiladi va uni oʻqish va tahlil qilishning muhimi qismini avtomatiklashtiradi. NestJS, Express.js ilovasining foydali imkoniyatlarini qoʻllab-quvvatlashi, bağımlılık enjeksiyonu asosida modullar yaratishni osonlashtirishi va koʻp qismlardan iborat ilovalarni osonlashtirishi bilan tanlanib qolgan. Bu server-side ilovalarni qurishda modular, toza va toʻgʻridan-toʻgʻri yuritish va yaxshi test qilish imkonini taʻminlayadi.

Biz ilovamizni yaratishda maʻlumotlar tizimiga uzviy boʻgliqligimiz uchun maʻlumotlar tizimi bilan ishlovchi MySQL texnologiyasini ham qoʻshdik.

MySQL bu - maʻlumotlar ombori boshqarish tizimi (DBMS) sifatida ishlaydigan bepul va ochiq manbali (open-source) maʻlumotlar omboridir. Ushbu tizim relational (munosabatli) maʻlumotlar ombori sifatida ishlaydi va bir nechta platformalarda oʻrnatilishi mumkin. Ushbu tizimning bosh vazifasi maʻlumotlarni saqlash, ularga murojaat qilish va ularga toʻgʻridan-toʻgʻri soʻrovlar bilan ishlashni taminlashdir. MySQL, ommaviy va korporativ sohalar uchun qoʻllaniladi. Ushbu maʻlumotlar ombori ulkan maʻlumotlar kundaligi, foydalanuvchilar, mijozlar, mahsulotlar, buyurtmalar, hisob-kitob, xodimlar kabi maʻlumotlar toʻplamlarini saqlashga yordam beradi. MySQL, yuqori tezlik va ishlab chiqarish yetishmovchiliklari bilan hamda yuqori ishlaydiganlik va stabil ishlash imkoniyatlari bilan mashhurdir. Ushbu maʻlumotlar ombori, ichki va tashqi taʻminot, biznes-tizimlar va Veb ilovalarining backend qismida ommaviy tarqatuvlar kabi bir nechta sohada keng qoʻllaniladi. MySQL oʻzbek tilida ham foydalaniladi va ulugʻlanishga ega boʻlgan maʻlumotlar omborlaridan biridir.

MySQL maʻlumotlar ombori, bir nechta arxitekturalar yaratishga imkon beradi, bunday arxitekturalar quyidagilarni oʻz ichiga oladi:

* Standart (monolit) arxitektura: Bu arxitekturada barcha funksiyalarni bitta boshqaruvchi (monolit) tizim amalga oshiradi. Bunda maʻlumotlar omborini boshqarish, soʻrovlar va tashqi integratsiyalar barchasi bitta tizim ichida joylashgan.
* Boʻlimlangan arxitektura: Bu arxitekturada maʻlumotlar ombori va loyihalar bir nechta modullar (boʻlimlar)ga boʻlingan. Har bir boʻlim oʻzining maqsad va vazifalarga ega boʻladi. Ushbu arxitekturada modullar oʻzining oʻzgaruvchanliklari bilan faol ishlayadi va oʻzaro interfeyslar orqali aloqalar oʻrnatiladi.
* Maʻlumotlar ombori federatsiyasi: Bu arxitekturada bir nechta maʻlumotlar ombori oʻzaro aloqa orqali birlashadi va bitta boʻsh ombor kabi koʻrinadi. Har bir ombor oʻzining maʻlumotlarini saqlaydi va oʻzining ustida operatsiyalarni bajaradi. Bu usulda maʻlumotlar omborlari oʻzining isteʻmolchilari uchun birlikda ishlaydi.
* Sharding: Sharding arxitekturasi, maʻlumotlarni bir nechta shard (qism)ga boʻlishni taʻminlayadi. Har bir shard oʻzining maʻlumotlarni oʻz ichiga saqlaydi va soʻrovlarni oʻz omboriga yoʻnaltiradi. Ushbu usul yordamida maʻlumotlar omborining ishlab chiqarish va qidirish tezligi oshiriladi.
* Master-Slave replikatsiya: Bu arxitekturada asosiy maʻlumotlar ombori (Master) oʻzining oʻzgaruvchanliklarini koʻchirib turadi va undan bir yoki bir nechta Slave omborlariga maʻlumotlarni replikasiya qiladi. Replikasiya omborlarida oʻqish amalga oshirilishi mumkin, shuningdek, oʻzgartirishlarni koʻchirish imkoniyati ham mavjud.

Ushbu arxitekturalar MySQL maʻlumotlar omborini yaratishda va tashqi ilovalar bilan integratsiya qilishda foydalaniladigan turli usullardir. Har bir usul oʻzining afzalliklari va chegaralariga ega boʻlib, maqsad va talablarga qarab tanlanadi.

MySQL, ACID (Atomiklik, Konsistentlik, Ishtirokchilik, Dastlabki holat) tamoyillarini qoʻllab-quvvatlayan transaktsiya boshqaruvini taqdim etadi. Bu bir yoki bir nechta soʻrovlardan iborat ishlar toʻplamini bajarishda maʻlumotlar toʻplamining toʻliqlikni taʻminlash imkonini beradi. MySQL, indekslarni qoʻllab-quvvatlaydi, bu maʻlumotlar omboridagi maʻlumotlarga tez va samarali qidiruv imkonini beradi. Indekslar maʻlumotlar omborining ustida tartiblanadi va qidiruvni tezlashtiradi. MySQL, saqlangan usullarni oʻz ichiga oladi. Bu SQL soʻrovlarining qatorlarini yigʻish, server tomonidan saqlanish va kutilgan vaqtida ishlatilishi imkonini beradi. Saqlangan usullar kodni qayta qayta yozishdan qutqaradi va server tomonidan toʻgʻridan-toʻgʻri ishga tushiriladi.

MySQL maʻlumotlar omboridagi ulangan maʻlumotlarni uzluksizlikni taʻminlash uchun bajarilgan soʻrovlardan bir bir oʻqishning taʻqiqlanmagan holatini taqdim etadi. Bunda, boshqa soʻrovlarga tegishli oʻqishlar tartiblanmasdan amalga oshirilishi mumkin. MySQL birlashtirish (Join) operatsiyasini qoʻllab-quvvatlaydi, bu esa bir nechta jadvallarni bogʻlash orqali maʻlumotlarni birlashtirish imkonini beradi. Buning yanada samarali boʻlishi uchun birlashtirish indekslari yordamchi indekslarni oʻz ichiga oladi. MySQL maʻlumotlar omborining shakllalanishiga qoʻllanish imkonini beradi. Ushbu funksiya ombordagi maʻlumotlarning hajmini va qidiruvlarning tezligini oshirish imkonini beradi. Bu faqat bir necha kuchli funksiyalar bilan chegaralanmagan. MySQLning qoʻllab-quvvatlaydigan turli funksiyalari mavjud. Boshqa funksiyalar ham mavjud boʻlishi mumkin, shuningdek, yangilanib borayotgan MySQL versiyalarida yangi funksiyalar qoʻshish mumkin. Qoʻshimcha sifatida quyidagilarni keltiramiz:

* View (Koʻrinish): View, bir yoki bir nechta jadvallardan olingan maʻlumotlarni virtual koʻrinish sifatida saqlaydi. Buning bilan, maʻlumotlarni real jadvalidan ajratib koʻrish, maʻlumotlar ombori tarkibidagi maʻlumotlar bilan ishlash va oʻqish amalga oshirish imkoniyatini beradi. View, asl jadval maʻlumotlarida oʻzgarishlar boʻlsa ham oʻzini avtomatik ravishda yangilaydi.
* Trigger: Trigger, maʻlumotlar ombori ustida amalga oshirilgan har qanday amalga oshirishni avtomatik boshqarish imkonini beradi. Triggerlar, biror bir tadbirni belgilash orqali biror bir hodisani kuzatish, qoʻllanish, yangilash yoki oʻchirish jarayonlarida ishlatiladi. Misol uchun, bir jadvalga yangi qator qoʻshildigan boʻlsa, trigger avtomatik ravishda boshqa amallarni bajarishi mumkin.
* Stored Procedure (Saqlangan usul): Stored Procedure, SQL soʻrovlari va boshqarish qatorlarini birlashtirgan va server tomonidan saqlangan kod blokini ifodalaydi. Ushbu usullar maʻlumotlar omborining barcha tizimlarida saqlanadi va qayta-qayta ishlatilishi mumkin. Stored Procedurelar oʻzgaruvchilar, shartlar, oʻzgarishlar, toʻplamlar va boshqa logika bilan birgalikda ishlatilishi mumkin.

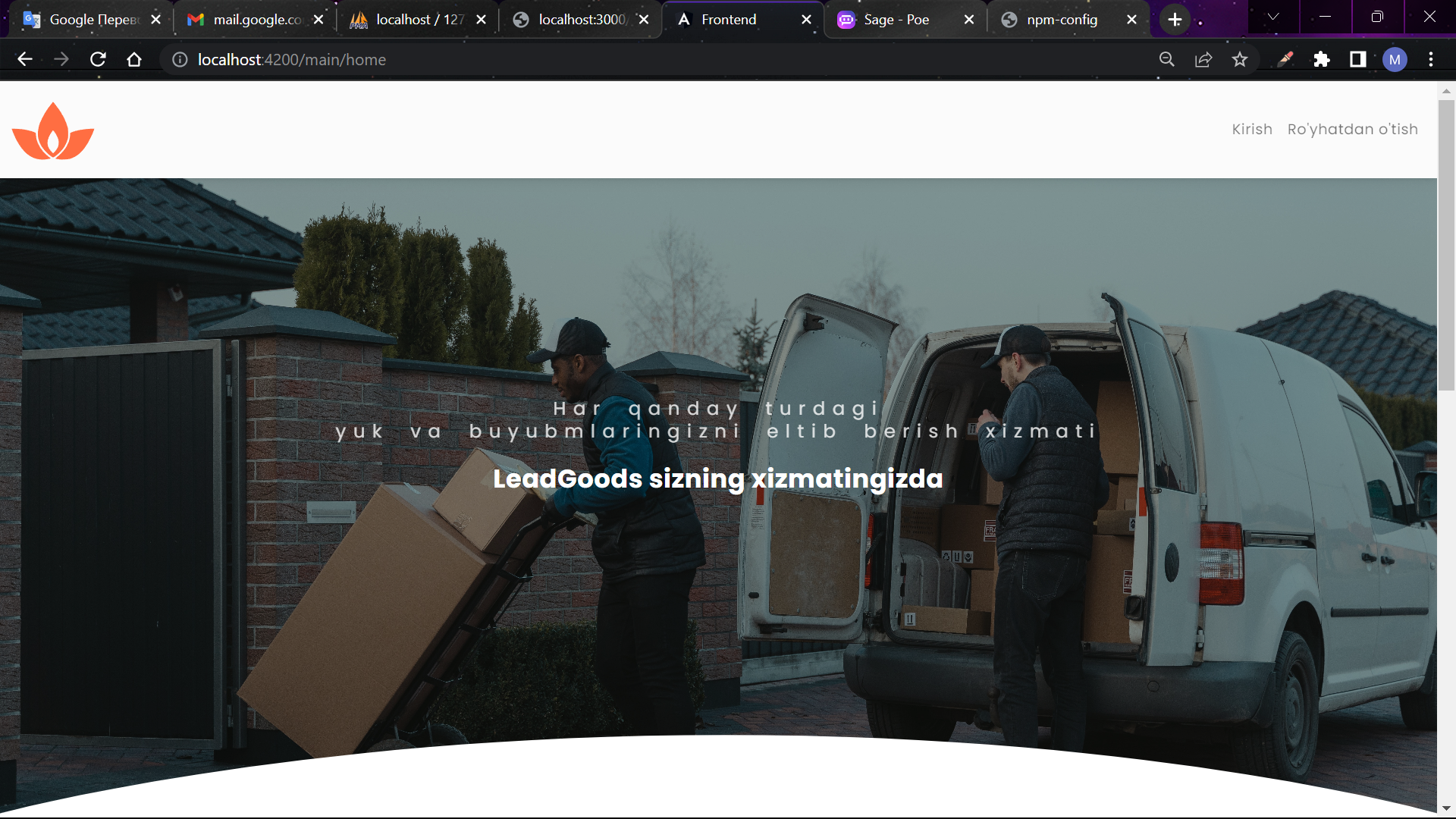
Bu funksiyalar MySQL da maʻlumotlar omborini boshqarish, maʻlumotlarni ajratib koʻrish, oʻzgarishlarni kuzatish va avtomatik operatsiyalarni bajarish imkonini beradi. Ularning qoʻllanishiga qarab, maʻlumotlar omborini yangilash, maʻlumotlarni birlashtirish, maʻlumotlarni kuzatish va ish jarayonlarini avtomatlashtirish oson va samarali boʻladi.

# 2.2-§ Dastur interfeyslarining tavsifi

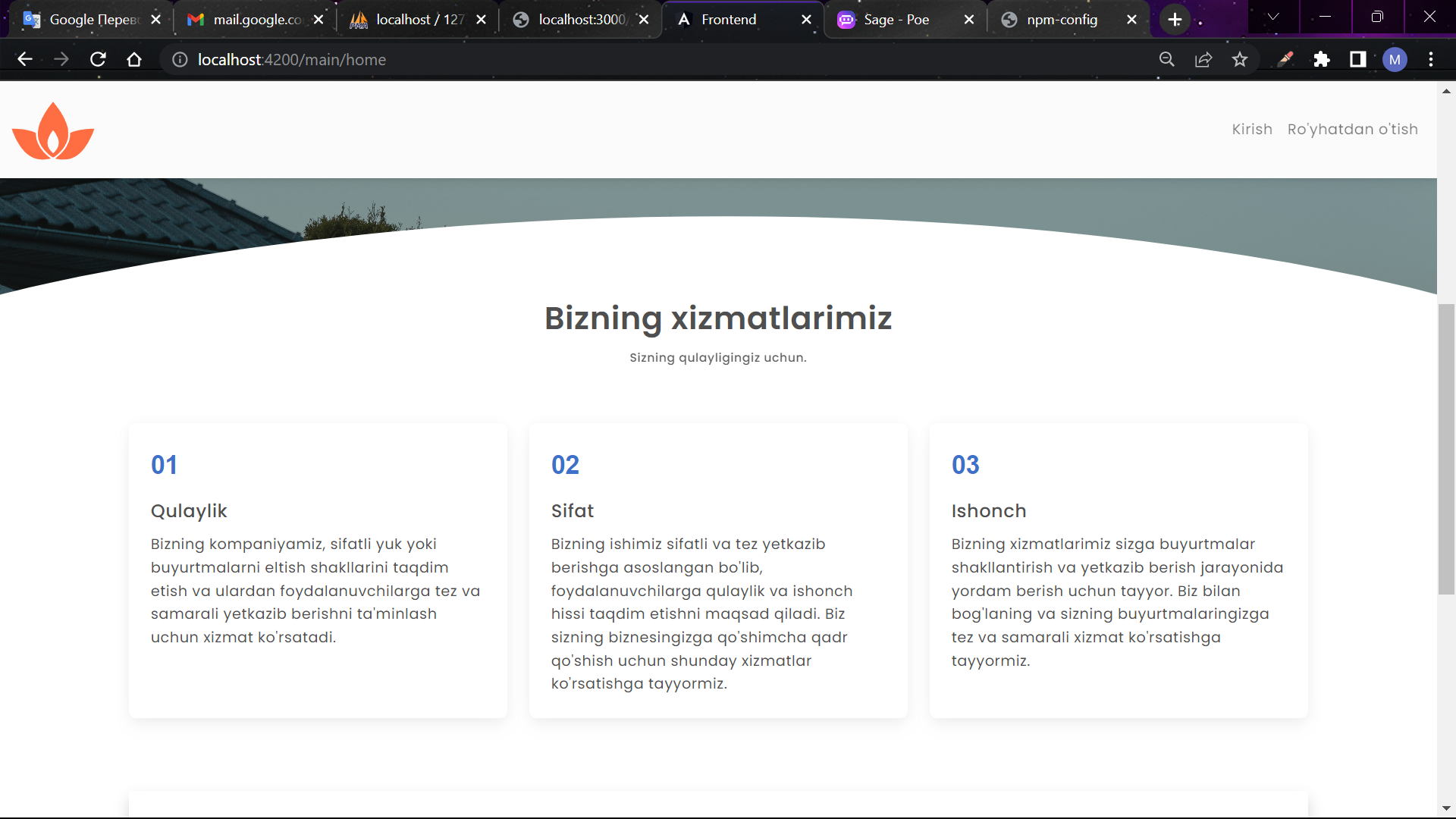
Veb-saytning mijoz qismini ishlab chiqishda, eng yangi texnologiyalar va frontendlarni ishlab chiqish tendentsiyalaridan xabardor boʻlish muhimdir. Bunga javob beruvchi dizayn, progressiv veb-ilovalar va veb-komponentlar kabi zamonaviy veb-texnologiyalardan foydalanish kiradi. Bundan tashqari, semantik HTML-dan foydalanish, Internet uchun tasvirlar va boshqa aktivlarni optimallashtirish, toza va parvarish qilinadigan kod yozish kabi ilgʻor amaliyotlar va sanoat standartlariga rioya qilish muhimdir.

Ushbu bosqichlarni bajarish va toʻgʻri vositalar va usullardan foydalanish orqali siz vizual jihatdan jozibali, funktsional va ishlash va foydalanish imkoniyati uchun optimallashtirilgan mijoz kod bazasini yaratishingiz mumkin. Bu sizning foydalanuvchilar va manfaatdor tomonlarning ehtiyojlariga javob beradigan uzluksiz va qiziqarli foydalanuvchi tajribasini taqdim etishga yordam beradi.

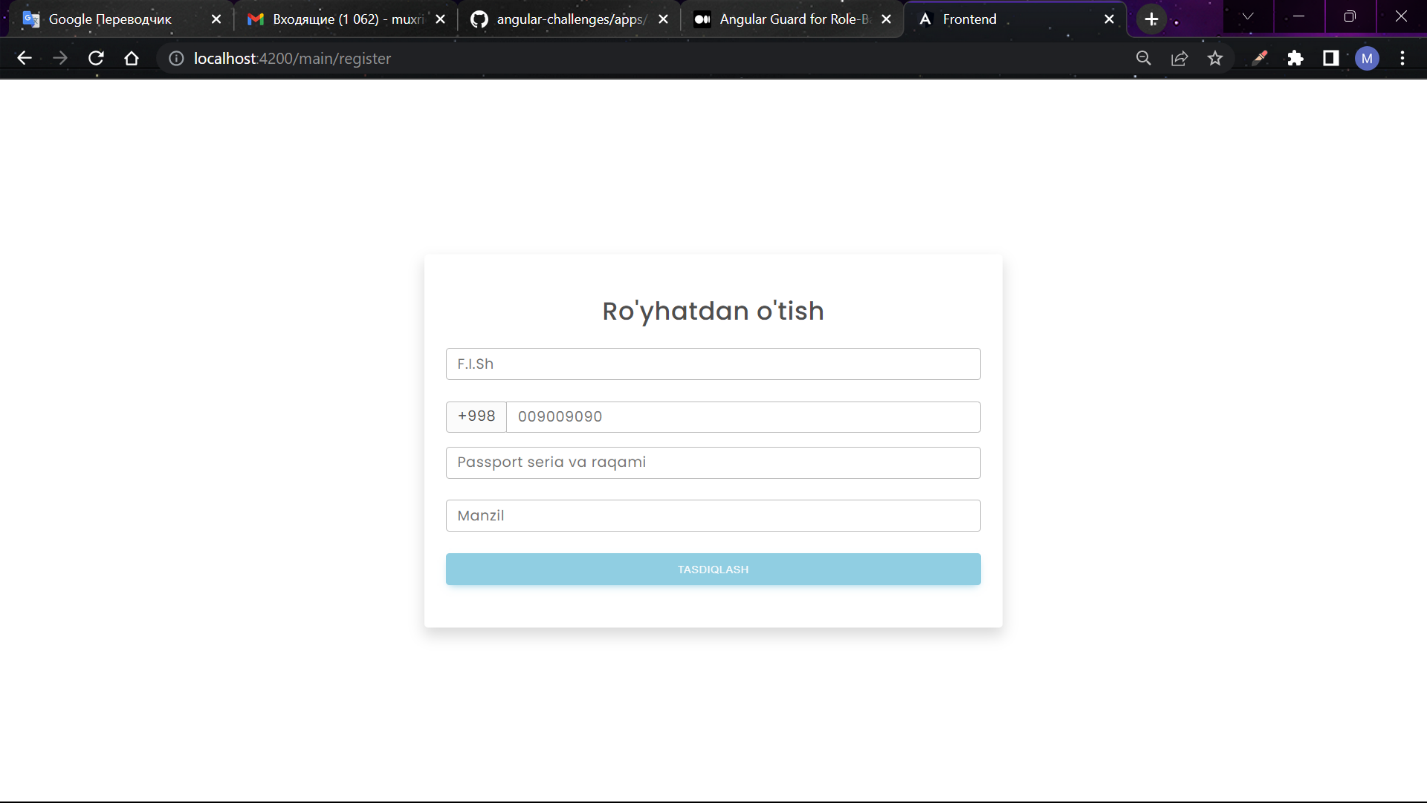
Veb saytni yaratishda HTML CSS va Javascript foydalanilgan. HTML va CSS orqali Veb sayt foydalanuvchi interfeysi ishlab chiqildi, javascript orqali unga turli xil animatsiyalar qoʻshildi. Dizaynerlar uchun vositalarga qoʻshimcha ravishda, Figma avtomatik kod yaratishni qoʻllab-quvvatlaydi, buning yordamida siz element yoki ular guruhining CSS xususiyatlarini tezda olishingiz mumkin. Sayt dizayni ishini namoyish qilish rejimi ham mavjud. Bu ishlab chiquvchiga sahifadagi elementlar qanday harakat qilish kerakligini yaxshiroq tushunish imkonini beradi. Onlayn muharrirning asosiy funksiyasi bepul, ammo qoʻshimcha funksiyalar uchun pul toʻlanishi mumkin. U oʻz API kutubxonalaringizni ulash va sozlash orqali ham kengaytirilishi mumkin. Figmaning asosiy xususiyat shundaki, u boshqa dizayn dasturlaridan farqli oʻlaroq, kompyuterga oʻrnatishni talab qilmaydi. Barcha ishlar brauzer orqali amalga oshiriladi. Funktsionallik hali Photoshop va boshqa professional grafik muharrirlarga etib bormagan boʻlsa-da, bu veb-sayt va ilovalar dizaynini barcha bosqichlarda - birinchi eskizlardan tayyor prototiplargacha ishlab chiqish uchun yetarli.



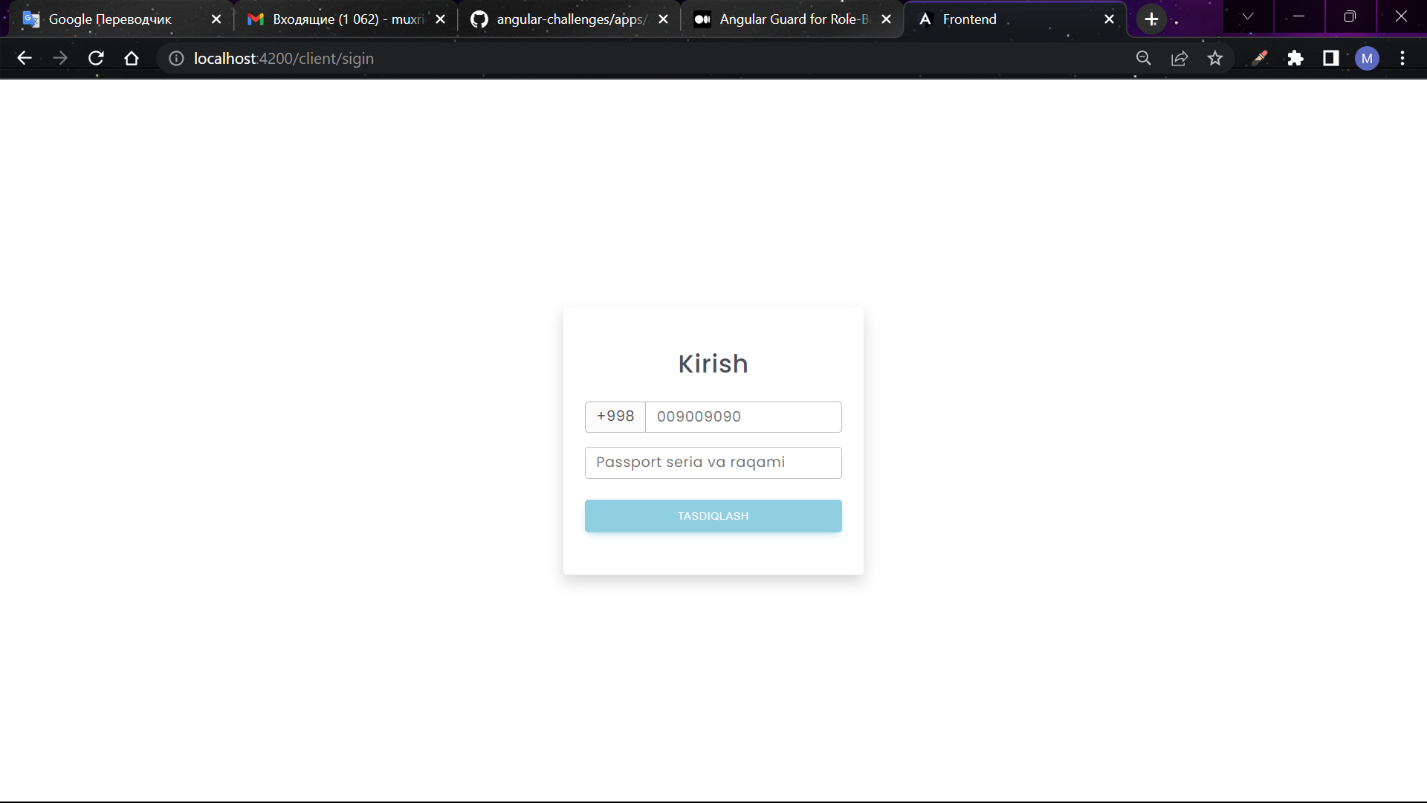
1-rasm. Mijozlarni jalb qiluvchi va asosiy qism.



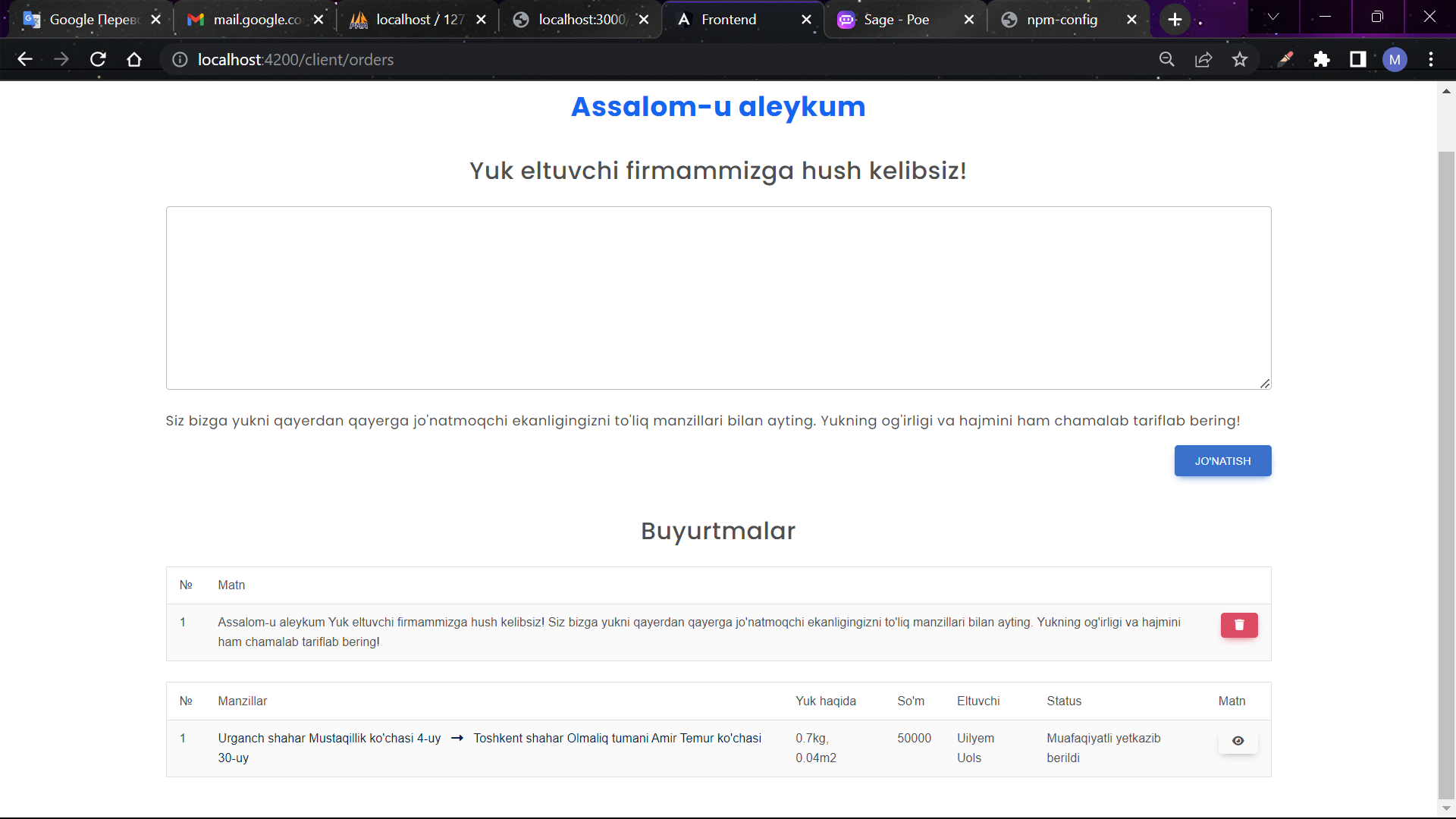
2-rasm. Mijozlarga nima bera olishimizni koʻrsatuvchi qism.



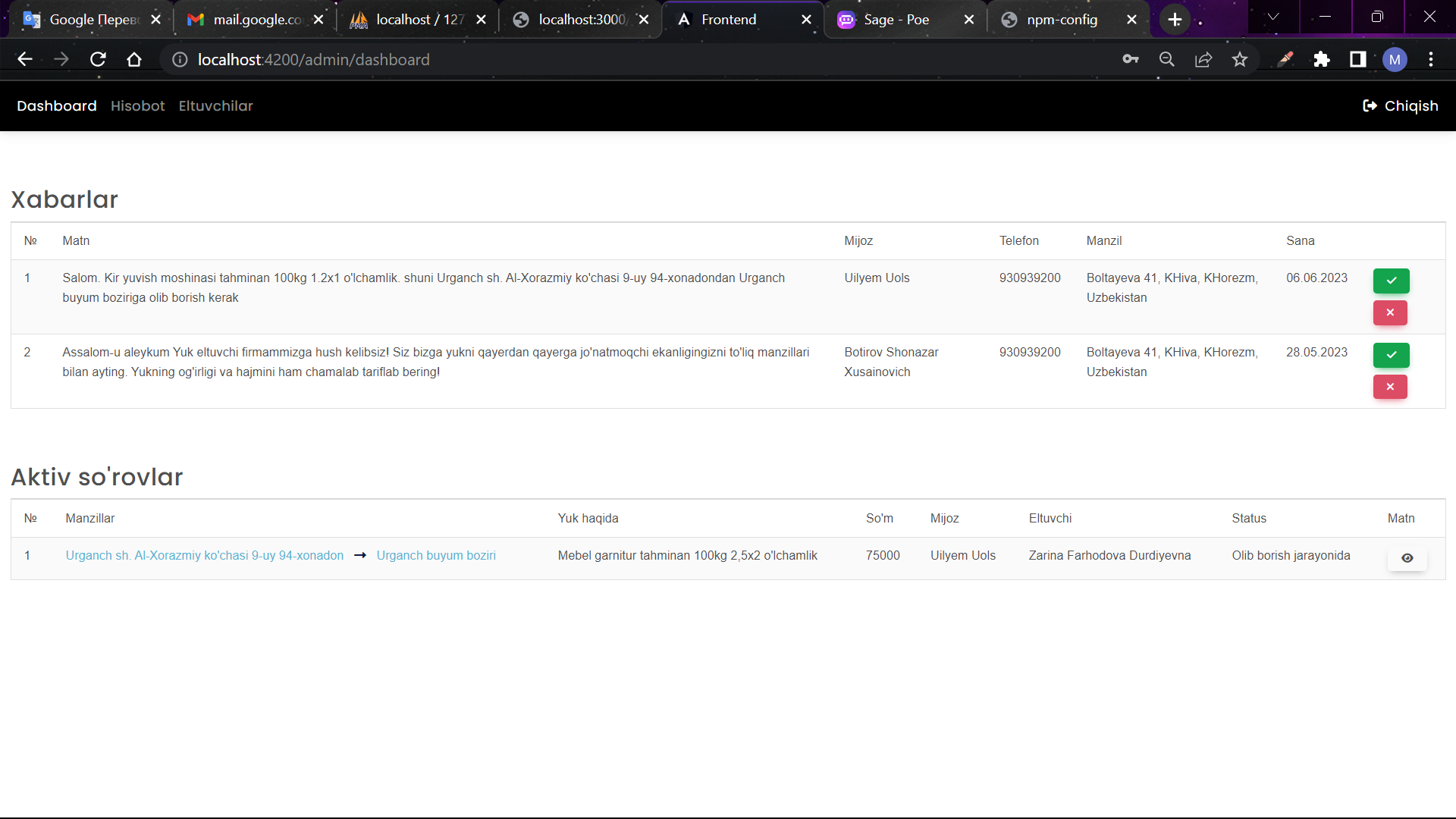
3-rasm. Mijozlarning tizimdan roʻyhatdan oʻtish sahifasi.



4-rasm. Mijozlarning oʻz profiliga kirish qismi.

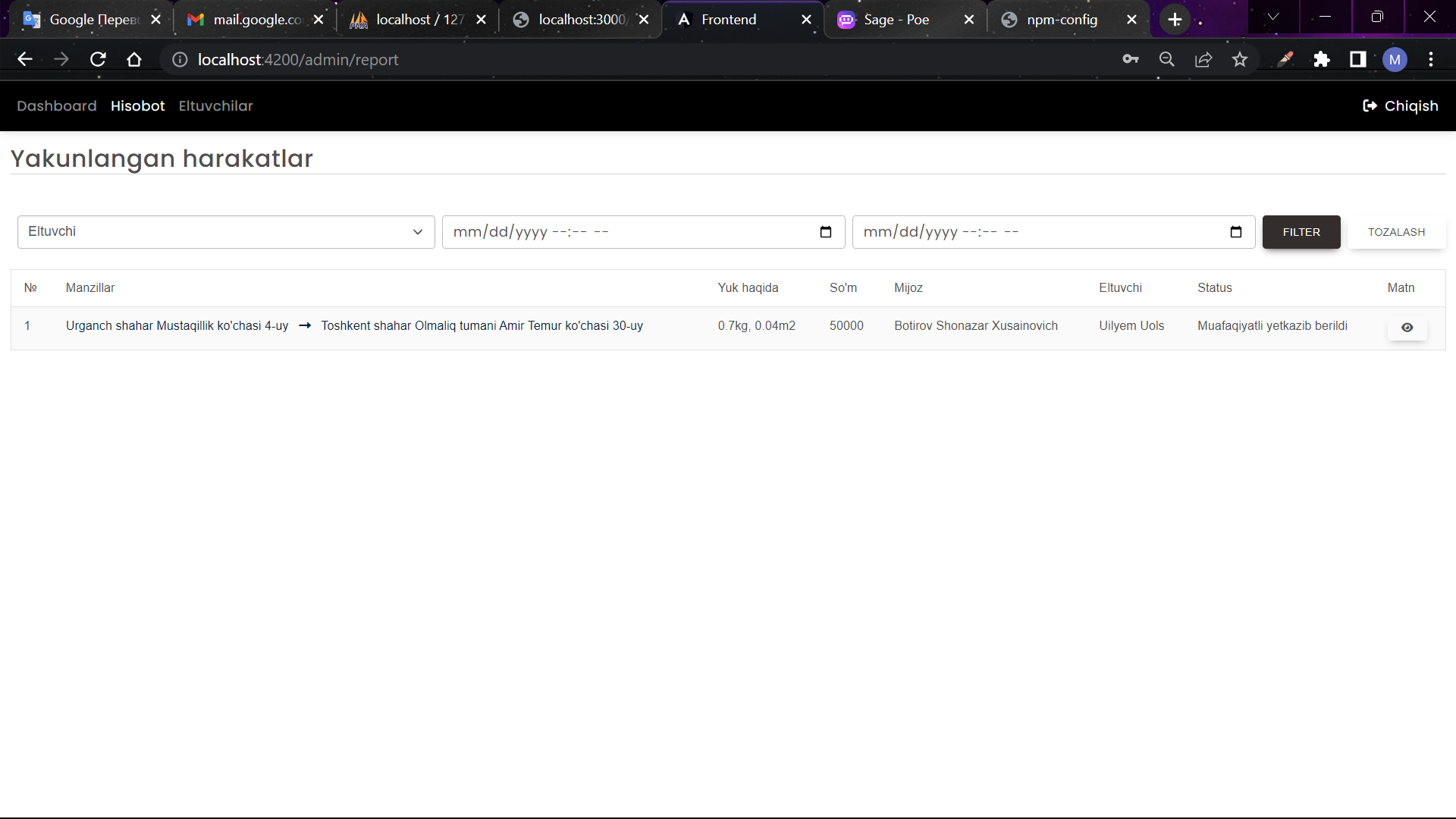


5-rasm. Mijozlarning oʻz profilidagi soʻrovlari va har bir soʻrovning statuslari koʻrsatilib turadi.

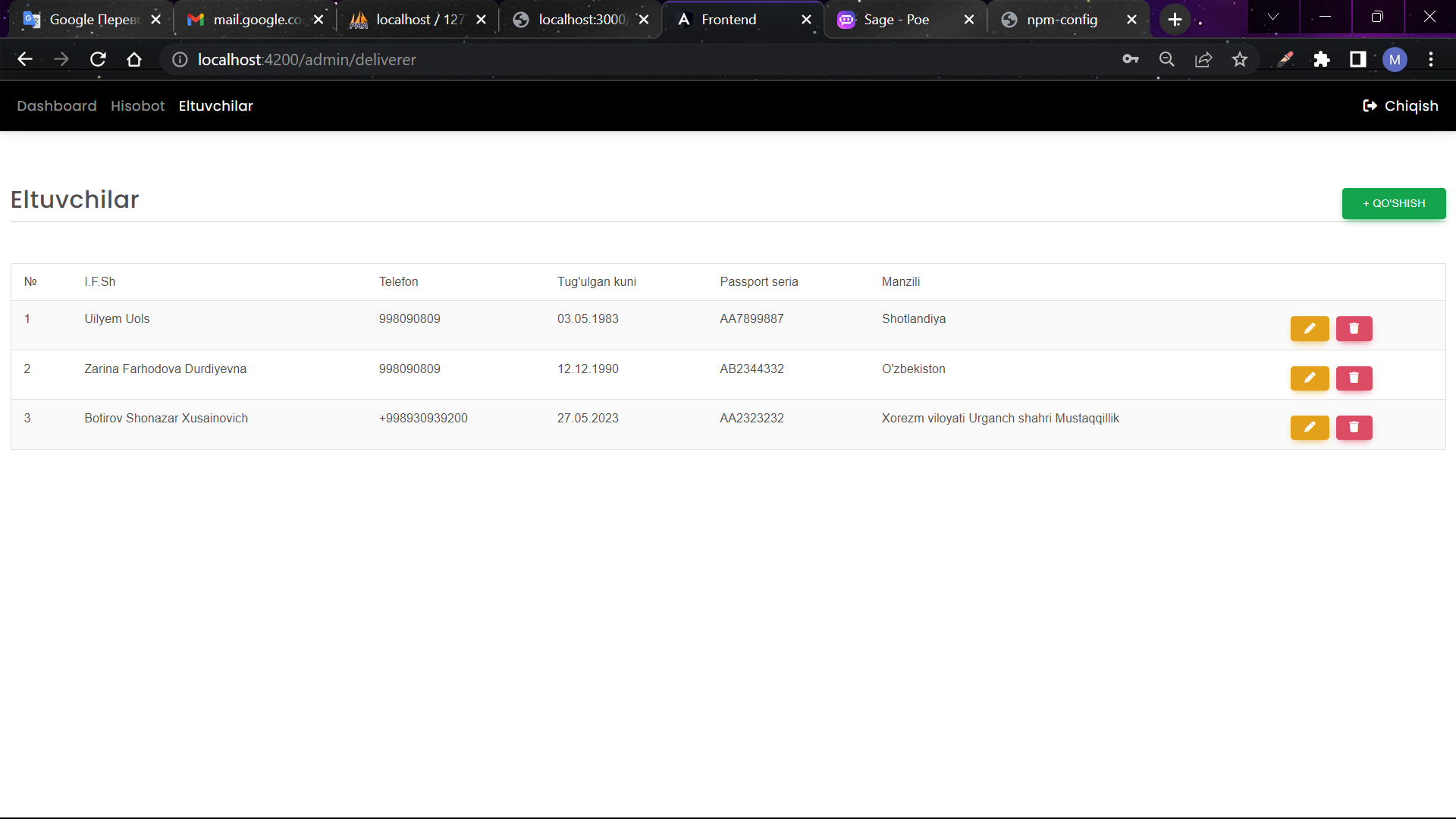


6-rasm. Admin qism.

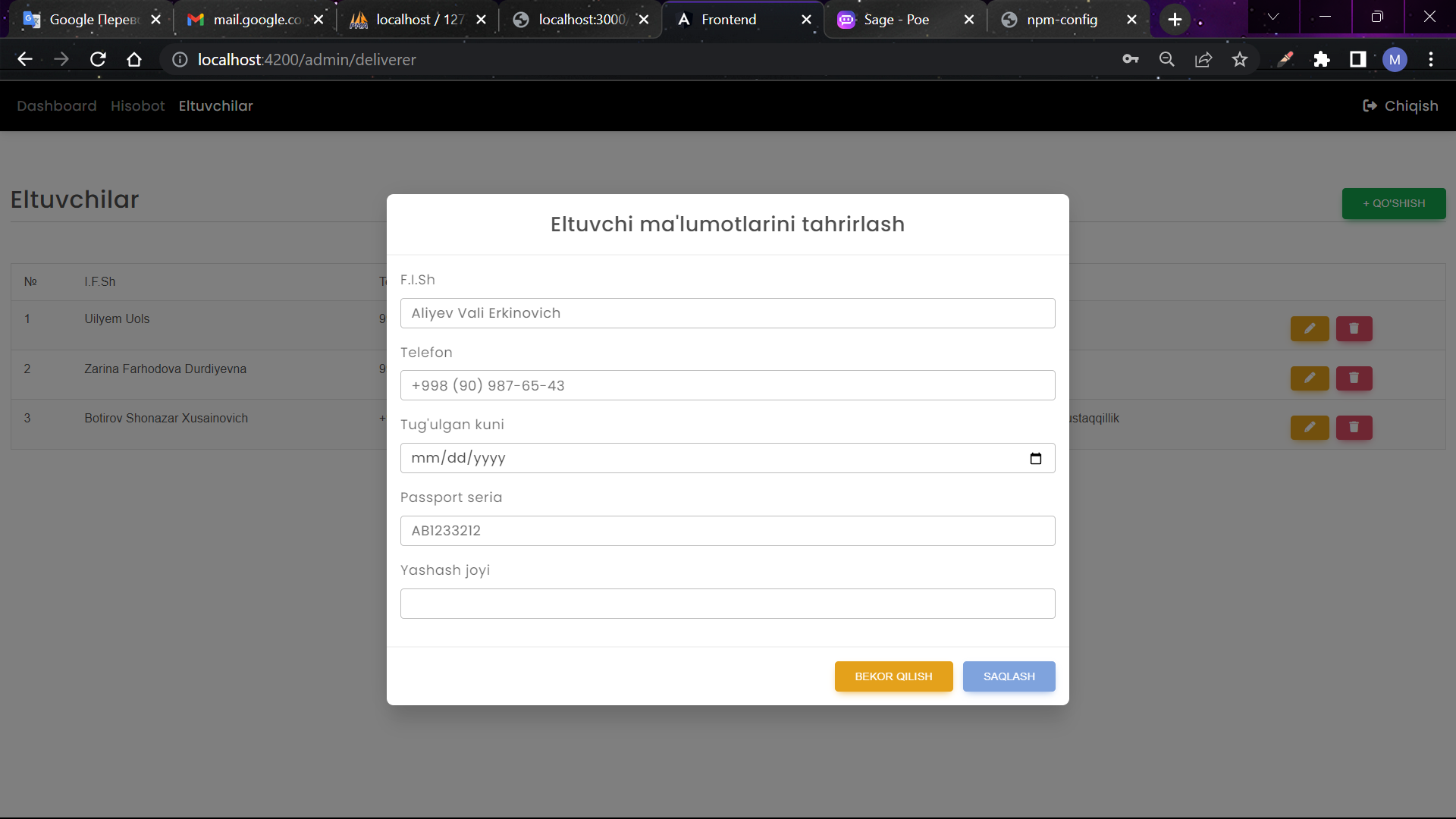
Admin qismida mijozlardan keladigan soʻrovlar shakkilantiriladi va eltuvchilarga birlashtiriladi.



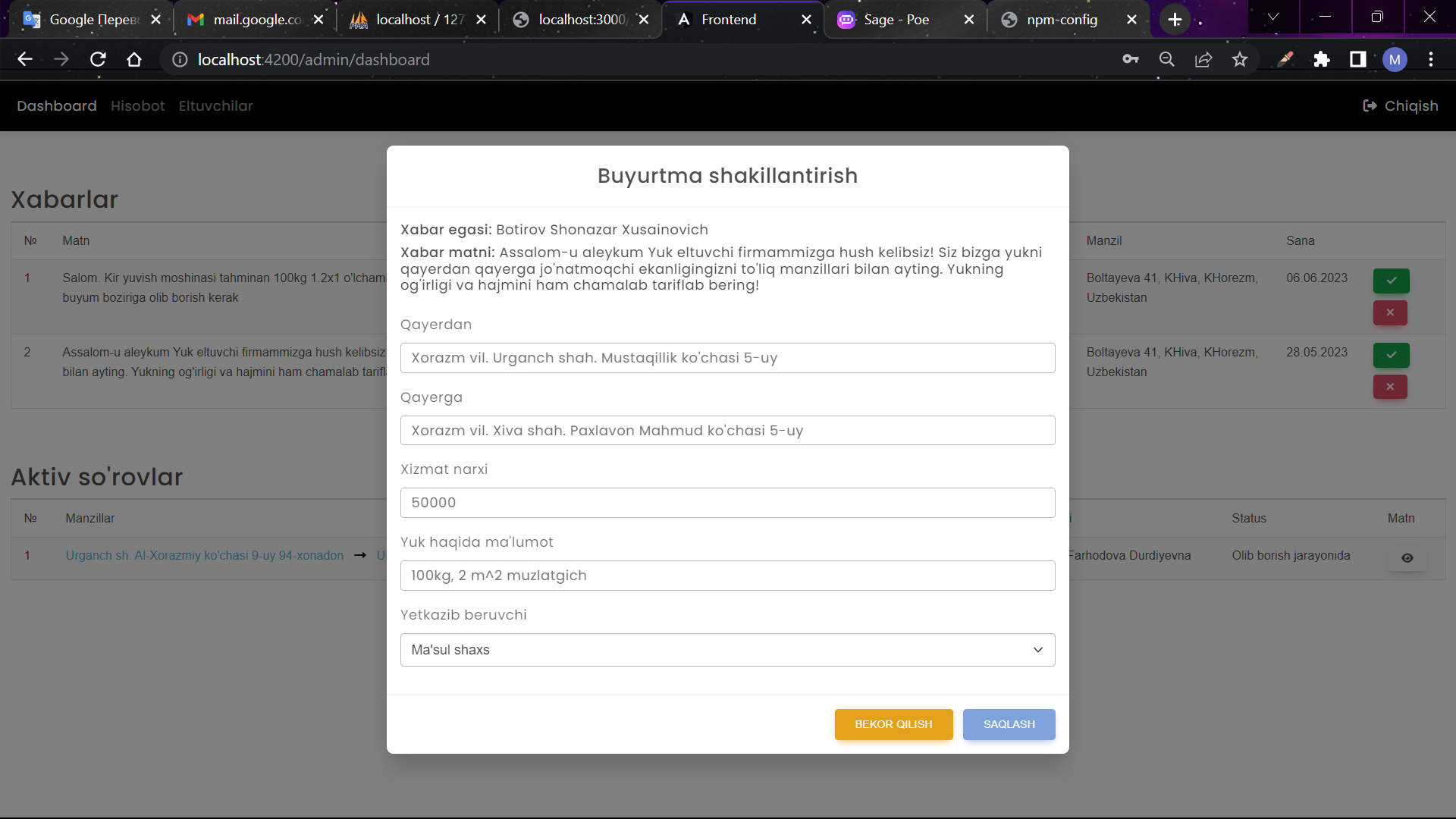
7-rasm. Admin qism Xisobotlar boʻlimi filterlari bilan.



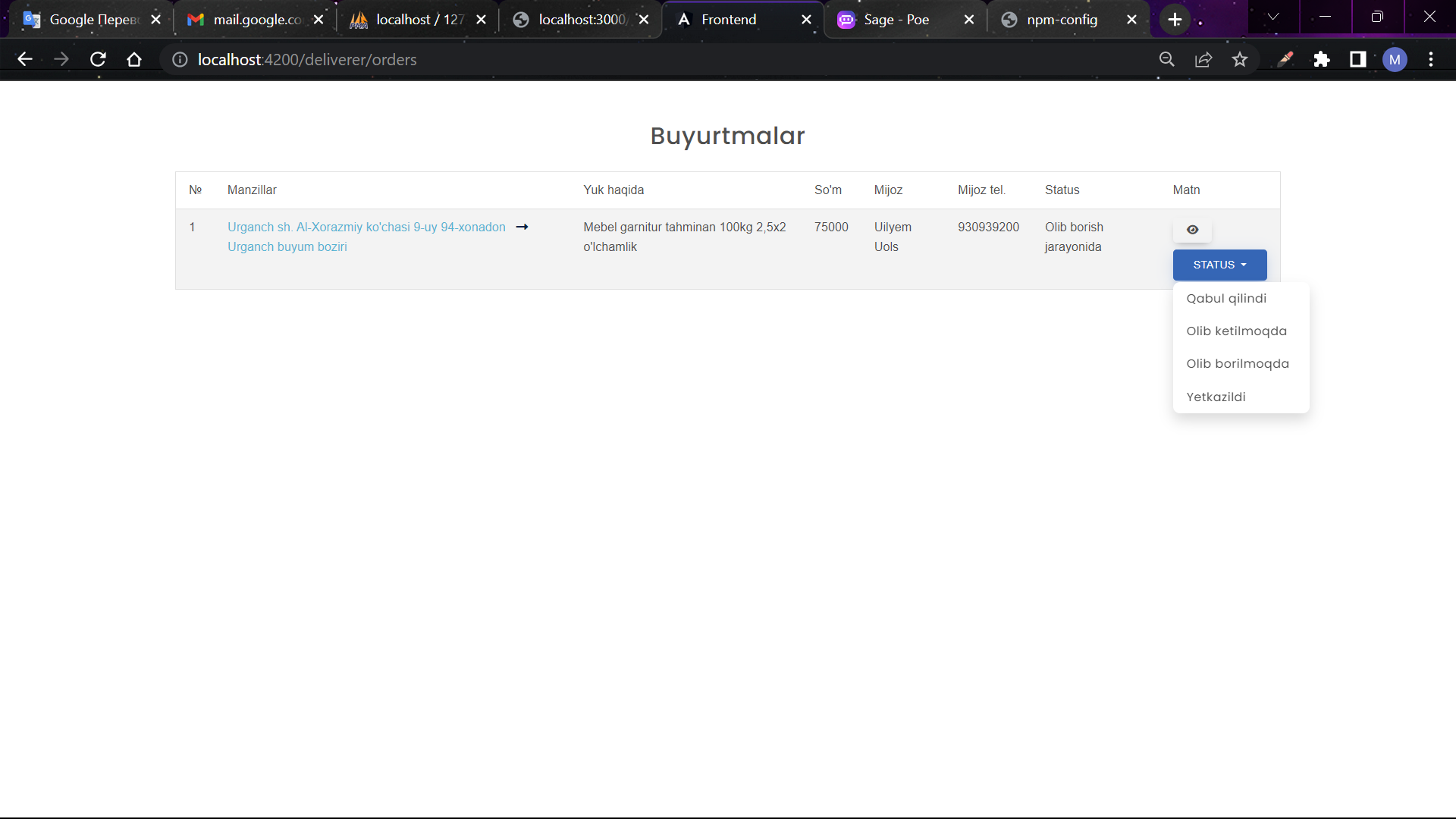
8-rasm. Admin qism. Eltuvchilarni tizimga kiritish va oʻchirish sahifasi.



9-rasm. Admin qism. Yangi eltuvchi qoʻshish.



10-rasm. Admin qism. Yangi buyurtmani tizimga kiritish.



11-rasm. Eltuvchi qism. Oʻziga biriktirilgan buyurtmalarni

holatlarini oʻzgartira oladi.

Mijozlarning soʻrovlari jarayonga qarab oʻzgarib boradi, yaʻni “Qabul qilindi”, “Olib ketish jarayonida”, “Olib boorish jarayonida”, “Yuk muaffaqiyatli topshirildi”, “Arxiv”. Bu statuslarni eltuvchi va admin qism oʻzgartira oladi va oʻz vaqtida mijoz profiliga ham koʻrsatib boriladi.

import { Controller, Get, Post, Body, Param, Delete } from ʻ@nestjs/commonʻ;

import { MessageService } from ʻ./message.serviceʻ;

import { MessageModel } from ʻ./message.modelʻ;

@Controller(ʻmessageʻ)

export class MessageController {

constructor(private readonly messageService: MessageService) {}

@Post()

create(@Body() createMessageDto: MessageModel) {

return this.messageService.create(createMessageDto);

}

@Get(ʻactiveʻ)

findActive() {

return this.messageService.active();

}

@Get()

findAll() {

return this.messageService.findAll();

}

@Delete(ʻ:idʻ)

remove(@Param(ʻidʻ) id: string) {

return this.messageService.remove(+id);

}

}

Yuqoridagi kod orqali mijozning backend ga qiladigan API controllerining kodi.

Quyida mijoz soʻrovlariga javob beradigan backend kodni koʻrsatib oʻtamiz.

import { Injectable } from ʻ@nestjs/commonʻ;

import { MessageModel } from ʻ./message.modelʻ;

import { Connection } from ʻmysql2ʻ;

import { InjectClient } from ʻnest-mysqlʻ;

@Injectable()

export class MessageService {

constructor(@InjectClient() private readonly con: Connection) { }

async create(newData: MessageModel) {

return await this.con.query(ʻINSERT INTO `message`(`client\_id`, `text`) VALUES(?, ?)ʻ, [newData.client\_id, newData.text]);

}

async findAll() {

const messages = await this.con.query(ʻSELECT m.\*, c.fullname, c.phone, c.address FROM `message` AS m, `client` AS c WHERE m.client\_id=c.id ORDER BY m.`date` DESCʻ);

return messages[0];

}

async active() {

const messages = await this.con.query(ʻSELECT m.\*, c.fullname, c.phone, c.address FROM `message` AS m, `client` AS c WHERE m.client\_id=c.id AND `status`=0 ORDER BY m.`date` DESCʻ);

return messages[0];

}

async remove(id: number) {

return await this.con.query(ʻDELETE FROM `message` WHERE id=?ʻ, id);

}

}

Quyida Mijoz qismini frontend dastur kodi keltirilgan:

<div class="container">

<h2 class="section-title">Assalom-u aleykum</h2>

<h3 class="text-center mb-4">Yuk eltuvchi firmammizga hush kelibsiz!</h3>

<form #f="ngForm" (ngSubmit)="send(f)">

<input type="hidden" name="client\_id" [ngModel]="client\_id">

<textarea name="text" ngModel rows="8" class="form-control mb-4"></textarea>

<p class="d-block">Siz bizga yukni qayerdan qayerga joʻnatmoqchi ekanligingizni toʻliq manzillari bilan ayting.

Yukning ogʻirligi va hajmini ham chamalab tariflab bering!</p>

<button type="submit" class="btn btn-primary d-block ms-auto mb-4">Joʻnatish</button>

<div class="mt-4 bg-success text-light py-4 px-3 text-center" [class.d-none]="!showMessage">

<h6 class="mb-0">Habaringiz muafaqiyatli joʻnatildi, iltimos habaringiz tekshiruv natijasini kuting!</h6>

</div>

</form>

<div class="my-5">

<h3 class="mb-4 text-center">Buyurtmalar</h3>

<div class="table-responsive" \*ngIf="orders.length > 0; else notFound">

<table class="table table-striped table-hover border mb-0">

<thead>

<tr>

<th>№</th>

<th>Manzillar</th>

<th>Yuk haqida</th>

<th>Soʻm</th>

<th>Eltuvchi</th>

<th>Status</th>

<th>Matn</th>

</tr>

</thead>

<tbody>

<tr \*ngFor="let item of orders; let i = index">

<td>{{ i + 1 }}</td>

<td [ngClass]="{

ʻtext-dangerʻ: item.status == 0,

ʻtext-warningʻ: item.status == 1,

ʻtext-infoʻ: item.status == 2,

ʻtext-successʻ: item.status == 3,

ʻtext-darkʻ: item.status == 4

}">

{{ item.from }} <i class="fa fa-right-long text-dark mx-2"></i> {{ item.to }}

</td>

<td>{{ item.goods }}</td>

<td>{{ item.summ }}</td>

<td>{{ item.deliverer\_fullname }}</td>

<td>{{ item.status\_name }}</td>

<td>

<button (click)="setText(item.message\_text)" class="btn btn-sm" data-mdb-toggle="modal" data-mdb-target="#textModal">

<i class="fa fa-eye"></i>

</button>

</td>

</tr>

</tbody>

</table>

</div>

<ng-template #notFound>

<div class="my-5">

<p class="text-center text-black">Hech narsa topilmadi!</p>

</div>

</ng-template>

</div>

</div>

<div class="modal fade" id="textModal" tabindex="-1" aria-labelledby="textModalLabel" aria-hidden="true">

<div class="modal-dialog">

<div class="modal-content">

<div class="modal-header text-center">

<h5 class="modal-title" id="textModalLabel">Buyurtma matni</h5>

</div>

<div class="modal-body">

{{ text }}

</div>

<div class="modal-footer">

<button data-mdb-dismiss="modal" type="button" class="btn btn-primary">Yopish</button>

</div>

</div>

</div>

</div>

Quyida keltirilgan kod esa mijoz qism uchun maʻlumotlar bazasining jadvalining kodi;

CREATE TABLE `client` (

`id` int(11) NOT NULL,

`fullname` varchar(50) NOT NULL,

`address` varchar(200) NOT NULL,

`phone` varchar(20) NOT NULL,

`passport\_seria` varchar(12) NOT NULL,

`date` datetime NOT NULL DEFAULT current\_timestamp()

)

Pastdagi kod yordamida esa bazadagi barcha soʻrovlarni koʻrish mumkin boʻladi

async findAll() {

let sql = ʻSELECT o.\*, m.text AS `message\_text`, ʻ +

ʻc.fullname AS `client\_fullname`, c.phone AS `client\_phone`, ʻ +

ʻd.fullname AS `deliverer\_fullname`, d.phone AS `deliverer\_phone`, ʻ +

`IF(o.status = 0, ʻQabul qilindiʻ, ` +

`IF(o.status = 1, ʻOlib ketish jarayonidaʻ, ` +

`IF(o.status = 2, ʻOlib borish jarayonidaʻ, ` +

`IF(o.status = 3, ʻMuafaqiyatli yetkazib berildiʻ, ʻBajarilganʻ)))) AS status\_name ` +

ʻFROM `order` AS o, `message` AS m, `deliverer` AS d, `client` AS c ʻ +

ʻWHERE o.message\_id = m.id AND o.deliverer\_id = d.id AND m.client\_id=c.idʻ;

const data = await this.con.query(sql);

return data[0];

}

# 2.3-§ Foydalanuvchiga yoʻriqnoma

Hurmatli foydalanuvchi ushbu Veb ilova turistik firmalar ishini elektronlashtirish uchun ishlab chiqildi.

Dasturdan foydalanish boʻyicha tushunmovchiliklar boʻlsa ushbu yoʻriqnomani toʻliq oʻqib chiqishingizni iltimos qilamiz.

Veb ilovaga kirganingizdan keyin Veb ilovaning asosiy qismi ochiladi. Unda yuklarni eltuvchi firmalar uchun maʻlumot qismi va roʻyhatdan oʻtish sahifasi mavjud. Mizojlar yuklarini eltish uchun firma saytidan roʻyhatdan oʻtgach Kirish qismidan oʻz profiliga oʻtishadi va veb saytimizning muhim qismi boshlanadi.

Veb ilovaning asosiy quyidagi 4 ta boʻlim mavjud:

- **Admin**

- **Eltuvchilar**

- **Mijoz profili**

- **Mijozlar uchun reklama va aloqa qismi**

Yuqoridagi boʻlimlarning har birsining ozʻ maqsadi va vazifasi mavjud.

# XULOSA

Ushbu bitiruv malakaviy ishida yuk eltuvchi xizmati uchun veb sayt yaratilishidan oldin boshqa shu kabi veb saytlar oʻrganib chiqildi. Internetdagi barcha shunga oʻxshash veb saytlar oʻrganib chiqildi va ulardagi bazi xato kamchiliklar toʻgʻirlanib ushbu veb sayt yaratildi. Veb saytni yaratish uchun turli dasturlash muhitlari, framework va kutubxonalari tahlil qilindi va aniq masalalar belgilab olindi. Avvalo veb sayt qaysi dasturlash tili, framework va kutubxonalar asosida yaratilishi aniqlab olindi.

Xulosa qilib aytganda, yuk eltuvchi firmalar uchun qilingan veb sayt yuk eltuvchi xizmat usullarini inqilob qildi. Ushbu veb saytlar qulaylik, shaffoflik va tanlash uchun kengroq imkoniyatlar kabi bir qator afzalliklarni taklif qiladi. Mijozlar soʻrov qilishdan oldin turli oʻlcham va ogʻirlikdagi yukllarni osongina koʻrib chiqishlari, narxlarni solishtirishlari va boshqa mijozlarning sharhlarini oʻqishlari mumkin. Firma egalari esa reklama qilish uchun turli marketing vositalaridan foydalanishlari mumkin. Shuningdek, ular ushbu veb saytlar tomonidan taqdim etilgan xizmatlardan foydalanib, potentsial xaridorlar bilan muzokaralar olib borish va hujjatlar bilan shugʻullanish muammosidan qochishlari mumkin.

Umuman olganda, yuk eltuvchi firma veb saytlari odamlarga yuklarini ishonchli va tez yetkazish usulini taqdim etdi, ammo har qanday qaror qabul qilishdan oldin ushbu operatsiyalarga ehtiyotkorlik bilan yondashish va toʻgʻri tadqiqot qilish juda muhimdir. Prototip tayyor boʻlgach bevosita front end qismi uchun Angular va back end qismi uchun NestJS dan foydalanib yozildi va testlandi. Veb saytda kelajakda foydalanuvchilarda tushunmovchiliklar yuzaga kelganda qoʻllashlari uchun “Foydalanuvchi yoʻriqnomasi” yozildi. Hamma tekshiruvlardan muvaffaqiyatli oʻtgach veb sayt uchun domen olinib, internetga joylanmoqda.

# **FOYDALANILGAN ADABIYOTLAR ROʻYHATI**

1. Казаков Г. С. Система автоматизации проектирования баз данных на основе модели «сущность-связь». // Информационные системы и измерительно-вычислительные комплексы: сборник докладов студентов и аспирантов кафедры «Измерительно вычислительные комплексы» на научно-технических конференциях / Под ред. В. В. Родионова, 2016. С. 25–31.

2. Balashov E.P., Puzankov D.V. Axborot va boshqaruv tizimlarini loyihalash.

3. Dimov E.M., Diyazitdinova A.R., Kachkov D.A. Axborot tizimlarini loyihalash: Samara: PGATI, 2013.

4. Рыжко A. Л., Рыбников Н. А., Рыжко Н. А. Информационные системы управления производственной компанией. М.: Юрайт, 2016. 356 с.

5. Дэйт К. Дж. Введение в системы баз данных. М.: Вильямс, 2017. 1328 с.

6. Кузнецов С. Д. Основы баз данных. М.: Бином, 2007. 488 с.

7. Вигерс К. И. Разработка требований к программному обеспечению. СПб.: БХВ-Петербург, 2018. 736 с

8. Методические рекомендации по подготовке и оформлению выпускной квалификационной работы (проекта) для технических направлений подготовки 09.03.01 Информатика и вычислительная техника, 09.03.04 Программная инженерия, 12.03.01 Приборостроение, 23.03.01 Технология транспортных процессов / сост. Л.Н. Буйлушкина. - Нижневартовск, 2017.

9. Информационные технологии / О.Л. Голицына, И.И. Попов, Н.В. Максимов, Т.Л. Партыка. – М, Издательство Инфра-М, 2009. – 608 с

**Internet resurslar**

1. Angular hujjatlari (<https://angular.io/docs>): Angularning asosiy tushunchalari va xususiyatlarini, jumladan komponentlarni yaratish, marshrutlash, xizmatlardan foydalanish va shablonlar bilan ishlashni oʻrganish uchun ishlatilgan
2. NestJS hujjatlari (<https://nestjs.com/documentation>): NestJS hujjatlari kontrollerlar, xizmatlarni yaratish, marshrutlash va maʼlumotlar bazasi bilan ishlash kabi asosiy tamoyillari va xususiyatlarini oʻrganish uchun ishlatilgan.
3. MySQL hujjatlari (<https://dev.mysql.com/doc/>): MySQL hujjatlari MySQL maʼlumotlar bazasi bilan ishlash asoslarini, jumladan jadvallar yaratish, tanlangan soʻrovlarni bajarish, maʼlumotlarni qoʻshish, yangilash va oʻchirishni oʻrganish uchun ishlatilgan.
4. JavaScript - rasmiy sayt [Elektron resurs]. - URL: <https://https://www.javascript.com/>.