|  |
| --- |
| **Full stack development** |
| AKBAR MAMANAZAROV |

**Unit 7:** Full stack development

**O'qituvchi:** Muxammadsobir Turdaliyev

**Guruh raqmi:** 24-212

**Talaba ID raqami:** 240258

**Taqdim etilgan sana:** 27.05.2025

**BTEC O'QUVCHILAR TOPSHIRIQLARINI BAHOLASH VA DEKLRITSIYASI**

Baholash uchun ishlarnni taqdim etganda, har bir o'quvchi ish o'ziniki ekanligini tasdiqlovchi deklaratsiyani imzolashi kerak.

|  |  |
| --- | --- |
| **O'quvchi (talaba) identifikatori:** | 240258 |
| **Baholovchi nomi:** | Muxammadsobir Turdaliyev |
| **BTEC dasturi nomi:** | Pearson BTEC Higher Nationals in Information Technologies |
| **Birlik yoki komponent raqami va nomi:** | Full stack development |
| **Topshiriq nomi:** | Full stack development |
| **Topshiriq topshirilgan sana:** | 27.05.2025 |

Iltimos, har bir topshiriq uchun berilgan ishlarni sanab o'ting. Ishlarni topish mumkin bo'lgan sahifa raqamlarini ko'rsating yoki ishlarning mohiyatini tavsiflang (masalan, diagramma, rasm).

|  |  |  |
| --- | --- | --- |
| **Topshiriq vazifasi ma'lumoti** | **Ishlar taqdim etildi** | **Sahifa** |
|  |  |  |
|  |  |  |
|  |  |  |

|  |
| --- |
| **O'quvchi deklaratsiyasi**  Ushbu topshiriq uchun taqdim etilgan ish meniki ekanligini tasdiqlayman. Ishda foydalanilgan manbalarga aniq havola qildim. Men noto'g'ri deklaratsiya noto'g'ri ishlashning bir shakli ekanligini tushunaman.  O'quvchi imzosi: \_\_\_\_\_\_\_\_\_\_\_\_\_\_ Sana: 27.05.2025 |

**KIRISH**

Bugungi raqamli asrda kompaniyalar uchun tezkorlik, qulaylik va ishonch asosiy tamoyillarga aylanib bormoqda. Ayniqsa, IT xizmatlari bilan shug‘ullanadigan kichik korxonalar uchun mijozlar bilan to‘g‘ridan-to‘g‘ri, oddiy va samarali aloqa o‘rnatish nihoyatda muhim. Dern-Support kompaniyasi aynan shunday yo‘nalishda faoliyat yuritadi — ya’ni kompyuter tizimlarini ta’mirlash, texnik xizmat ko‘rsatish va foydalanuvchilarga texnik yordam berish bilan shug‘ullanadi. Ammo kompaniya hozirgi kunda qog‘ozli jarayonlar va telefon orqali xizmat ko‘rsatishga tayanmoqda. Bu esa ishning sifatiga, samaradorligiga va mijozlarning qoniqish darajasiga bevosita ta’sir qilmoqda.

Shu sababli, ushbu loyiha orqali kompaniyaning kundalik ish faoliyatini avtomatlashtirish, mijozlar bilan bog‘lanish tizimini soddalashtirish va xizmat sifatini oshirish uchun zamonaviy to‘liq stek asosida ishlaydigan veb-ilova ishlab chiqish maqsad qilindi. Loyihada frontend va backend qismlari birgalikda ishlab chiqilib, real muammolarni hal qiluvchi, funksional va foydalanuvchiga qulay interfeysga ega tizim yaratilmoqda.

Ushbu tizim mijozlarga o‘zlari uchun shaxsiy hisob ochish, xizmat buyurtma qilish, muammo haqida xabar yuborish va xizmat narxini bilish imkonini beradi. Boshqa tarafdan, admin foydalanuvchilar esa xabarlarni ko‘rib chiqish, ularni holatiga qarab saralash, kerak bo‘lsa o‘chirish yoki o‘qilgan deb belgilash, shuningdek foydalanuvchilar ro‘yxatini yuritish kabi amallarni bajarishi mumkin. Bularning barchasi to‘liq stek yondashuvi orqali amalga oshiriladi. Backendda Node.js va Express.js ishlatilgan bo‘lib, ma’lumotlar JSON fayllarda saqlanmoqda. Frontend esa HTML, CSS (Tailwind) va JavaScript orqali qurilgan.

Loyiha doirasida avvalo kompaniyaning ehtiyojlari chuqur tahlil qilindi. Shundan so‘ng, foydalanuvchilarning roli asosida tizim strukturasini yaratish yo‘lga qo‘yildi. Har bir foydalanuvchi — u mijoz bo‘ladimi yoki administrator — o‘z ehtiyojiga mos sahifaga ega bo‘ladi. Kirish va ro‘yxatdan o‘tish sahifalari, foydalanuvchilarning shaxsiy kabineti, admin panel, xabarlar bo‘limi va xizmatlar haqidagi sahifalar — bularning barchasi o‘zaro uzviy bog‘langan holda ishlaydi. Bunda foydalanuvchi tajribasi birinchi o‘ringa qo‘yilgan.

Tizimning asosiy ustunligi — uning soddaligi va tezkor ishlashidir. Murakkab CMS yoki katta hajmli ma’lumotlar bazasi o‘rniga, oddiy JSON fayllardan foydalanilgan. Bu kichik korxonalar uchun ayni muddao: ham tejamkor, ham texnik xizmat ko‘rsatish oson. Dastlabki versiya ishlab chiqilib, foydalanuvchilardan fikr-mulohaza olindi va ularning asosida bir qancha yaxshilanishlar amalga oshirildi. Masalan, foydalanuvchi nomi takrorlanmasligi uchun autentifikatsiya bo‘limi kuchaytirildi, xabar yuborishdagi validatsiya tekshiruvlari aniqlandi, admin panelga foydalanuvchi rollari asosida kirish imkoniyati yaratildi.

Bu loyiha orqali nafaqat kompaniya ichki ishlarini raqamlashtirish, balki biznesning keyingi rivojlanishiga mustahkam poydevor qo‘yish ham ko‘zda tutilgan. Masalan, hozircha lokal fayllarda saqlanayotgan ma’lumotlar kelajakda MongoDB yoki MySQL bazasiga o‘tkazilishi, foydalanuvchilar soni oshganda esa tizimda autentifikatsiya tizimi JWT yoki OAuth kabi zamonaviy texnologiyalar bilan mustahkamlanishi rejalashtirilmoqda.

Shuningdek, frontend dizayn ham mijozlarning fikriga moslab tanlandi — sodda, zamonaviy va moslashuvchan. Tailwind CSS yordamida barcha sahifalar moslamalarga mos holatda ishlab chiqilgan: kompyuter, planshet va telefonlarda to‘liq mos ko‘rinadi. Natijada, foydalanuvchi qayerda bo‘lishidan qat’iy nazar, kompaniya bilan bemalol bog‘lanish imkoniga ega bo‘ladi.

Bu loyiha orqali kompaniyaning butun xizmat jarayoni raqamli shaklga o‘tkazilmoqda. Bu esa nafaqat mijozlar uchun qulaylik, balki kompaniya ichki ishlari uchun aniqlik, statistik tahlil va ish unumdorligini oshirish imkonini beradi. Yaratilgan to‘liq stek veb-ilova real ehtiyojlarga javob beradigan, foydalanuvchi ehtiyojini tushunadigan va foydalanishda soddaligi bilan ajralib turadigan zamonaviy yechimdir.

**1-TOPSHIRIQ**

Bu loyiha ustida ishlash davomida men to‘liq stek (full stack) yondashuvining real hayotdagi muammolarni qanday hal qilishga yordam berishini o‘z ko‘zim bilan ko‘rdim. Ayniqsa, Dern-Support kompaniyasi uchun ishlab chiqilgan platforma misolida, to‘liq stek yondashuvi nafaqat texnik, balki biznes va foydalanuvchi tajribasi nuqtayi nazaridan ham qanday foyda keltirishini chuqur angladim.

**Texnik yondashuv**

Dastlab kompaniya ehtiyojlarini o‘rganganimda, eng katta muammo bu xizmat so‘rovlarini faqat telefon orqali qabul qilinishi, ma’lumotlar esa biron joyda markazlashmagan holda saqlanishi ekanligini tushundim. Shunday ekan, men ishlab chiqqan yechim bu muammoni to‘liq stek yondashuvi yordamida hal qila oldi.

Frontendni HTML, CSS (Tailwind) va JavaScript yordamida yaratdim. U orqali mijozlar tizimga kiradi, ro‘yxatdan o‘tadi va xabar qoldiradi. Backend qismi esa Node.js va Express.js orqali yozildi. Bu server ma’lumotlarni JSON fayllarda saqlaydi va kerak bo‘lsa ularni qayta ishlaydi. Men uchun bu yondashuv ancha qulay bo‘ldi, chunki barcha tizim bitta texnologik to‘plamda boshqarildi.

**Ish jarayoniga ta’siri**

Loyihani yakka holda bajarayotganim sababli, to‘liq stekni tanlaganim juda foydali bo‘ldi. Men frontend va backend kodlarini birgalikda boshqarib, turli komponentlar o‘rtasida tez muvofiqlashtirishga erishdim. Bu nafaqat ishlab chiqish jarayonini tezlashtirdi, balki tizimni sinab ko‘rishda, xatolarni topishda va ularni tuzatishda ham yengillik berdi. Ayniqsa, foydalanuvchi roli asosida sahifalarga kirishni boshqarish (masalan, admin va oddiy foydalanuvchi farqini ajratish) uchun backend va frontendni bog‘lashda tajribam ancha oshdi.

**Foydalanuvchi tajribasiga ta’siri**

Men yaratayotgan tizimda foydalanuvchining qulayligini har doim birinchi o‘ringa qo‘ydim. Chunki mijozlar uchun qulay interfeys ularning tizimdan foydalanish istagini oshiradi. Kirish/ro‘yxatdan o‘tish sahifalarini iloji boricha soddalashtirdim, xabar yuborish formasini esa faqat zarur maydonlar bilan to‘ldirdim. Natijada, foydalanuvchilar o‘z so‘rovlarini hech qanday ortiqcha murakkabliksiz yubora olishdi.

Adminlar uchun esa maxsus “dashboard” sahifasini yaratdim. U orqali ular kelgan xabarlarni ko‘ra oladi, ularni saralab, holatini o‘zgartiradi yoki kerak bo‘lsa o‘chiradi. Men bu sahifani ko‘proq real ehtiyojlarga moslab, minimal, lekin samarali ko‘rinishda tuzdim.

Bu loyiha orqali men shuni tushundimki, to‘liq stek faqat kod yozish bilan bog‘liq emas. U biznes jarayonlarini raqamlashtirish, statistik tahlillar qilish va qaror qabul qilishda katta foyda beradi. Masalan, har bir yuborilgan xabar sanasi, mavzusi, holati va mijoz ma’lumotlari bilan birga saqlanmoqda. Bu ma’lumotlar asosida keyinchalik kompaniya xizmatlar sifati, eng ko‘p uchraydigan muammolar yoki xizmatga ketadigan o‘rtacha vaqt haqida to‘liq tahlil yurita oladi.

Men kod yozayotganda buni hisobga olib, har bir foydalanuvchi harakatini (xabar yuborish, holatni o‘zgartirish) maxsus “timestamp” bilan birga yozilishini ta’minladim. Bu esa statistik tahlil uchun juda muhim bo‘ldi.

**Kengaytirish va moslashuv**

Men bu loyihani yaratishda kelajakni ham hisobga oldim. Hozircha JSON fayllardan foydalanilgan bo‘lsa-da, kerak bo‘lsa MongoDB yoki boshqa ma’lumotlar bazasiga oson o‘tkazish mumkin bo‘lgan strukturani tuzdim. Bundan tashqari, foydalanuvchi autentifikatsiyasi ham shunday yozilganki, uni kelajakda JWT yoki boshqa xavfsizlik protokollari bilan kengaytirish oson.

Frontend esa Tailwind yordamida to‘liq moslashuvchan qilingan: telefon, planshet yoki kompyuter — qaysi qurilmada ochilmasin, sahifa o‘zini avtomatik moslashtiradi. Men bu ishga alohida e’tibor berdim, chunki foydalanuvchilar har doim bir xil sifatli tajribani his qilishi kerak deb hisoblayman.

**FULLSTACK YONDASHUV ASOSIDA VEB-LOYIHA ISHLAB CHIQISH**

Men Dern-Support uchun veb-platforma ishlab chiqish jarayonida fullstack yondashuvni tanladim, chunki bu usul aynan mening vaziyatimga eng to‘g‘ri keladigan echim edi. Kompaniyaning ehtiyoji oddiy: mijozlar bilan tez, qulay va avtomatlashtirilgan aloqa o‘rnatish, xizmat so‘rovlarini qabul qilish, ularni ko‘rib chiqish va adminlar uchun nazorat tizimini yaratish kerak edi. Bu ehtiyojni hal qilishda fullstack uslub menga juda ko‘p yordam berdi.

Avvalo, foydali tomonlarini aytadigan bo‘lsam — bu usul menga mustaqil harakat qilish imkonini berdi. Men frontend (interfeys) qismini Tailwind CSS va JavaScript yordamida soddalashtirilgan, foydalanuvchiga qulay ko‘rinishda qurdim. Aynan shu paytda foydalanuvchi nima ko‘radi, qanday forma to‘ldiradi, qayerga bosadi — bularni aniq his qildim. So‘ng backend (server logikasi) qismini Node.js va Express yordamida yozdim. Bu yerda xabarlarni saqlash, foydalanuvchini autentifikatsiyadan o‘tkazish, va ma’lumotlar oqimini boshqarish kabi ishlarni yo‘lga qo‘ydim.

Shunday qilib, men bir paytning o‘zida ikkita yo‘nalishni bitta tizimda bog‘lay oldim. Har qanday o‘zgarishni darhol test qilib, xatoni o‘zim tuzata oldim. Bu meni nafaqat vaqt, balki ortiqcha muvofiqlashtirishdan ham qutqardi. Ayniqsa, test qilish va debugging jarayoni ancha tezlashdi.

Loyihani yakka o‘zim qilayotganim uchun bu yondashuv menga pul va vaqt tejashda ham foydali bo‘ldi. Hech kimdan kutmasdan, barcha kod ustidan o‘zim nazorat qildim. Bunda menga eng ko‘p yordam bergan narsa — frontend va backendda bir xil til, ya’ni JavaScript ishlatilgani bo‘ldi. Bu nafaqat kodlar o‘rtasida uyg‘unlik berdi, balki mantiqan ham boshqaruvni osonlashtirdi.

Endi kamchilik tomonlariga to‘xtaladigan bo‘lsam — to‘g‘risi, hamma narsani o‘zim qilishim ba’zida qiyin bo‘ldi. Ayniqsa, vaqtni to‘g‘ri taqsimlashda qiynalgan paytlarim bo‘ldi. Masalan, frontenddagi dizayn bilan ko‘p shug‘ullanib qolganimda, backenddagi xavfsizlikni biroz kechroq ko‘rib chiqganman. Yoki aksincha. Yana bir qiyin joyi — har bir narsani o‘zing chuqur tushunishga majbur bo‘lasan. Faqat ishlashi yetarli emas, har bir funksiya nima qiladi, nima uchun shunday ishlaydi, va bu foydalanuvchi uchun qanday natija beradi — bularning hammasini hisobga olmasang, tizimda muammolar chiqishi mumkin.

Yana shuni ham aytishim kerakki, fullstack yondashuv o‘z ichiga juda katta mas’uliyatni oladi. Har bir kichik o‘zgarish butun tizimga ta’sir qilishi mumkin. Shu bois, men kod yozayotganimda ham, foydalanuvchi qanday ko‘radi, admin qanday boshqaradi, serverda nima bo‘ladi — bularning har birini oldindan tasavvur qilib harakat qildim.

Kengroq mulohaza qilib aytadigan bo‘lsam, fullstack yondashuv kichik loyihalar uchun ayni muddao. Ayniqsa, bitta odam yoki kichik jamoa ishlayotgan bo‘lsa, bu yondashuv juda foydali. Men yaratgan platforma real ehtiyojlarga mos keldi: foydalanuvchi hisob yaratadi, so‘rov yuboradi, adminlar esa ularni boshqaradi. Kelajakda bu tizimni kengaytirish, masalan, MongoDB bazasiga o‘tish, yoki autentifikatsiyani kuchaytirish ham juda oson bo‘ladi. Chunki asos barqaror va modulli tuzilgan.

Xulosa qilib aytsam, men fullstack yondashuv orqali nafaqat texnik jihatdan, balki tashkiliy va amaliy jihatdan ham foydali tajriba oldim. Har bir bosqichda tizimning qanday ishlashini o‘zim nazorat qildim, har bir xatolikni o‘z vaqtida tuzatdim, va eng muhimi — foydalanuvchi uchun haqiqatan qulay tizim yarata oldim.

**FULLSTACK YONDASHUVNING ZAMONAVIY IT SANOATIDAGI AHAMIYATI VA AMALIY QO‘LLANISHI**

Men fullstack yondashuvda ishlashga kirishganimda, bu faqat kichik loyihalar uchun emas, balki butun dunyo bo‘yicha IT kompaniyalarida eng ko‘p ishlatiladigan yondashuvlardan biri ekanini tushundim. Ayniqsa, bugungi sanoat muhitida tezlik, samaradorlik va moslashuvchanlik asosiy mezonlarga aylangan bir paytda, fullstack dasturchilar ko‘plab loyihalarda asosiy rollarni bajarishyapti.

Masalan, **startap kompaniyalar** dastlabki versiyalarni ishlab chiqishda byudjetni tejash uchun fullstack dasturchini yollashadi. Chunki u bitta o‘zi butun tizimni – frontenddan tortib backendgacha – ishlab chiqib bera oladi. Bu ularga ortiqcha jamoa tuzmasdan tezda tayyor mahsulot chiqarish imkonini beradi. Men bu tajribani o‘zim ham boshdan kechirdim: Dern-Support kompaniyasi uchun ishlab chiqqan platformamda butun loyihani yakka o‘zim bajarishga to‘g‘ri keldi, va aynan fullstack yondashuv buning uddasidan chiqishimga yordam berdi.

Katta IT kompaniyalar ham, masalan, **Netflix**, **Airbnb**, **Uber** kabi texnologik gigantlar, fullstack yondashuvdan keng foydalanishadi. Ular o‘z ilovalari uchun **MERN stack** yoki **MEAN stack** kabi to‘plamlarni tanlashadi, bu degani — bir xil texnologiyalar asosida barcha qatlamlar boshqariladi: MongoDB (bazaga), Express (server), React yoki Angular (interfeys), va Node.js (backend). Bu arxitektura ularga kodni qayta ishlashda moslashuvchanlik, test qilishda aniqlik va bir xil texnologik muhitni yaratadi.

Masalan, Netflix platformasi orqali foydalanuvchilar film tomosha qilganda ularning tanlovi frontendda React orqali ko‘rsatiladi, bu tanlov backend orqali tahlil qilinadi va keyingi tavsiyalar avtomatik tarzda chiqariladi. Bu bir qarashda oddiy ko‘ringan jarayon aslida backend va frontendning uzviy ishlashi orqali amalga oshadi.

Men o‘zimning loyiha jarayonimda bu yondashuvning qulayligini yaqqol his qildim. Foydalanuvchi interfeysi (kirish, ro‘yxatdan o‘tish, xabar yuborish) bilan backenddagi funksiyalar (xabarlarni saqlash, foydalanuvchi ma’lumotlarini qayta ishlash) birgalikda ishlashi orqali muammosiz tizim yaratdim. Bu nafaqat ishlab chiqish bosqichida, balki keyinchalik test, takomillashtirish va kengaytirish paytida ham katta yordam berdi.

Shuni aytishim mumkinki bugungi sanoatda fullstack dasturlash universal yondashuv sifatida qabul qilinadi. U dasturchiga kodning har ikki tomonini boshqarish, loyihani to‘liq nazorat qilish va foydalanuvchining ehtiyojiga mos, ishlaydigan tizimlar yaratish imkonini beradi. Aynan shu sababli men ham bu yondashuvni tanladim va real ehtiyojlar asosida amalda sinab ko‘rdim.

**TECHREPAIRSERVICE LOYIHASIDA FULLSTACK YONDASHUVNING AMALIY QO‘LLANILISHI: TAJRIBA, AFZALLIKLAR VA SINOVLAR**

Men o‘z loyihamni – **TechRepairService** deb nomlagan veb-platformani ishlab chiqishda **fullstack yondashuvni** tanladim. Bu qarorim tasodifiy emas edi, balki real ehtiyojlar asosida kelib chiqqan edi. Loyihaning maqsadi kompyuter ta’mirlash xizmatlarini raqamlashtirish, mijozlar bilan bog‘lanishni osonlashtirish va administratorlarga xizmat so‘rovlarini boshqarish imkonini berish edi.

**Afzalliklar:**

1. **Bitta tizimda hamma narsa:**  
   Fullstack yondashuv menga bitta tizim ichida foydalanuvchi interfeysi (frontend) va server logikasini (backend) birgalikda ishlab chiqish imkonini berdi. Masalan, foydalanuvchi xabar yuborganida bu xabarni qanday qabul qilish, qayerga yozish va qachon adminga ko‘rsatish kerakligini to‘liq o‘zim belgilay oldim.
2. **Vaqt va resurs jihatdan tejamli:**  
   Men frontend va backendda alohida jamoa yollamasdan, barchasini o‘zim bajardim. Bu menga ishlab chiqish vaqtini qisqartirishga, sinovlarni tezroq o‘tkazishga va xatolarni tez aniqlashga yordam berdi.
3. **Texnologik muvofiqlik:**  
   Frontend va backendda JavaScriptdan foydalanganim sababli, kodlar o‘zaro yaxshi bog‘landi. Express.js va Tailwind CSS kombinatsiyasi esa menga kerakli funksiyalarni soddalik bilan yozish imkonini berdi.
4. **Moslashuvchanlik va kengayish imkoniyati:**  
   Loyihani ishlab chiqqan kodlarim shunday tuzilgan ediki, agar kelajakda MongoDB yoki PostgreSQL bazasiga o‘tish zarurati tug‘ilsa, backend strukturasini o‘zgartirishga hojat qolmaydi. Bu menga professional darajada o‘ylab kod yozishni o‘rgatdi.

**Kamchiliklar:**

1. **Ko‘p vazifa va yuklama:**  
   Men ham frontend, ham backendga javobgar bo‘lganim uchun, ba’zida qaysi biriga ko‘proq e’tibor qaratishni tanlay olmay qiynaldim. Dizayn ustida ishlayotganimda serverdagi autentifikatsiya yoki validatsiya masalalarini kechiktirib yuborgan holatlarim ham bo‘ldi.
2. **Tajriba talab qiladi:**  
   Har bir texnologiyani alohida o‘rganish va uni to‘g‘ri ishlatish ko‘p vaqt va bilim talab qildi. Dastlabki sinovlarda xatoliklar bo‘lgan, ba’zilarini tuzatishim uchun qo‘shimcha o‘rganishga to‘g‘ri kelgan.
3. **Xavfsizlikni mustahkamlash qiyinlashadi:**  
   Foydalanuvchi ma’lumotlarini to‘g‘ri saqlash, kirishni cheklash, admin panelga ruxsatsiz kirishni oldini olish – bu kabi xavfsizlik masalalariga alohida e’tibor berishga to‘g‘ri keldi. Bu qismlar anchagina mas’uliyat talab qildi.

Fullstack yondashuv menga faqat kod yozish emas, balki butun tizimni boshqarish, foydalanuvchi ehtiyojini chuqur tushunish, ish jarayonini boshidan oxirigacha o‘zim nazorat qilish imkonini berdi. Shu sababli, bu yondashuv faqat texnik vosita emas, balki strategik qaror ham bo‘ldi.

Men shuni angladimki, fullstack yondashuv faqat kichik loyihalar uchun emas — u katta kompaniyalar, real biznes ehtiyojlari, mijoz tajribasi va xizmat sifati uchun ham juda foydali. Shuningdek, bu yondashuv dasturchini ko‘proq mustaqil, mas’uliyatli va tizimni boshidan o‘zlashtirishga undaydi.

**2-TOPSHIRIQ**

**Mijoz ehtiyojlariga asoslangan texnik yechim**

Loyihani boshlashimdan oldin, mijozning asosiy talablarini diqqat bilan o‘rganib chiqdim. TechRepairService nomli IT texnik xizmat ko‘rsatish kompaniyasining ehtiyojlari aniq edi: foydalanuvchilarga xizmatga yozilish, xabar yuborish, narx so‘rash, va shuningdek kompaniya uchun kundalik ishlarni boshqarish, xodimlar va foydalanuvchilar o‘rtasidagi muloqotni avtomatlashtirish kerak edi. Shuni hisobga olgan holda, men qulay, xavfsiz va foydalanuvchi uchun sodda interfeysga ega bo‘lgan to‘liq stek (fullstack) yechim taklif qildim.

**Biznes konteksti va muammo mazmuni:**

TechRepairService doimiy ravishda mijozlar soni oshib borayotgan kichik biznes bo‘lib, asosiy muammo – xizmatga yozilish, buyurtma berish va xizmat holatini kuzatishning qo‘lda amalga oshirilayotganligidir. Bu jarayon vaqtni oladi, chalkashliklarga sabab bo‘ladi va xizmat sifatini pasaytiradi. Shu sababli, mijozlar bilan raqamli shaklda aloqaga chiqadigan web-platforma ishlab chiqish zarur edi.

**Taklif etilayotgan yechim tavsifi:**

Men ishlab chiqqan yechim – bu foydalanuvchilar uchun xizmatlarga yozilish, aloqa formasi orqali muammo yuborish, foydalanuvchi hisobi yaratish va admin panel orqali xabarlarni boshqarish imkonini beruvchi fullstack veb-ilova. Frontend HTML, CSS, JS orqali yozilgan; backend esa Node.js va Express.js orqali boshqariladi. Ma’lumotlar user.json fayliga yoziladi. Har bir modul mustaqil ishlaydi, ammo umumiy tizimga uyg‘unlashtirilgan.

**Funktsional talablar:**

* Mijoz hisob yaratishi va xizmatga yozilishi mumkin
* Xabar yuborish formasi ishlaydi va backendga ulanadi
* Admin panel mavjud, u orqali foydalanuvchi xabarlarini ko‘rish va o‘chirish mumkin
* Har bir foydalanuvchi uchun to‘liq ma’lumot saqlanadi (fullName, email, phone, message)
* Ma’lumotlar JSON formatida saqlanadi va yangilanadi

**Funktsional bo‘lmagan talablar:**

* Sahifa mobil qurilmalarda ham moslashgan (responsive)
* Tizim xavfsizligi oddiy tekshiruv bilan ta’minlangan
* Brauzerlararo moslik tekshirildi
* Dizayn professional va foydalanuvchi uchun tushunarli

**KPI (asosiy samaradorlik ko‘rsatkichlari):**

* Har oy xizmatga yozilayotgan foydalanuvchilar soni
* Foydalanuvchi formani to‘g‘ri to‘ldirishi va yuborishi (validatsiya muvaffaqiyati)
* Admin tomonidan o‘qilgan va o‘chirilgan xabarlar soni
* Xizmat ko‘rsatish vaqtida kechikish holatlarining kamayishi

**Tavakkalchiliklar va oqibatlar:**

* JSON fayli asosidagi bazaga haddan ortiq yuklama bo‘lsa, tormozlik ehtimoli mavjud
* Agar foydalanuvchi formani noto‘g‘ri to‘ldirsa va validatsiya ishlamasa, bu tizimga noto‘g‘ri ma’lumot kirishiga olib keladi
* Xavfsizlik kuchli parolsiz tizimlarda buzilish xavfini oshiradi

Shu asosda men mijozga yuqoridagi yechimni taklif qildim. Taklifdan oldin albatta foydalanuvchi tajribasi, texnik talablar, xavfsizlik va ishlash tezligi kabi jihatlarni hisobga oldim. Har bir funksiyani ishlab chiqishdan oldin uni qog‘ozda chizdim, algoritm va oqimlarni oldindan rejalashtirdim. Yechimim mijozga nafaqat funksional, balki estetik va foydalanuvchiga qulay tizimni taklif qila oldi.

Loyihamni aynan mijozga taklif qilganimda ular rozi bo‘lishdi, sababi taklif aniq, moslashtirilgan va real muammolarni yechishga qaratilgan edi. Bu bosqichda ishlab chiqqan hujjatlarim yechim sifatini ko‘rsatib bera oldi.