**O‘ZBEKISTON RESPUBLIKASI RAQAMLI TEXNOLOGIYALAR VAZIRLIGI**

**MUHAMMAD AL-XORAZMIY NOMIDAGI TOSHKENT AXBOROT TEXNOLOGIYALARI UNIVERSITETI SAMARQAND FILIALI**

**KOMPYUTER INJINIRING FAKULTETI**

**«DASTURIY INJINIRING» KAFEDRASI**

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

**boʻyicha bakalavr darajasini olish uchun**

**BITIRUV MALAKAVIY ISHI**

“Himoyaga ruxsat etildi”

“Dasturiy injiniring” kafedrasi mudiri \_\_\_\_\_ dots. \_\_\_\_\_\_\_\_\_\_\_\_\_.

2025 yil “\_\_\_”\_\_\_\_\_\_\_\_\_\_\_\_\_

**TIPOGRAFIYA VA REKLAMA XIZMATLARINI TAQDIM**

**QILUVCHI ONLAYN PLATFORMASINI ISHLAB CHIGISH**

Bitiruvchi: \_\_\_\_\_\_\_\_\_\_\_\_\_ Abdushukurov D. T.

Ilmiy rаhbаr: \_\_\_\_\_\_\_\_\_\_\_\_\_ Xakimov J.

**S A M A R Q A N D – 2025**

**MUNDARIJA**

[**KIRISH** 3](#_Toc197590714)

[**I BOB. TIPOGRAFIYA VA REKLAMA XIZMATLARI BOZORI: HOZIRGI HOLAT VA TENDENSIYALAR** 7](#_Toc197590715)

[1.1. Tipografiya va reklama xizmatlari bozoriga umumiy nazar 7](#_Toc197590716)

[1.2. Raqamli transformatsiya va onlayn platformalarning o‘rni 9](#_Toc197590717)

[**II BOB. ONLAYN PLATFORMA ISHLAB CHIQISH UCHUN ZAMONAVIY TEXNALOGIYALAR** 12](#_Toc197590718)

[2.1. Zamonaviy veb dasturlash texnalogiyalar 12](#_Toc197590719)

[2.2. JavaScriptga mo‘ljallangan freymvorklar 16](#_Toc197590720)

[2.3. React framevorki va zamonaviy UI komponentalaridan foydalanish 21](#_Toc197590721)

[**III BOB. ONLAYN PLATFORMASINI ISHLAB CHIQISH BOSQICHLARI** 34](#_Toc197590722)

[3.1. Dasturiy ta’minotni loyihalashtirish 34](#_Toc197590723)

[3.2. Veb ilovaning dizayn qismini loyhalash 35](#_Toc197590724)

[**XULOSA** 38](#_Toc197590725)

[**FOYDALANILGAN ADABIYOTLAR RO`YXATI** 40](#_Toc197590726)

[**ILOVALAR** 43](#_Toc197590727)

[**WEB SAYT KO’RINISHI** 74](#_Toc197590731)

# **KIRISH**

Hozirgi tez sur’atlarda rivojlanayotgan raqamli asrda axborot texnologiyalarining hayotimizdagi o‘rni tobora ortib bormoqda. Xususan, biznes jarayonlar, xizmat ko‘rsatish sohasi, jumladan, tipografiya va reklama xizmatlari ham raqamli transformatsiyaga moslashmoqda. An’anaviy tipografik xizmatlarning zamonaviy talablarga javob berishi, mijozlar bilan masofadan ishlash, tezkor aloqa o‘rnatish va shaffof buyurtma tizimini yaratish zarurati bugungi kunda dolzarb masalalardan biri hisoblanadi. Ayniqsa, onlayn platformalar orqali ushbu xizmatlarni ko‘rsatishning afzalliklari keng miqyosda e’tirof etilmoqda.

O‘zbekiston Respublikasi Prezidentining raqamli iqtisodiyot va innovatsion rivojlanish sohasida olib borayotgan islohotlari, davlat dasturlari doirasida axborot-kommunikatsiya texnologiyalari infratuzilmasining modernizatsiyasi ushbu yo‘nalishda ilmiy va amaliy loyihalarni amalga oshirish uchun qulay muhit yaratmoqda. Mazkur bitiruv malakaviy ishi aynan shu ijtimoiy-iqtisodiy ehtiyojlar va texnologik taraqqiyotga asoslangan holda ishlab chiqilgan bo‘lib, u tipografiya va reklama xizmatlarini yagona onlayn platforma orqali taqdim etish imkoniyatlarini chuqur tahlil qilish va amaliy yechim taklif qilishga qaratilgan [3, b. 50].

Shuni ta’kidlash kerakki, so‘nggi yillarda onlayn xizmatlarga bo‘lgan talab keskin ortdi. Mijozlar xizmatlar va mahsulotlarga tez, sifatli va ishonchli tarzda ega bo‘lishni istaydilar. Tipografik mahsulotlarni – vizitkalar, bukletlar, bannerlar, reklama materiallari va boshqa poligrafik mahsulotlarni – onlayn tarzda buyurtma qilish, ularni shaxsiylashtirish va to‘lov tizimlarini integratsiya qilish orqali xizmat ko‘rsatishni soddalashtirish mumkin. Bu esa raqobatbardoshlikni oshiradi, ish unumdorligini kuchaytiradi va xizmat sifatini yangi bosqichga olib chiqadi.

**Bitiruv malakaviy ishining dolzarbligi** – zamonaviy texnologiyalar yordamida tipografiya va reklama xizmatlarini raqamlashtirish, ya’ni ularni yagona veb-platforma orqali boshqarish va taqdim etish imkoniyatlarini o‘rganishdir. Bu orqali mijozlarga qulay, intuitiv interfeys, avtomatlashtirilgan buyurtma va dizayn tizimi, real vaqt monitoringi va interaktiv muloqot imkoniyatlari taqdim etiladi. Shuningdek, ushbu platforma kichik va o‘rta biznes vakillari uchun ham samarali yechim sifatida xizmat qilishi mumkin.

**Bitiruv malakaviy ishining asosiy maqsadi** – zamonaviy dasturlash texnologiyalari asosida tipografiya va reklama xizmatlarini taqdim etuvchi funksional, interaktiv va foydalanuvchi uchun qulay onlayn platformani ishlab chiqishdir. Bu maqsadga erishish uchun quyidagi asosiy vazifalar belgilangan:

* Tipografiya va reklama xizmatlari bozorining hozirgi holatini o‘rganish;
* Raqamli transformatsiya va onlayn xizmatlarning bu sohadagi o‘rnini tahlil qilish;
* Zamonaviy veb-dasturlash texnologiyalari, ayniqsa JavaScript freymvorklari va React kutubxonasining imkoniyatlarini tadqiq qilish;
* Onlayn platformani loyihalash, interfeys dizaynini ishlab chiqish, foydalanuvchi tajribasini (UX/UI) optimallashtirish;
* Tayyorlangan platformaning samaradorligini baholash, foydalanuvchi ehtiyojlariga mosligini tahlil qilish.

Tipografiya xizmatlari – bu grafik dizayn, bosma materiallar tayyorlash, banner va vizitkalar yaratish kabi keng ko‘lamli faoliyatni o‘z ichiga oladi. Shu bilan birga, reklama xizmatlari ham mahsulot yoki xizmatni ommaga yetkazish va iste’molchining e’tiborini tortishda muhim rol o‘ynaydi. Ammo aksariyat holatlarda mijozlar va ijrochilar o‘rtasidagi aloqa, narx kelishuvi, fayl almashinuvi kabi jarayonlar offline tarzda yoki ijtimoiy tarmoqlar orqali amalga oshiriladi, bu esa tizimli yondashuv va sifatni pasaytiradi.

Ushbu bitiruv malakaviy ishda tipografiya va reklama xizmatlarini yagona platforma orqali taqdim etishga imkon beruvchi onlayn xizmatlar platformasini ishlab chiqish masalasi ko‘rib chiqiladi. Platformaning asosiy vazifasi – mijozlarga qulay interfeys orqali buyurtma berish, dizayn fayllarini yuklash, narxlarni ko‘rish, ijrochilar bilan bog‘lanish va yakuniy mahsulotni olish imkoniyatini yaratishdan iborat.

**Ishning maqsadi** – foydalanuvchi uchun qulay, ishonchli va samarali ishlovchi onlayn platforma ishlab chiqish orqali reklama va tipografiya sohasidagi xizmatlar ko‘rsatishni raqamlashtirish hamda avtomatlashtirishdir.

**Tadqiqot obyekti** sifatida tipografiya va reklama xizmatlarini ko‘rsatish jarayonlari tanlangan bo‘lsa, predmeti esa mazkur jarayonlarni raqamli platforma orqali amalga oshirish texnologiyalari hisoblanadi.

**Ishning vazifalari:**

Ushbu maqsadga erishish uchun quyidagi **ilmiy-amaliy vazifalar**ni hal etish belgilangan:

1. Tipografiya va reklama xizmatlari bozorining hozirgi holatini o‘rganish va raqamli transformatsiyadagi o‘rnini tahlil qilish;
2. Onlayn xizmatlar infratuzilmasini yaratishda qo‘llaniladigan zamonaviy dasturlash texnologiyalarini o‘rganish;
3. ReactJS va unga tegishli UI kutubxonalarini chuqur tahlil qilish;
4. Foydalanuvchi interfeysini (UI) va foydalanuvchi tajribasini (UX) hisobga olgan holda platforma dizaynini ishlab chiqish;
5. Loyihaning arxitekturasini ishlab chiqish, ma’lumotlar oqimini (flow) modellashtirish;
6. Platformani testlash, ishonchliligini va funksionalligini baholash;
7. Amaliy taklif va tavsiyalar ishlab chiqish.

**Ishning metodologiyasi**

Tadqiqotda quyidagi metodologik yondashuvlar qo‘llanilgan:

* Tahliliy va solishtirma usullar yordamida bozordagi mavjud yechimlar o‘rganildi;
* Ilmiy adabiyotlar, maqolalar va onlayn resurslar asosida nazariy asoslar yaratildi;
* Amaliy loyihalashtirish jarayonlarida iterativ va komponental yondashuvlar qo‘llanildi;
* UX/UI dizayn tamoyillari asosida foydalanuvchi qulayligi tahlil qilindi;
* Frontend va backend dasturlash vositalari orqali prototip yaratildi.

**Ishning amaliy ahamiyati**

Ushbu ishda ishlab chiqilgan platforma konsepsiyasi real biznes loyihalarida, ayniqsa kichik va o‘rta tipografik xizmat ko‘rsatish korxonalarida amaliyotga joriy etilishi mumkin. U mijozlar bilan ishlash tizimini soddalashtiradi, xizmatlar portfelini onlayn muhitda kengaytiradi, buyurtmalarni avtomatlashtiradi va xizmat ko‘rsatish tezligini oshiradi.

**Ishning ilmiy yangiligi shundan iboratki**, mavjud texnologik echimlar asosida bir nechta xizmat turlarini yagona onlayn maydonda to‘plash va ularni boshqarish imkonini beruvchi amaliy tizim yaratiladi. Ushbu tizim foydalanuvchilarga xizmatlarni tez, aniq va ishonchli shaklda olish imkonini beradi.

Ushbu ishda loyiha tahlili, texnik talablarni aniqlash, dasturiy echim tanlash, foydalanuvchi interfeysi yaratish va tizimni testdan o‘tkazish bosqichlari amalga oshiriladi. Natijada reklama sohasidagi xizmatlarni yangi bosqichga olib chiqish mumkin bo‘ladi.

**Bitiruv malakaviy ishi** - kirish, uchta asosiy bob, xulosa, foydalanilgan adabiyotlar va ilovalardan iborat. Birinchi bobda tipografiya va reklama xizmatlari bozorining hozirgi holati va raqamli tendensiyalar ko‘rib chiqiladi. Ikkinchi bobda esa zamonaviy veb-dasturlash texnologiyalari, xususan React freymvorki asosida platformani ishlab chiqish jarayonlari yoritiladi. Uchinchi bobda onlayn platformani ishlab chiqish bosqichlari va dizayn, testlash, foydalanuvchi interfeysi kabi jihatlar tahlil qilinadi. Ishning yakunida tadqiqot natijalari umumlashtirilib, xulosalar chiqariladi va kelajakdagi takliflar beriladi.

# **I BOB. TIPOGRAFIYA VA REKLAMA XIZMATLARI BOZORI: HOZIRGI HOLAT VA TENDENSIYALAR**

* 1. **Tipografiya va reklama xizmatlari bozoriga umumiy nazar**

So‘nggi yillarda dunyo bo‘yicha, xususan, O‘zbekiston reklama va tipografiya xizmatlari bozori jadal rivojlanmoqda. Yangi texnologiyalar, mobil qurilmalar kengayishi va internetga kirish imkoniyatlarining kengayishi ushbu sohada katta o‘zgarishlarga olib keldi. Reklama va tipografiya xizmatlari, aslida, jamiyatning iqtisodiy va madaniy rivojida muhim rol o‘ynaydi. Ular mahsulot va xizmatlarni targ‘ib qilish, kompaniya brendini shakllantirish va auditoriya bilan samarali aloqa o‘rnatishda vositachi bo‘lib xizmat qiladi [8, b. 347].

Tipografiya xizmatlari: mohiyati va turlari

Tipografiya – bu matn va grafik elementlarni vizual tarzda loyihalash va joylashtirish san’atidir. Bugungi kunda tipografiya xizmatlari nafaqat bosma materiallar (vizitkalar, bukletlar, kataloglar, afishalar) tayyorlashni, balki raqamli dizayn (sotsial tarmoq postlari, veb-bannerlar, e-mail marketing uchun vizuallar) yaratishni ham o‘z ichiga oladi. Tipografiya xizmatlariga quyidagilar kiradi:

* Grafik dizayn (logotip, firma uslubi, bannerlar)
* Bosma mahsulotlar dizayni (vizitkalar, bukletlar, posterlar)
* Brending elementlari ishlab chiqish
* Matbaa uchun tayyorlash (pre-press)
* Raqamli materiallar dizayni

Zamonaviy bosma mahsulotlar ko‘plab sohalarda – ta’lim, sog‘liqni saqlash, savdo, xizmat ko‘rsatish va boshqa yo‘nalishlarda keng qo‘llaniladi. Har bir korxona yoki tashkilot o‘zining korporativ uslubini ifoda etuvchi dizayn vositalariga ehtiyoj sezadi.

Reklama (lotincha: reklamo – “qichqirmoq“) — tovarlarning sifati, ularni sotib olishdan koʻriladigan naf haqidagi axborot; bevosita yoki bilvosita foyda (daromad) olish maqsadida yuridik va jismoniy shaxslar yoki mahsulot toʻgʻrisida tarqatiladigan maxsus axborot. Reklama marketingning tarkibiy qismi hisoblanib, talab va isteʼmolchilar didining shakllanishiga taʼsir koʻrsatadi. Reklama korxona imijini yaratish (uzoq davrli taʼsir koʻrsatish), joriy sotish hajmini koʻpaytirish (qisqa davrli taʼsir koʻrsatish), haridorlarga nima sotib olishni va uni nima uchun sotib olish zarurligini hal qilishda yordamlashish va boshqa maqsadlarda olib boriladi. Reklamaning asosiy vazifasi — xaridorni qiziqtirish, mijoz xaridorlarni yoʻqotmaslik va ularni kengaytirishga qaratiladi. (1)

Reklama xizmatlari: vazifalari va evolyutsiyasi

Reklama xizmatlari iste’molchilarga mahsulot yoki xizmat haqida ma’lumot yetkazish, ularning qaror qabul qilish jarayoniga ta’sir o‘tkazish va brendga sodiqlikni shakllantirishga qaratilgan. Ilgari reklama asosan gazeta, radio, televizor orqali tarqatilgan bo‘lsa, hozirda internet reklamasi asosiy kuchga aylanmoqda.

Reklama xizmatlariga quyidagilar kiradi:

* Bosma reklama (gazeta, jurnallar, varaqalar)
* Tashqi reklama (billboard, banner, ekranlar)
* Televizion va radio reklama
* Raqamli reklama (sotsial tarmoq, Google reklama, influencer marketing)
* PR xizmatlari va kontent marketing

Reklama bozorida bugungi kunda eng katta o‘sishni digital (raqamli) reklama yo‘nalishi ko‘rsatmoqda. Ayniqsa, TikTok, Instagram, Telegram kabi platformalardagi maqsadli reklama kampaniyalari katta samaradorlik beradi. Bu esa bizneslarni o‘z marketing strategiyalarini qayta ko‘rib chiqishga undaydi [11, b. 267-275].

## **1.2. Raqamli transformatsiya va onlayn platformalarning o‘rni**

Zamonaviy davrda raqamli transformatsiya barcha sohalar, xususan, marketing, dizayn va reklama xizmatlari uchun tub o‘zgarishlarni olib keldi. Raqamli texnologiyalarning jadal rivojlanishi natijasida biznes yuritish uslublari, iste’molchilar bilan muloqot qilish usullari va xizmat ko‘rsatish formati sezilarli darajada o‘zgarib bormoqda.

Raqamli transformatsiyaning mohiyati

Raqamli transformatsiya (Digital Transformation) — bu texnologiyalarni joriy etish orqali biznes jarayonlarini, mahsulot va xizmatlarni takomillashtirish, yangi imkoniyatlar yaratish va mijozlar ehtiyojini samarali qondirish jarayonidir. Bu nafaqat texnik yangiliklarni qo‘llash, balki butun tashkilot madaniyatini o‘zgartirishni ham talab qiladi [9, b. 67].

Raqamli transformatsiyaning asosiy yo‘nalishlariga quyidagilar kiradi:

Avtomatlashtirish va sun’iy intellekt: murakkab ishlarni avtomatlashtirish orqali samaradorlik oshiriladi.

Bulutli texnologiyalar: ma’lumotlar saqlanishi, xizmatlar va ilovalar istalgan joydan boshqarilishi mumkin.

Mobil texnologiyalar: foydalanuvchilarning istalgan vaqtda xizmatlardan foydalanish imkoniyati.

Ma’lumotlarga asoslangan qarorlar qabul qilish: analitika va monitoring orqali aniq strategiyalar tuziladi.

Marketing va dizayn sohalarida raqamli transformatsiya ta’siri

Marketing va dizayn sohalari raqamli texnologiyalardan eng ko‘p ta’sirlangan yo‘nalishlardandir. An’anaviy reklama vositalari o‘rnini bosgan raqamli reklamalar, sun’iy intellekt asosidagi dizayn vositalari va onlayn platformalar bugungi kundagi asosiy trendga aylangan.

1. Internet reklamalari:

Google Ads, Facebook Ads, TikTok Ads kabi tizimlar maqsadli auditoriyaga qisqa vaqt ichida yetib borishni ta’minlaydi. Bu esa reklama byudjetining samarali sarflanishiga xizmat qiladi.

2. Onlayn dizayn vositalari:

Canva, Figma, Adobe Express kabi vositalar dizayn ishlarini tez va soddalashtirilgan shaklda amalga oshirish imkonini beradi. Bu esa grafik dizayn xizmatlarini ommaviylashtirmoqda.

3. Sun’iy intellekt yordamida dizayn:

Midjourney, DALL·E, Adobe Firefly kabi tizimlar sun’iy intellekt asosida dizayn yaratishni avtomatlashtirib, ijodiy jarayonni yangi bosqichga olib chiqmoqda.

Onlayn platformalarning afzalliklari va rolini tahlil qilish

Raqamli transformatsiya jarayonida onlayn platformalar muhim o‘ringa ega. Ular turli xizmatlarni bir joyda to‘plash, foydalanuvchilarga qulay interfeys orqali xizmatlar bilan tanishish, buyurtma berish, ijrochilarni tanlash va to‘lovni amalga oshirish imkonini beradi [12, b. 234].

Onlayn platformalarning afzalliklari:

Shaffoflik va ishonchlilik

– Har bir ijrochining profili, reytingi va sharhlari mavjud bo‘lishi mijozlarga tanlovda yordam beradi.

– Platforma orqali amalga oshirilgan to‘lovlar ishonchliroq bo‘ladi.

Vaqt va joy cheklovlarisiz ishlash

– Mijoz va ijrochi o‘rtasidagi muloqot onlayn ravishda amalga oshadi.

– Buyurtmalar har qanday vaqtda qabul qilinadi, bu esa xizmatlar doirasini kengaytiradi.

Tizimlashtirilgan boshqaruv

– Buyurtmalar, chatlar, fayllar va to‘lovlar bitta tizim orqali boshqariladi.

– Statistika, hisobotlar va tahlillar orqali platforma faoliyati doimiy ravishda monitoring qilinadi.

Yangi ish o‘rinlari va imkoniyatlar

– Frilanserlar, grafik dizaynerlar, kontent yaratuvchilari kabi ijodiy mutaxassislar uchun yangi bozorlar ochiladi.

– Har qanday yoshdagi va tajribadagi dizaynerlar platformaga qo‘shilib daromad topishi mumkin.

Dunyo va O‘zbekiston tajribasi

Xalqaro miqyosda Fiverr, Upwork, DesignCrowd kabi platformalar grafik dizayn, marketing va reklama xizmatlari uchun keng imkoniyatlar yaratdi. Ularda millionlab foydalanuvchilar faoliyat yuritadi.

O‘zbekistonda esa bu yo‘nalishda hali to‘liq shakllangan yagona tizimli platforma mavjud emas. Aksariyat dizaynerlar Telegram yoki Instagram orqali aloqada bo‘lishadi. Bu esa tartibsizlik, muammoli to‘lovlar va sifatsiz xizmatlar xavfini oshiradi.

Shu sababli, mahalliy onlayn platformani yaratish nafaqat iqtisodiy jihatdan foydali, balki xizmat ko‘rsatish madaniyatini ham oshiradi. Bunday platforma:

* Ishonchli dizaynerlar bazasini yaratadi
* Buyurtmalarni markazlashtirib, samaradorlikni oshiradi
* Mahalliy freelanserlar va dizaynerlar uchun daromad manbaiga aylanadi

# **II BOB. ONLAYN PLATFORMA ISHLAB CHIQISH UCHUN ZAMONAVIY TEXNALOGIYALAR**

## **2.1. Zamonaviy veb dasturlash texnalogiyalar**

Axborot texnologiyalari sohasidagi jadal rivojlanish natijasida veb dasturlash zamonaviy raqamli dunyoning ajralmas qismiga aylandi. Bugungi kunda korxonalar, tashkilotlar, shaxsiy loyihalar va davlat tizimlari uchun ham veb ilovalar zarur bo‘lib qolgan. Veb texnologiyalar foydalanuvchilarga qulay interfeys yaratish, ma’lumotlar almashinuvi, xavfsizlik, tezkorlik va masshtablanuvchanlik kabi talablarga javob berishi lozim. Ushbu bo‘limda veb dasturlash sohasidagi eng so‘nggi texnologiyalar, ular bilan ishlash uslublari va afzalliklari keng yoritiladi [15, b. 32].

Frontend texnologiyalari:

Frontend — bu foydalanuvchining ko‘rishi va bevosita muloqot qilishi mumkin bo‘lgan interfeys qismini bildiradi. Zamonaviy frontend texnologiyalari foydalanuvchi tajribasini oshirishga katta hissa qo‘shmoqda.

HTML5 — bu veb sahifaning asosiy karkasini yaratish uchun qo‘llaniladigan til. U avvalgi versiyalar bilan solishtirganda yanada boyroq imkoniyatlarga ega. HTML5 yordamida video, audio, forma elementlari, Canvas API va boshqa multimedia imkoniyatlar to‘g‘ridan-to‘g‘ri integratsiya qilinadi. Bundan tashqari, semantik teglari yordamida sahifani to‘g‘ri strukturalash mumkin bo‘ladi, bu esa SEO va foydalanish qulayligini oshiradi.

CSS3 — bu sahifaga estetik ko‘rinish beruvchi texnologiya bo‘lib, u orqali elementlar joylashuvi, rang, animatsiya, responsive dizayn kabi xususiyatlar boshqariladi. Flexbox va CSS Grid texnologiyalari sahifalarning mobil qurilmalarda moslashuvchan va dinamik bo‘lishini ta’minlaydi.

JavaScript va SPA arxitekturasi

JavaScript — frontendning yuragi hisoblanadi. Bu til yordamida sahifaga interaktivlik, DOM manipulyatsiyasi, foydalanuvchi voqealari bilan ishlash kabi imkoniyatlar qo‘shiladi. Bugungi kunda JavaScript asosida yaratilgan SPA (Single Page Application) arxitekturasi mashhurlikka erishgan bo‘lib, sahifalar yuklanmasdan dinamik yangilanadi. Bunday texnologiyalar foydalanuvchining ilova bilan ishlash tajribasini ancha yaxshilaydi.

Zamonaviy frontend kutubxonalar va frameworklar

* React.js — Facebook tomonidan ishlab chiqilgan komponentli arxitekturaga ega JavaScript kutubxonasi. Virtual DOM va uni qayta ishlash mexanizmi orqali tezkorlikni oshiradi.
* Vue.js — o‘rganilishi oson va engil framework bo‘lib, kichik va o‘rta loyihalar uchun qulay.
* Angular — Google tomonidan ishlab chiqilgan va katta hajmli korporativ loyihalarda keng qo‘llaniladi. Uning TypeScript asosidagi tuzilmasi mustahkam va barqaror bo‘lishini ta’minlaydi.

Backend texnologiyalari

Backend — bu foydalanuvchining ko‘ziga ko‘rinmaydigan, ammo asosiy jarayonlarni boshqaradigan qismdir. U serverda ish olib boradi, ma’lumotlar bazasi bilan muloqot qiladi, foydalanuvchi autentifikatsiyasini tekshiradi, xavfsizlikni ta’minlaydi va hokazo [16, b. 212-226].

Node.js — JavaScript asosida ishlaydigan server muhit bo‘lib, bir vaqtning o‘zida minglab so‘rovlarni qayta ishlash imkoniyatiga ega. U asinxron va voqealarga asoslangan arxitekturasi bilan mashhur. Node.js o‘zi bilan birga NPM (Node Package Manager) kutubxona tizimini taqdim etadi, bu esa turli modul va vositalarni tez integratsiya qilish imkonini beradi.

Python va uning frameworklari

Python o‘zining sodda sintaksisi va kuchli kutubxonalar to‘plami bilan mashhur. Django va Flask kabi frameworklar yordamida tezkor va xavfsiz backend ilovalar yaratish mumkin. Django "barcha narsalar tayyor" printsipiga asoslangan bo‘lsa, Flask minimalistik va moslashuvchanlikni ta’minlaydi.

PHP va Laravel

PHP — veb dasturlashda uzoq yillardan beri foydalanilayotgan backend tili. Laravel frameworki esa uning eng zamonaviy talqinidir. Laravel MVC arxitekturasi, migratsiya tizimi, autentifikatsiya moduli kabi funksiyalari orqali samarali backend ishlab chiqish imkonini beradi.

Boshqa backend texnologiyalar

* Ruby on Rails — "konventsiyaga ko‘ra konfiguratsiya" printsipiga amal qiluvchi framework bo‘lib, tez rivojlantirish imkonini beradi.
* ASP.NET Core — Microsoft tomonidan yaratilgan backend platformasi bo‘lib, xavfsizlik va ishlash tezligi bilan ajralib turadi.
* Go (Golang) — yuqori samaradorlik va parallellikni ta’minlashda ustun bo‘lgan til. Ko‘p hollarda real vaqtli servislar uchun ishlatiladi.

Ma’lumotlar bazasi va API texnologiyalari

Ma’lumotlar bazalari

* MySQL, PostgreSQL — bu SQL (Structured Query Language) asosida ishlaydigan relatsion bazalardir. Ma’lumotlarni strukturaviy saqlash va murakkab so‘rovlarni bajarish uchun qulay.
* MongoDB, Firebase — NoSQL bazalar bo‘lib, ular hujjat asosida (document-based) ishlaydi. Bu turdagi bazalar noturg‘un yoki dinamik strukturalar bilan ishlashda qulay.

API texnologiyalari

Frontend va backend o‘rtasidagi muloqotni ta’minlash uchun API (Application Programming Interface) ishlatiladi. Hozirda keng tarqalgan API turlari:

* REST API — HTTP metodlariga (GET, POST, PUT, DELETE) asoslangan eng ommabop protokol.
* GraphQL — Facebook tomonidan ishlab chiqilgan texnologiya bo‘lib, foydalanuvchi aynan kerakli ma’lumotni so‘rashi mumkin bo‘ladi, bu esa ortiqcha trafikni kamaytiradi.

Versiya boshqaruvi va DevOps texnologiyalari

* Git — kodning turli versiyalarini boshqarish va bir nechta dasturchi bilan samarali ishlash imkonini beradi.
* GitHub, GitLab, Bitbucket — masofaviy Git platformalari bo‘lib, ular jamoaviy ish, avtomatik testlar va deployment jarayonlarini qo‘llab-quvvatlaydi.
* Docker — ilovani konteynerga joylashtirish orqali uni boshqa tizimlarda xatolarsiz ishga tushirishga yordam beradi.
* CI/CD (Continuous Integration/Continuous Deployment) — dasturiy ta’minotni avtomatlashtirilgan tarzda sinovdan o‘tkazish va yangilab borish texnologiyalaridir [19, b. 22].

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

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

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

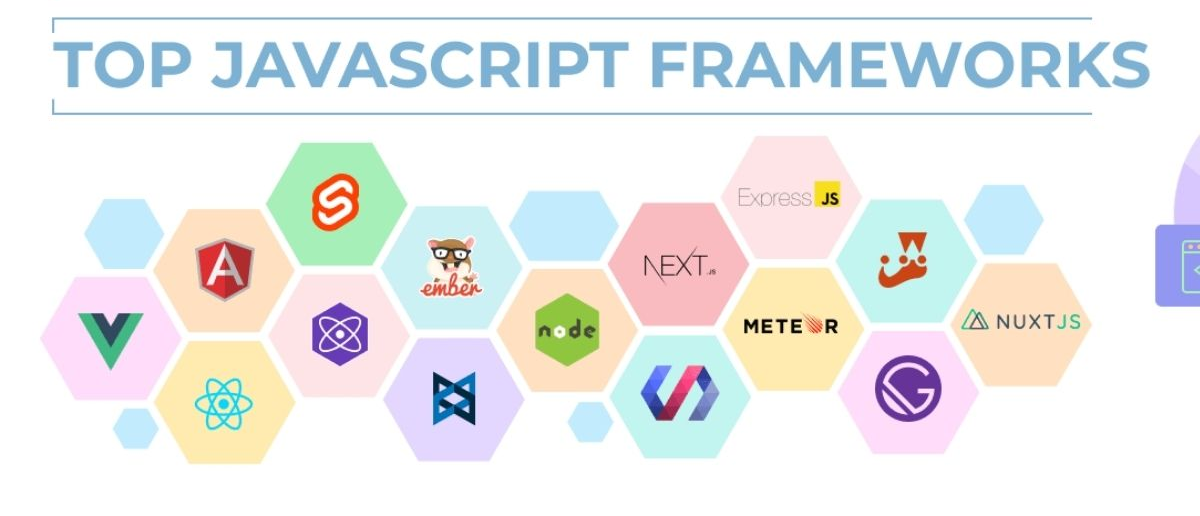
Asosiy arxitektura xususiyatlari: tiplarni dinamik aniqlash, avtomatik xotirani boshqarish, prototip dasturlash, birinchi darajali ob'ektlar sifatida funktsiyalar.

Frameworklar malum bir dasturlash tillariga asoslangandir. React, Flutter, Angular, Vue, PyTorch, Spring Boot, React Native, Apache Spark, Ionic va boshqalar eng ko‘p qo‘llaniladigan mashhur framworklardir. Bu framworklar ishlab chiquvchilarga mustahkam va boy funksiyalarga ega dasturiy ta'minotni yaratishga imkon beradi [21, b. 75-89].

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

Frameworklardan foydanalish afzalliklari quyidagilardir:

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

Ishlab chiqish jarayonini boshlashdan oldin, qaysi dasturlash tilini loyihangizga mos kelishini tanlashingiz kerak. Misol uchun, agar JavaScript-ni tanlansa, veb dastur uchun Angular, React, Next.js, Vue.js kabi JavaScript frameworklardan birini tanlab ishlatish mumkin (2.1-rasm).

*2.1-ram. Mashhur javscript framevorklari*

TypeScript (qisqacha TS) Microsoft tomonidan ishlab chiqilgan bepul va ochiq manbali yuqori darajadagi dasturlash tili boʻlib, JavaScript-ga ixtiyoriy turdagi izohlar bilan statik yozishni qoʻshadi. U katta ilovalarni ishlab chiqish va JavaScript-ga transpilyatsiya qilish uchun mo'ljallangan.

TypeScript-dan mijoz tomonida ham, server tomonida ham (Node.js, Deno yoki Bun kabi) JavaScript ilovalarini ishlab chiqishda foydalanish mumkin. Transpilatsiya uchun bir nechta variant mavjud. Standart TypeScript kompilyatoridan foydalanish mumkin yoki TypeScript-ni JavaScript-ga aylantirish uchun Babel kompilyatoridan foydalanish mumkin [23, b. 112-128].

TypeScript mavjud JavaScript kutubxonalarining turdagi ma'lumotlarini o'z ichiga olishi mumkin bo'lgan aniqlovchi fayllarni qo'llab-quvvatlaydi, xuddi C++ sarlavha fayllari mavjud ob'ekt fayllari tuzilishini tavsiflashi mumkin. Bu boshqa dasturlarga fayllarda belgilangan qiymatlardan xuddi statik tarzda yozilgan TypeScript ob'ektlari kabi foydalanish imkonini beradi. JQuery, MongoDB va D3.js kabi mashhur kutubxonalar uchun uchinchi tomon sarlavha fayllari mavjud. Node.js kutubxona modullari uchun TypeScript sarlavhalari ham mavjud boʻlib, ular TypeScript ichida Node.js dasturlarini ishlab chiqish imkonini beradi.

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

1. React (ba'zan React.js yoki ReactJS ) foydalanuvchi interfeyslarini ishlab chiqish uchun ochiq manbali JavaScript kutubxonasi.

React Facebook, Instagram va individual ishlab chiquvchilar va korporatsiyalar hamjamiyati tomonidan ishlab chiqilgan va qoʻllab-quvvatlanadi.

React-dan bir sahifali saytlar va mobil ilovalarni ishlab chiqish uchun foydalanish mumkin. Uning maqsadi yuqori ishlash tezligi, soddaligi va keng imkoniyatga ega ekanligidir. Foydalanuvchi interfeyslarini ishlab chiqish uchun kutubxona sifatida React ko'pincha MobX, Redux va GraphQL kabi boshqa kutubxonalar bilan birga ishlatiladi.

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

2. Angular (shuningdek, Angular 2+ deb ham ataladi) bu TypeScript-ga asoslangan bepul va ochiq manbali bir sahifali veb-ilovalar framevorki. U Google tomonidan hamda jismoniy shaxslar va korporatsiyalar hamjamiyati tomonidan ishlab chiqilgan. Angular - bu AngularJS-ni yaratgan jamoaning to'liq qayta yozishi. Angular ekotizimi 1,7 milliondan ortiq dasturchilar, kutubxona mualliflari va kontent yaratuvchilardan iborat turli guruhdan iborat. Stack Overflow Developer Survey ma'lumotlariga ko'ra, Angular eng ko'p ishlatiladigan veb-ramkalardan biridir.

3. Vue.js foydalanuvchi interfeyslari va bir sahifali ilovalarni yaratish uchun ochiq manbali JavaScript asosidir. U birinchi marta 2013-yilda Evan You tomonidan chiqarilgan va shundan so'ng uning soddaligi va moslashuvchanligi tufayli mashhurlik oshgan [24, b. 50-59].

Vue.js - progressiv ramka, ya'ni uni bosqichma-bosqich qabul qilish mumkin. Bu shuni anglatadiki, u kichik, mustaqil komponentlarni, shuningdek, katta, murakkab ilovalarni yaratish uchun ishlatilishi mumkin. Shuningdek, u oʻrganish oson boʻlishi uchun yaratilgan, shuning uchun ishlab chiquvchilar u bilan tezda qurishni boshlashlari mumkin.

Vue.js ning asosiy afzalliklaridan biri uning reaktivlik tizimidir. Ushbu tizim Vue.js-ga ma'lumotlar o'zgarganda ko'rinishni avtomatik ravishda yangilash imkonini beradi. Bu shuni anglatadiki, ishlab chiquvchilar ma'lumotlar o'zgarganda ko'rinishni qo'lda yangilashlari shart emas, bu ko'p vaqt va kuchni tejash imkonini beradi.

4. Next.js server tomonidan ko'rsatilgan React ilovalarini yaratish uchun mashhur JavaScript ramkasidir. Vercel (sobiq Zeit) tomonidan ishlab chiqilgan Next.js yuqori unumli va kengaytiriladigan veb-ilovalarni yaratishni osonlashtiradigan xususiyatlar va vositalar to'plamini taqdim etadi.

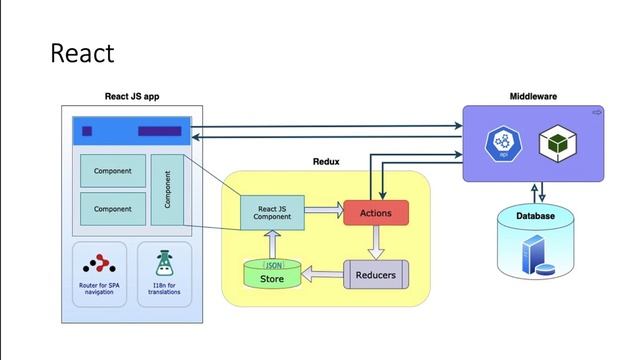
Next.js ning asosiy afzalliklaridan biri uning server tomonida renderlash (SSR) uchun o‘rnatilgan qo‘llab-quvvatlashidir. Server tomonida ko'rsatish sahifani ko'rsatish uchun mijoz tomoni JavaScript-ga tayanishdan ko'ra, serverda dastlabki HTMLni yaratish orqali veb-ilovalarning ishlashi va SEO-ni yaxshilashi mumkin. Bu, shuningdek, sekinroq internet ulanishlari yoki eski qurilmalari bo'lgan foydalanuvchilar uchun yaxshi foydalanuvchi tajribasini taqdim etishi mumkin.

Next.js shuningdek, o‘rnatilgan ishlab chiqish serverini taqdim etadi, bu esa dasturni ishlab chiqish jarayonida sinovdan o‘tkazish va disk raskadrovka qilishni osonlashtiradi. Bundan tashqari, Next.js ilovangiz ish faoliyatini optimallashtirishni osonlashtiradigan bir qator xususiyatlarni taqdim etadi, masalan, kodni ajratish va avtomatik tasvirni optimallashtirish.

Next.js shuningdek, dinamik marshrutlash uchun o'rnatilgan yordamga ega, bu ramka dasturning marshrutini avtomatik ravishda boshqaradi va har bir sahifa uchun to'g'ri URL-manzillarni yaratadi, bu bir nechta sahifali veb-ilovani yaratishni osonlashtiradi.

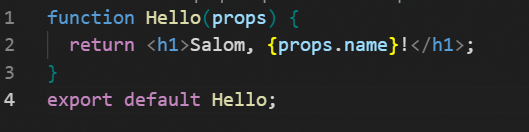
## **2.3. React framevorki va zamonaviy UI komponentalaridan foydalanish**

React.js — bu Facebook tomonidan ishlab chiqilgan ochiq manbali JavaScript kutubxonasi bo‘lib, foydalanuvchi interfeyslari (UI) yaratish uchun ishlatiladi. React komponentli arxitektura, virtual DOM, va reaktiv yangilanish tizimi orqali zamonaviy veb ilovalarda tezkor, interaktiv va samarali interfeyslar yaratish imkonini beradi. Uning asosiy maqsadi — murakkab foydalanuvchi interfeyslarini modullashtirilgan va qayta ishlatiladigan komponentlar asosida qurishdir. Uning ishlash prinsipi 2.4-rasmda keltirilgan.

**

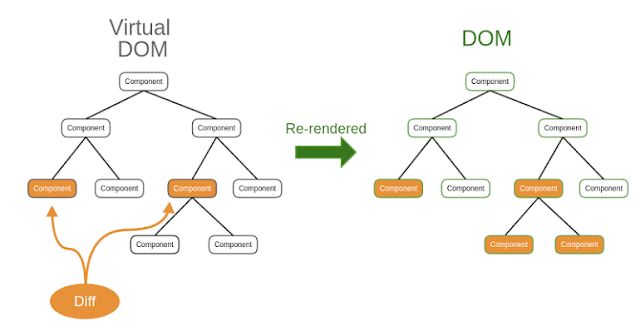
*2.2-react arvitekturasi*

Komponentli arxitektura. React ilovalari komponentlar deb ataluvchi kichik, mustaqil bo‘laklardan tashkil topadi. Har bir komponent o‘zining holatiga (state) ega va boshqa komponentlar bilan aloqada bo‘lishi mumkin. Bu komponentlar funksional yoki klassik (class-based) ko‘rinishda bo‘ladi [25, b. 75-89].



JSX (JavaScript XML). JSX — bu React’ning o‘ziga xos sintaksisi bo‘lib, u JavaScript kod ichida HTML elementlariga o‘xshash strukturalarni yozish imkonini beradi. JSX orqali kodlar yanada o‘qilishi oson va tuzilmasi aniq bo‘ladi.



Virtual DOM. React ilovalari **virtual DOM** texnologiyasidan foydalanadi. Ya’ni, dastlab butun sahifa DOM daraxtining virtual nusxasi yaratiladi. Keyin foydalanuvchi bilan o‘zaro aloqada bo‘lish natijasida faqat o‘zgargan qismlar aniqlanib, real DOM’da faqat shu qismlar yangilanadi. Bu esa sahifaning ishlash samaradorligini sezilarli darajada oshiradi.

State va Props. Props (properties) — komponentlarga tashqi tomondan uzatiladigan ma’lumotlar. Ular o‘zgartirilmaydi.State — komponent ichidagi o‘zgaruvchan ma’lumot. U foydalanuvchi harakati bilan o‘zgaradi va komponent qayta render qilinishiga sabab bo‘ladi.

React Hooks. Hooks — bu React 16.8 versiyasidan boshlab joriy etilgan funksiyalar bo‘lib, ular funksional komponentlarda state, lifecycle va boshqa xususiyatlar bilan ishlash imkonini beradi.

Asosiy hooklar:

* useState()
* useEffect()
* useContext()
* useRef()

React ekotizimi. React o‘zi faqat UI uchun kutubxona bo‘lsa-da, uning atrofida kuchli ekotizim mavjud:

| **Texnologiya** | **Tavsifi** |
| --- | --- |
| **React Router** | SPA (Single Page App) ichida yo‘nalishlarni boshqarish |
| **Redux / Zustand** | Ilova holatini (state) markazlashtirib boshqarish |
| **Next.js** | Server-side rendering va SEO uchun kengaytma |
| **Tailwind CSS / Styled-components** | Zamonaviy CSS uslublash uslublari |
| **Axios / Fetch API** | Tashqi serverlar bilan muloqot qilish uchun |

React’ning afzalliklari.

* Qayta foydalaniladigan komponentlar: Modular arxitektura kodni tartibli saqlash va kengaytirishni osonlashtiradi.
* Yuqori samaradorlik: Virtual DOM sahifani tezroq render qilish imkonini beradi.
* Katta hamjamiyat: React uchun millionlab ochiq manbali kutubxonalar mavjud.
* SEO uchun qulay (Next.js orqali): Server-side rendering bilan Google indekslash imkonini beradi.
* React Native orqali mobil ilovalar yaratish imkoniyati: React kodini mobil qurilmalarga moslashtirish mumkin.

Ilova yaratish uchun quyidagi qadamlarni bajarish kerak:

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

* npx create-react-app@latest

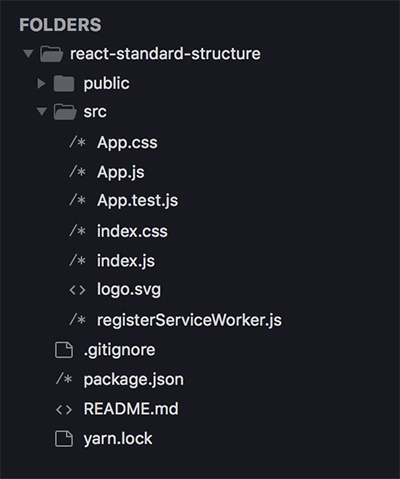
1. Keyingi qadamda ilova nomi va kerakli paketlarni belgilash kerak.
2. Ilova paketlarni yuklangandan so‘ng, terminal orqali ilova joylashga papkaga o‘tish kerak.

* cd app\_name

1. Ilovani ishga tushirish uchun quyidagi kod ishlatiladi:

* npx run dev

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



*2.6-rasm. React fayl strukturasi*

1. Bunda asosiy va boshlang‘ich yuklanuvchi kod App.js fayliga yozilgan bo‘ladi. Uni o‘zgatirish orqali dasturni o‘zgatirish munkin.

React UI kutubxonalari. React UI komponentlar kutubxonalari dasturiy ta'minotni ishlab chiqishni tezlashtirishga yordam beradigan foydalanishga tayyor yoki oldindan kodlangan foydalanuvchi interfeysi elementlari to'plamidir. Bu oldindan ishlab chiqilgan elementlar tugmalar, modallar va kartalardan tortib ma'lumotlar jadvallari va diagrammalar kabi murakkab komponentlargacha bo'ladi.

React UI komponentlar kutubxonalari ishlab chiquvchilarga dizayn va foydalanishning eng yaxshi amaliyotlariga amal qiladigan qurilish bloklari va elementlarini taqdim etish orqali foydalanuvchi interfeyslarini yaratishni soddalashtirish uchun yaratilgan [23, b. 112-128].

Quyida React UI komponentlar kutubxonasidan foydalanishning bir qancha afzalliklari keltirilgan:

Tezlik: React UI komponentlar kutubxonasidan foydalanish oldindan tuzilgan yechimlarni taqdim etish orqali dizayn xususiyatlari va funksiyalarini amalga oshirish jarayonini soddalashtiradi va tezlashtiradi. Bu shuni anglatadiki, biz har bir React komponenti uchun kodni noldan yozishimiz shart emas.

Foydalanish oson: CSS-ni o'rganish uchun vaqt talab etiladi va murakkab foydalanuvchi interfeyslarini loyihalash qiyin bo'lishi mumkin. React UI kutubxonalari bilan hatto yangi boshlanuvchilar ham CSS dizaynlariga ko'p e'tibor bermasdan mavjud elementlarni import qilishlari va ulardan foydalanishlari mumkin.

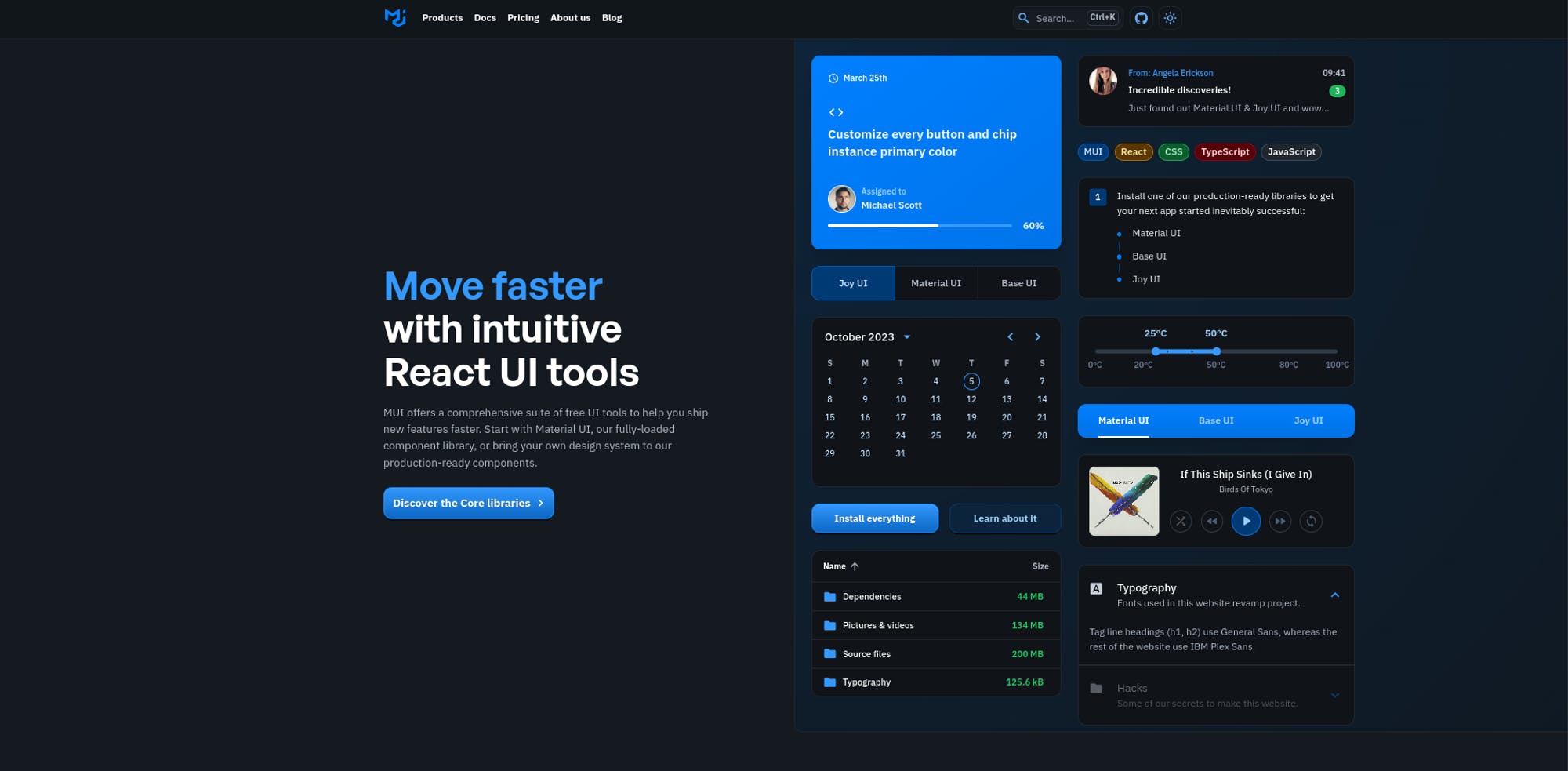
Muvofiqlik: React UI kutubxonalari ilovangiz boʻylab izchil koʻrinish va hissiyotni taʼminlaydi, chunki ular birlashtirilgan uslublar qoʻllanmasi bilan yaratilgan.

Unimdorlik: React UI komponentlar kutubxonalari tarkibiy qismlarni noldan yaratish zaruratini kamaytirish orqali ishlab chiqish vaqtini sezilarli darajada tezlashtiradi, bu esa ishlab chiquvchilarga yanada murakkab funksiyalarga e'tibor qaratish imkonini beradi.

Moslashtirish: foydalanishga tayyor React komponentlarini taklif qilishdan tashqari, bu kutubxonalar odatda loyihangizning maxsus talablariga mos ravishda sozlanishi mumkin.

Hamjamiyatni qo'llab-quvvatlash: Ommabop kutubxonalar katta jamoalardan foyda ko'radi, bu uzluksiz rivojlanish, xatolarni tuzatish va ko'plab hujjatlarni anglatadi. Siz ishlab chiquvchilar ko'pincha o'zlari foydalanadigan kutubxonalarni yangi va noyob React komponentlari bilan kengaytirishlarini topasiz., chunki bu UI kutubxonalarida ko'pincha katta jamoalar va xizmat ko'rsatuvchilar mavjud.

Foydalanish imkoniyati: Ko'pgina UI kutubxonalari foydalanish imkoniyatini birinchi o'ringa qo'yadi, bu esa hamma foydalanishi mumkin bo'lgan ilovalarni yaratishni osonlashtiradi.

1. Material UI - Google Material Design tomonidan ishlab chiqilgan ochiq manba boʻlib, u dasturini amalga oshiradigan react komponentlari kutubxonasidir. U veb ilova ishlab chiqarishda foydalanishga tayyor bo'lgan, tayyor bo'lmagan komponentlar to'plamini o'z ichiga oladi.

Asosiy xusuiyatlari:

Tipografiya: Material UI barcha matn elementlarida izchil matn uslubiga ega bo'lishini ta'minlash uchun tipografiya ko'rsatmalariga amal qiladi.

Belgilar: Material UI ilovangizda ishlatilishi mumkin bo'lgan piktogramma to'plamini taqdim etadi. Belgilarni @mui/icons-material paketidan import qilish mumkin [16, b. 212-226].

Shakl boshqaruvlari: Material UI foydalanuvchilarga qulay shakl yaratishni oson va qulay qilish uchun matn maydonlari, belgilash katakchalari, radio tugmalar va tanlash menyusi kabi bir qator shakl komponentlarini taqdim etadi.

Mavzular: Material UI qorong'u va yorug'lik rejimlarini qo'llab-quvvatlagan holda mavzuni almashtirishni sozlashni taklif qiladi. ThemeProvider komponentidan foydalanib, mavzuni o'zingiz yoqtirgan tanlovga moslashtirishingiz mumkin.

Ustun jihatlari:

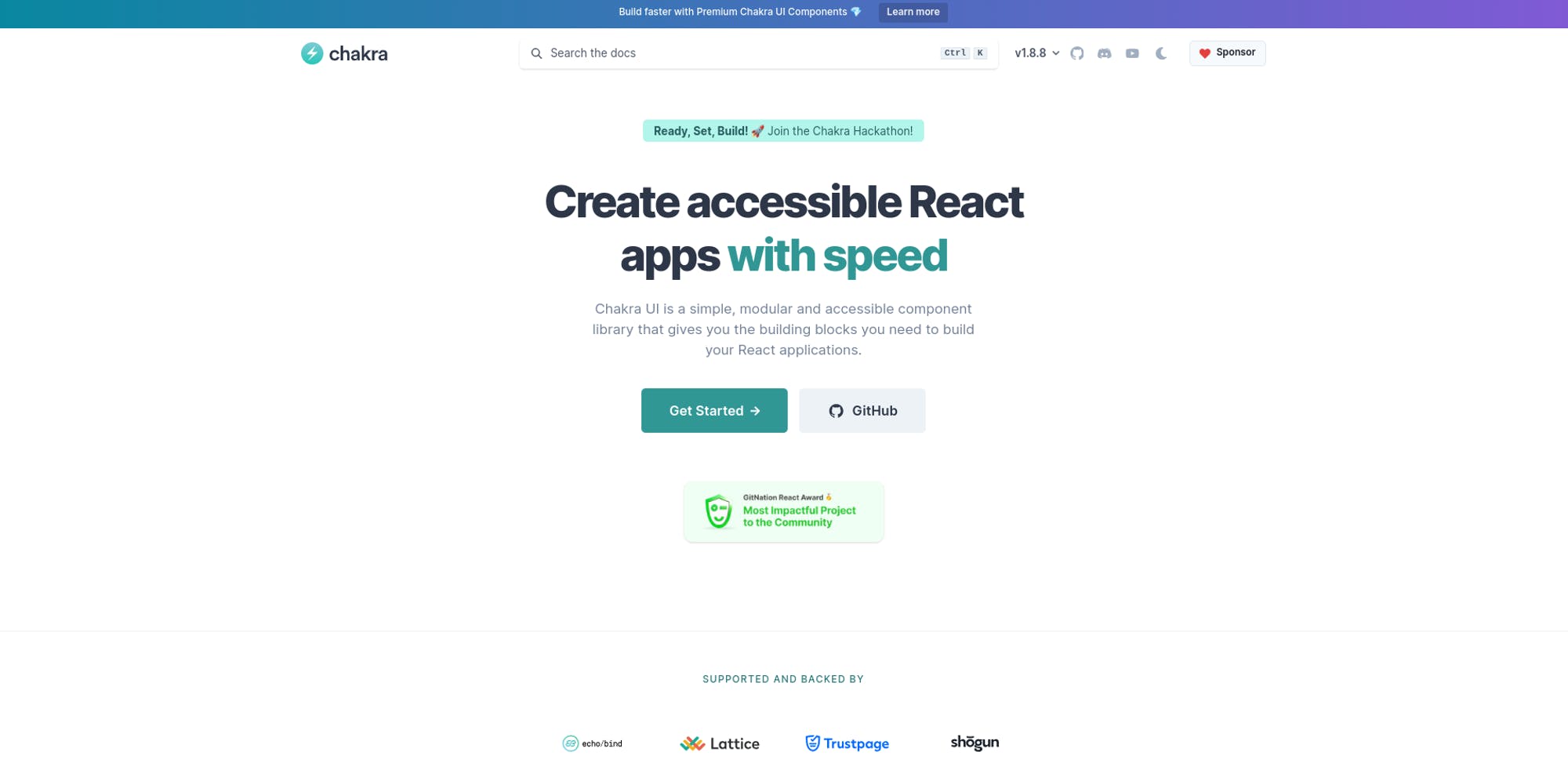
Estetika: Material UI tanlash uchun zamonaviy, chiroyli dizaynlarga ega. Dizayn estetikasi yuqori darajada ekanligiga ishonch hosil qilish uchun ular o'zlarining juda intuitiv dizayn ko'rsatmalariga amal qilishadi.

Moslashtirish: Material UI ishlab chiquvchilarga komponentlar va mavzularni o'zlarining afzal ko'rgan dizaynlariga osongina moslashtirishga imkon beruvchi sozlash variantlarini taklif qiladi.

Foydalanish imkoniyati: Material UI foydalanish imkoniyatini birinchi o‘ringa qo‘yadi. Kutubxona sukut bo'yicha foydalanish mumkin bo'lishi uchun yaratilgan bo'lib, hamma uchun foydali ilovalar yaratishni osonlashtiradi.

Ta'sirchan dizayn: Material UI komponentlari ilovangiz turli ekran o'lchamlari va qurilmalarida yaxshi ishlashini ta'minlash uchun javob beradi.

Dasturchilar uchun qulay: Hujjatlar yaxshi yozilgan, shuning uchun ishlab chiquvchilar kutubxonani osongina tushunishlari va undan foydalanishlari mumkin. Komponentlarni import qilish va amalga oshirish ham oson.

1. Chakra UI - Bu yana bir tez rivojlanayotgan, ommabop va foydalanish mumkin bo'lgan React UI kutubxonasi. Chakra UI bu oddiy, modulli kutubxona boʻlib, sizga React ilovalaringizni yaratish uchun qurilish bloklarini beradi.

Ustun jihatlari: Oddiylik: Chakra UI-dan foydalanish va tushunish oson, bu o'rganish egri chizig'ini qisqartiradi va ishlab chiquvchilarga rivojlanishga e'tibor qaratish imkonini beradi.

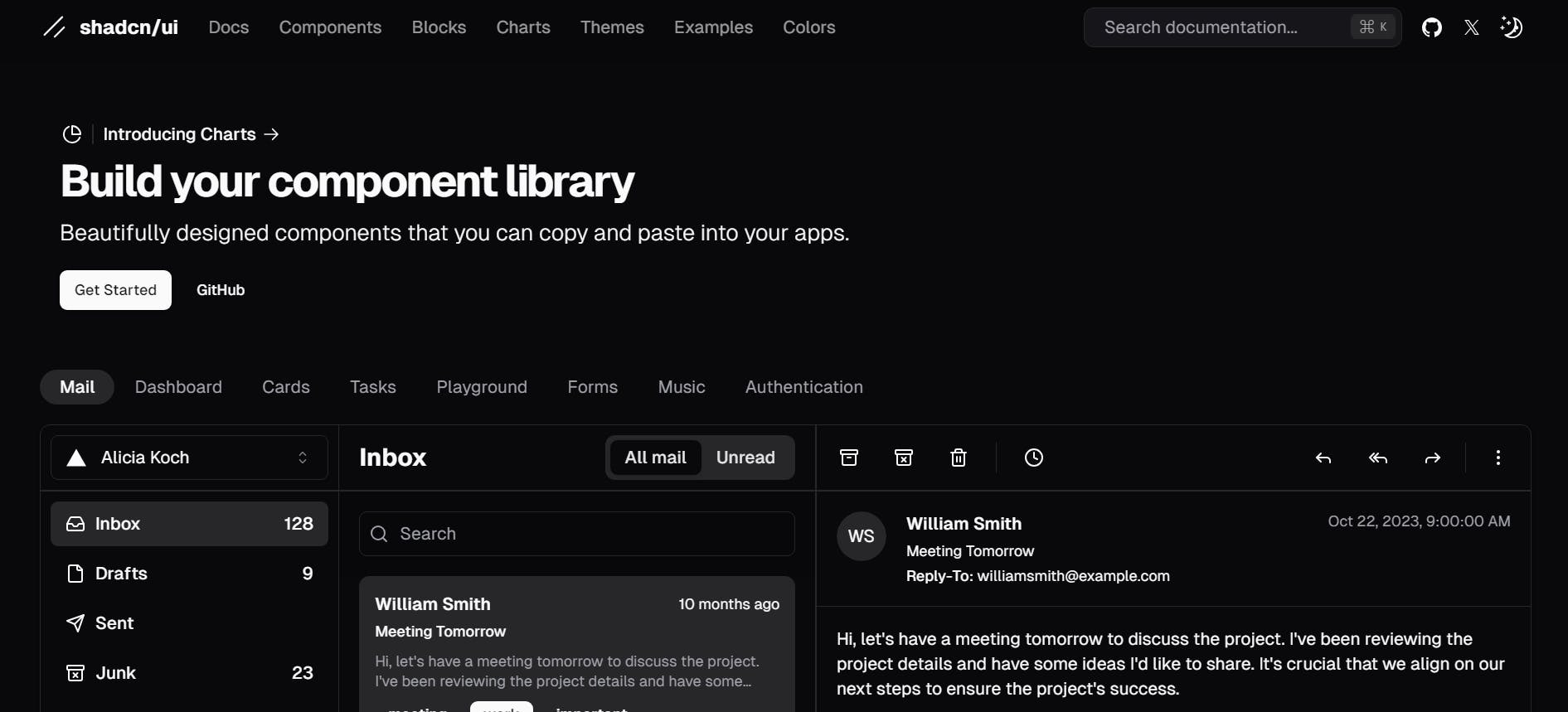
Foydalanish imkoniyati: Chakra UI foydalanish imkoniyatini hisobga olgan holda yaratilgan.

Mavzu: Chakra UI sukut bo'yicha qorong'u rejimga mos keladi. Shuningdek, siz mavzu rangini o'zingiz yoqtirgan rang rejimiga moslashtirishingiz mumkin.

Moslashtirish: Barcha komponent uslublarini uslublar rekvizitlari orqali bekor qilish yoki kengaytirish mumkin. Komponentni osongina sozlashingiz mumkin, jumladan ranglar va bo'shliqlar kabi [13, b. 23-24].

Hujjatlar: Hujjatlar foydalanish bo'yicha ko'rsatmalar va misollar bilan to'liq va yaxshi yozilgan.

Dasturchilar hamjamiyati: U faol va rivojlanayotgan ishlab chiquvchilar hamjamiyatiga ega. Asosiy jamoadan tashqari, yuzdan ortiq ishlab chiquvchilar loyihaga faol hissa qo'shmoqda, shuning uchun kutubxonadagi manbalar va yaxshilanishlarni ko'rishingiz mumkin.

3. Shadcn UI - Shadcn UI tez o'sib borayotgan komponentlar kutubxonasi bo'lib, u uslublar uchun Tailwind CSS va Radix UI-dan foydalanadi. U soddalik, tezlik va foydalanish imkoniyatiga e'tibor qaratgan holda foydalanuvchi interfeyslarini yaratishga yangicha yondashuvni taklif etadi.

Loyihangizga yuklab olishingiz uchun barcha komponentlarni paketga birlashtirish o‘rniga, Shadcn UI nusxa ko‘chirish va joylashtirish usulini qo‘llaydi va istalgan hujjat kodini to‘g‘ridan-to‘g‘ri loyihangizga nusxalash imkonini beradi.

Asosiy xususuyatlari:

Mavzular va mavzu muharriri: Bu sizning brend identifikatoringiz yoki loyiha talablariga mos keladigan ranglar, chegara radiusi va boshqa vizual elementlarni osongina sozlash imkonini beruvchi mustahkam mavzu muharririni taqdim etadi.

Yorug'lik va qorong'u rejim: Shadcn UI qutidan tashqari yorug'lik va qorong'i rejimni qo'llab-quvvatlaydi. Siz ushbu xususiyatdan mavzuni almashtirish funksiyasini ilovangizga integratsiya qilish uchun foydalanishingiz mumkin.

Tailwind CSS-ni qo'llab-quvvatlaydi: U Tailwind CSS-ning ustiga qurilgan va Tailwind-ning birinchi yordam dasturidan foydalanadi. Bu keng CSS kodini yozmasdan tezda maxsus uslublarni yaratishingiz mumkinligini anglatadi.

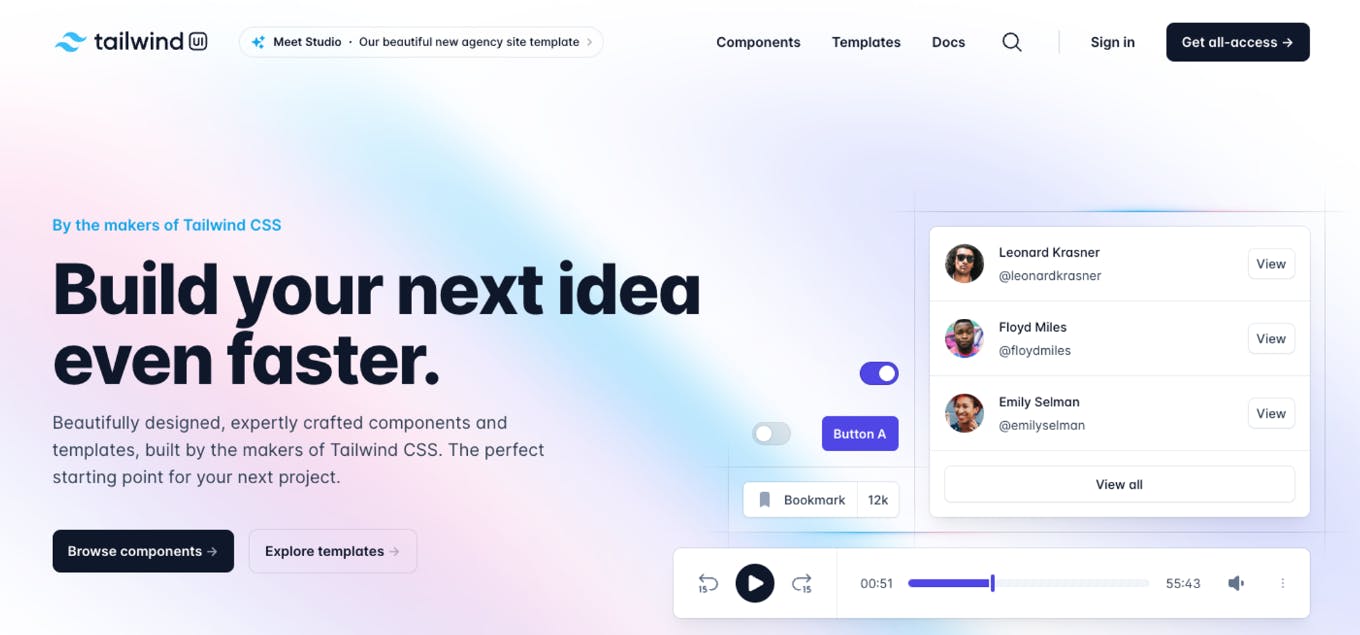
v0 da tahrirlash: Bu xususiyat v0 ilovasida Shadcn UI komponentini - React komponentlarini yaratuvchi Vercelning AI vositasini ochish va komponentni takliflar orqali sozlash imkonini beradi.

Ustun jihatlari:

Moslashtirilgan dizaynlar: Mavzu muharriri va uning Tailwind CSS bilan integratsiyalashuvi sozlash uchun katta erkinlik beradi. Bu sizning loyihangiz va brendingiz talablariga mos keladigan noyob va markali foydalanuvchi interfeyslarini yaratishga imkon beradi.

Komponentga egalik: Shadcn UI sizga komponentlar kodiga to'g'ridan-to'g'ri kirish imkonini berib, uning tarkibiy qismlari ustidan ko'proq nazorat va egalik huquqini beradi. Ushbu yondashuv bilan siz foydalanadigan har bir komponent bevosita kod bazangizda yashaydi.

4. Tailwind UI - Tailwind CSS yaratuvchilari tomonidan yaratilgan rasmiy UI kutubxonasi. U Tailwind CSS uchun oldindan tuzilgan, sozlanishi va ishlab chiqarishga tayyor UI komponentlarini taqdim etadi. Bu oylik va yillik narxlash imkoniyatlariga ega pullik kutubxona.



Asosiy xususiyatlari:

Oldindan oʻrnatilgan komponentlar: Tailwind UI navigatsiya paneli, qahramon boʻlimlari, kartalar, shakllar va boshqalar kabi oʻnlab oldindan ishlab chiqilgan, oldindan tuzilgan UI komponentlarini taklif etadi.

Mashhur ramkalar uchun dizaynlar: Komponentlar React, Vue va AlpineJS kod misollari bilan birga keladi, shuning uchun ularni osongina birlashtira olasiz.

Moslashuvchan: Barcha komponentlarni Tailwind CSS yordam dasturlari sinflari bilan sozlash mumkin. Ranglarni, intervallarni, shriftlarni va boshqalarni o'zgartiring [12, b. 213].

O'sib borayotgan kutubxona: foydalanuvchilarning fikr-mulohazalari va so'rovlari asosida yangi komponentlar doimiy ravishda qo'shiladi.

Premium qo'llab-quvvatlash: Pulli rejalar Tailwind UI jamoasidan birma-bir elektron pochta yordamini o'z ichiga oladi.

Ustun jihatlari:

Rivojlanish vaqtini tejaydi: Tayyor komponentlar noldan kodlashdan ko'ra tezroq UI yaratishga yordam beradi.

Chiroyli dizaynlar: Komponentlar estetika, qulaylik va qulaylik uchun eng yaxshi amaliyotlarga amal qiladi.

Sozlash oson: Tailwind CSS sinflari minimal kuch bilan keng ko'lamli sozlash imkonini beradi.

Mashhur ramkalarni qo'llab-quvvatlaydi: Komponentlarni React, Vue va AlpineJS ilovalariga osongina integratsiyalash.

Faol rivojlanish: Jamoa muntazam ravishda yangi komponentlar va yaxshilanishlarni qo'shib boradi.

Hujjatlar: Komponentlardan foydalanish va sozlash bo'yicha aniq ko'rsatmalar va misollar.

# **III BOB. ONLAYN PLATFORMASINI ISHLAB CHIQISH BOSQICHLARI**

## **3.1. Dasturiy ta’minotni loyihalashtirish**

Dasturiy ta’minotni ishlab chiqish jarayonining muhim bosqichlaridan biri bu - tizimni to‘g‘ri va puxta loyihalashtirishdir. Ushbu bosqichda dasturiy mahsulotning funksional imkoniyatlari, texnik yechimlari, ma’lumotlar bazasi strukturalari va interfeys dizayni aniqlanadi. Bu holat platformaning foydalanuvchilar uchun qulayligi, ishonchliligi hamda kelajakdagi kengaytirilish imkoniyatlarini belgilab beradi.

Platformaning asosiy funksiyalari. Ishlab chiqilishi rejalashtirilayotgan tipografiya va reklama xizmatlarini taqdim qiluvchi onlayn platforma quyidagi asosiy funksiyalarga ega bo‘lishi kerak:

* Xizmatlar katalogi (vizitka, flyer, banner, logotip, va h.k.)
* Ijrochini tanlash va aloqaga chiqish imkoniyati
* Fayllarni yuklash va qabul qilish

Tizim klient-server arxitekturasi asosida ishlab chiqiladi. Bu yondashuv foydalanuvchilar soni ortgani sari tizimning barqaror ishlashini ta’minlaydi. Arxitektura quyidagi asosiy komponentlardan iborat bo‘ladi:

* Frontend (klient qismi): HTML, Tailwind CSS va JavaScript yordamida, foydalanuvchilar uchun qulay interfeys yaratiladi. Dizaynerlar va mijozlar uchun alohida interfeyslar yaratiladi.
* Backend (server qismi): Django framework yordamida ma’lumotlar bazasi bilan ishlash, foydalanuvchi autentifikatsiyasi, buyurtmalarni boshqarish va API lar yaratiladi.
* Ma’lumotlar bazasi: SQLite yoki PostgreSQL dan foydalaniladi. Ma’lumotlar bazasida foydalanuvchilar, buyurtmalar, xizmat turlari, chatlar va to‘lovlar haqidagi ma’lumotlar saqlanadi.
* REST API: Frontend va backend o‘rtasida ma’lumot almashinuvi RESTful API orqali amalga oshiriladi. Bu tizimni kengaytirilishini osonlashtiradi.

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

Foydalanuvchilarga qulay interfeys taqdim etish — tizim muvaffaqiyatining muhim omillaridan biridir. Dizayn prinsiplariga quyidagilar amal qilinadi:

* Minimalistik va toza dizayn — ortiqcha elementlar yo‘q, asosiy e’tibor xizmatlarga qaratilgan.
* Responsive dizayn — tizim barcha qurilmalarda (kompyuter, planshet, telefon) to‘g‘ri ko‘rinadi.
* Foydalanuvchining roli asosida interfeyslar farqlanadi — mijozlar uchun buyurtma berish osonlashtirilgan, dizaynerlar uchun esa ishlash qulay interfeys yaratiladi.
* Qidiruv va filtr funksiyasi — xizmatlarni kategoriya, narx va reyting bo‘yicha tez topish imkoniyati.

Interfeys yaratishda Tailwind CSS, Alpine.js va Django templating engine dan foydalanish rejalashtirilmoqda.

Veb ilovasining dizayn qismini loyihalash jarayoni, foydalanuvchi tajribasini (UX) va foydalanuvchi interfeysini (UI) samarali va intuitiv tashkil etishga qaratilgan. Yaxshi dizayn nafaqat vizual jozibadorlikni ta’minlash, balki foydalanuvchining ilova bilan o‘zaro aloqasini engillashtirishga ham xizmat qiladi. Ushbu jarayonni muvaffaqiyatli amalga oshirish uchun quyidagi bosqichlarni ko‘rib chiqish zarur:

**Foydalanuvchi talablarini aniqlash**

Ilovaning dizaynini yaratishdan oldin, foydalanuvchilarning ehtiyojlarini va talablarini aniqlash juda muhim. Buning uchun foydalanuvchi so‘rovnomalari, intervyular, tahlil va foydalanish ma'lumotlari to‘plami o‘rganiladi. Foydalanuvchilarning asosiy maqsadlari va kutgan natijalari bilan tanishish, dizaynning eng muhim elementlarini aniqlashga yordam beradi.

**Wireframe va prototiplarni ishlab chiqish**

Ilovaning dastlabki shakli va tuzilmasini aniqlash uchun wireframe (skeletli dizayn) va prototiplar tuziladi. Bu bosqichda, ilovaning interfeysining asosiy tuzilmasi belgilangan bo‘ladi, masalan, sahifalar o‘rtasidagi navigatsiya, kontent joylashuvi va foydalanuvchi imkoniyatlari. Prototiplar foydalanuvchilarga ilova bilan ishlashni ta’minlash va qanday ishlashini oldindan ko‘rsatish imkonini beradi.

**UI dizayni**

UI dizayni – foydalanuvchining ilova bilan o‘zaro aloqasi va vizual ko‘rinishini shakllantirishni o‘z ichiga oladi. Bu bosqichda, ranglar palitrasi, shriftlar, tugmalar, ikonalar, grafikalar va boshqa dizayn elementlari tanlanadi. UI dizaynining maqsadi, ilovaning foydalanuvchi uchun estetik jihatdan jozibador va foydalanish uchun qulay bo‘lishini ta’minlashdir.

**Interaktivlikni ta’minlash**

Dizaynni ishlab chiqishning muhim jihatlaridan biri – foydalanuvchi interfeysining interaktivligini ta’minlashdir. Bu sahifalar o‘rtasida o‘tish, animatsiyalar, tugmalar va formalarni interaktiv qilishni o‘z ichiga oladi. Interaktiv elementlar foydalanuvchi tajribasini yaxshilash va ilova bilan ishlashni yanada intuitiv qilishga yordam beradi.

**Foydalanuvchi tajribasini (UX) optimallashtirish**

Ilovaning dizayni foydalanuvchi tajribasiga ta’sir qiladi. UX dizayni – foydalanuvchining ilova bilan qanday ishlashini va qanday his qilishini optimallashtirishga qaratilgan. Bu bosqichda, foydalanuvchining ehtiyojlarini qondirish, navigatsiyani soddalashtirish, foydalanuvchi oqimini to‘g‘ri tashkil etish, va xatoliklarni kamaytirish zarur. UX dizayni, ilovaning intuitiv bo‘lishini ta’minlaydi, shuning uchun foydalanuvchi maqsadiga yetishish uchun vaqtni tejashga yordam beradi.

**Mobil va javob beradigan dizayn**

Ilovalar mobil qurilmalarda ham ishlashi kerak, shuning uchun dizaynni javobgar (responsive) qilish juda muhimdir. Bu, foydalanuvchining ilovani har xil ekran o‘lchamlarida, ya'ni telefon, planshet, noutbuk va desktopda qulay foydalanishi uchun muhimdir. Dizaynning har xil qurilmalarga mos kelishini ta’minlash, foydalanuvchi tajribasini yaxshilaydi va ilovaning samarali ishlashini kafolatlaydi.

**Sinov va foydalanuvchi fikrlarini yig‘ish**

Dizaynni yaratish jarayonida foydalanuvchi fikrlarini doimiy ravishda olish muhimdir. Ilovaning prototipi yoki dastlabki versiyasini foydalanuvchilarga taqdim etish va ularning fikr-mulohazalarini yig‘ish, dizaynni yaxshilash va foydalanuvchilarning talablariga moslashtirishga yordam beradi. Bu bosqichda, foydalanuvchi sinovlari va tahlillari orqali dizaynni yanada mukammallashtirish mumkin.

**Dizaynning yakuniy tasdiqlanishi**

Ilovaning dizayni yakuniy shaklini olishidan oldin, barcha dizayn elementlari va funktsiyalari tekshiriladi. Sinovlar va foydalanuvchi fikrlari asosida kerakli o‘zgartirishlar kiritiladi. Yakuniy dizayn tasdiqlangach, ishlab chiqish bosqichi boshlanadi.

Veb ilovaning dizayn qismini loyihalash foydalanuvchi tajribasi va interfeysi nuqtai nazaridan juda muhimdir. Yaxshi ishlab chiqilgan dizayn nafaqat ilovaning funksional ishlashini ta’minlaydi, balki foydalanuvchining mahsulot bilan aloqasini engillashtiradi va samarali qiladi. Har bir dizayn bosqichi o‘ziga xos yondashuvni talab qiladi, shuningdek, foydalanuvchining ehtiyojlarini va xavfsizligini ta’minlashda asosiy o‘rin tutadi.

# **XULOSA**

Bitiruv malakaviy ishi reaklama xizmatlarini online amalga oshirish orqali foydalanuvchini ehtiyojlarini qondrishdan iborat. Bunda bizga ma’lum dasturlash tilining tuzilmalarini o‘rganish, frameworklar bilan ishlash va dasturning interfasini UI kutubxonalari ishlab chiqish kabi ustuvor vazifalar qo‘yildi. O‘rganish davomida ko‘plab yangi texnalogiyalar bo‘yicha tushunchalarga ega bo‘lindi va ishlab chiqilgan dasturning amaliyotga tadbiqi o‘rganildi.

Ushbu ishlab chiqiladigan dasturiy ilovaning afzalliklari shundan iboratki, oldin Foydalanuvchilar o’zlariga kerak bo’lgan xizmatlarni amalga oshirish uchun xizmat ko’rsatish offislariga borishi kerak bo’lgan. Bunda ko‘p vaqtni ketadi va xarajatlar ham oshadi. Yaratilgan dastur bilan esa mijozlar o’zlari uchun kerakli xizmatni tanlashi va 0 dan yaratishi yoki tayyor shablon asosida oldindan kiritilgan matinlarni tegishli qismlarini o’zgartirish orqali yaratishi mumkin. Ushbu bitiruv malakaviy ishida xorijiy tillarni o‘qitishda portfolioga asoslangan baholash usulining afzalliklari, mohiyati va amaliy qo‘llanilishi atroflicha yoritildi. O‘tkazilgan nazariy va amaliy tadqiqotlar asosida quyidagi asosiy xulosalarga kelindi:

Portfolio baholash an’anaviy baholashga qaraganda ko‘proq o‘quvchining ijodiy fikrlashi, o‘z-o‘zini nazorat qilishi va o‘zlashtirish jarayoniga faol jalb etilishini ta’minlaydi. Bu usul orqali o‘quvchilar o‘z yutuq va kamchiliklarini mustaqil tahlil qilishni o‘rganadilar.

Tadqiqotlar shuni ko‘rsatadiki, portfolio baholash o‘quvchilarning kommunikativ kompetensiyasini rivojlantirishda muhim vosita bo‘lib xizmat qiladi. Xususan, yozma va og‘zaki nutq ko‘nikmalarini baholashda bu usul talabalarning bilimini chuqur tahlil qilish imkonini beradi.

Portfolio usuli orqali o‘qituvchilar o‘quvchining rivojlanish dinamikasini tizimli kuzatib borishlari, shuningdek, o‘quvchilarga individual yondashuv asosida fikr-mulohaza (feedback) berishlari mumkin. Bu esa baholash jarayonini yanada shaffof va samarali qiladi.

Tahlil etilgan xorijiy tajribalar asosida aytish mumkinki, portfolioga asoslangan baholash nafaqat o‘qituvchi va o‘quvchi o‘rtasidagi aloqani mustahkamlaydi, balki o‘quvchilarda mas’uliyat hissini, o‘z-o‘zini rivojlantirish motivatsiyasini kuchaytiradi.

Portfolio usulining amaliy joriy etilishi uchun o‘qituvchilar malakasini oshirish, zarur metodik qo‘llanmalar ishlab chiqish va baholash mezonlarini standartlashtirish dolzarb hisoblanadi.

Shu asosda, xorijiy tillarni o‘qitishda portfolioga asoslangan baholashni keng joriy etish zamonaviy ta’lim tizimi oldida turgan asosiy vazifalardan biridir. Bu yondashuv ta’lim sifatini oshirish, o‘quv jarayonini shaxsga yo‘naltirilgan tarzda tashkil etishda muhim bosqich bo‘lib xizmat qiladi.

Foydalanuvchilarga veb dasturdan foydalanishi yanada qulaylashtirish va veb ilova dizayni jozibaliroq qilish uchun zamonaviy UI kutubxona va uning komponentalaridan foydalanildi. Bu orqali har qanday foydalanuvchini o’ziga jalb qilish va reklama xizmatlaridan foydalanishga undaydi va bu ilovadan foydalanishni yanada soddalashtiradi.

# **FOYDALANILGAN ADABIYOTLAR RO`YXATI**

**1. Normativ-huquqiy hujjatlar**

1. O'zbekiston Respublikasi Konstitutsiyasi. – Toshkent: O'zbekiston, 2019. – 100 b.
2. O'zbekiston Respublikasining "Ta'lim to'g'risida"gi Qonuni. – Toshkent: Adolat, 2020. – 75 b.
3. O'zbekiston Respublikasi Prezidentining "Ta'lim sifatini oshirishga oid chora-tadbirlar to'g'risida"gi farmoni. – Toshkent: Prezident matbuot xizmati, 2021. – 50 b.
4. O'zbekiston Respublikasi Vazirlar Mahkamasining "Milliy baholash tizimini takomillashtirish to'g'risida"gi qarori. – Toshkent: Hukumat nashriyoti, 2018. – 65 b.
5. O'zbekiston Respublikasi "Kadrlar tayyorlash milliy dasturi to'g'risida"gi qonuni. – Toshkent: O'zbekiston, 1997. – 80 b.
6. O'zbekiston Respublikasi "Oliy ta'lim to'g'risida"gi qonuni. – Toshkent: Adolat, 2019. – 90 b.
7. Mirziyoyev, Sh.M. Yangi O‘zbekiston strategiyasi. – Toshkent: O'zbekiston, 2021. – 350 b.

**2. Kitoblar**

1. Filer, A. Assessment: Social Practice and Social. – UK: Routledge Falmer, 2000. – 347 b.
2. Adam, S. Blog Post. – UK: Self-Published, 2013. – 67 b.
3. Atkins, M., Beattie, J., Dockrell, B. Higher Education Employment Department Assessment Issues. – Sheffield: University Press, 1993. – 68-72 b.
4. Baume, D. Portfolios for Learning and Evaluation. – UK: Routledge, 2000. – 267-275 b.
5. Brown, S., Glasner, A. (Ed.). Assessment Issues in Higher Education. – UK: Open University Press, 1999.
6. Caner, M. Students’ Views on the Use of Portfolio Assessment in EFL Writing Courses. – Turkey: Anadolu University, 2010. – 23-24 b.
7. Chirimbu, S. Use of Alternative Assessment Methods in Teaching Foreign Languages. – Romania: Educational Studies, 2013.
8. Davis, M. H. Portfolio Assessment. – USA: JVME, 2005. – 32(3).
9. Hamayan, E. V. Approaches to Alternative Assessment. – USA: Annual Review of Applied Linguistics, 1995. – 15, 212–226 b.
10. Huerta-Macias, A. Alternative Assessment: Answers to Frequently Asked Questions. – USA: TESOL Magazine, 1995. – 5, 8–10 b.
11. Moss, P. A. The Impact of Portfolio Assessment on the Writing Performance of EFL High School Students in Saudi Arabia. – Saudi Arabia: Educational Press, 2005.
12. Neiman, D. Assessment Types. – USA: Research Publications, 2007. – 22 b.
13. Rudner, L., Eric, A. E., Boston, C. Long Review of Actual Assessment. – USA: Educational Reports, 2008. – 78 b.

**3. Ilmiy maqolalar**

1. Til o'rganish muqobil baholash. // TESOL chorakda. – 2012. – №4. – B. 75–89.
2. Ismatov A. Oliy ta’lim muassasalarida til oʻrgatishning innovatsion metodlari. // Ilm-fan va ta’lim. – 2021. – №2. – B. 33–40.
3. Jones, M., Saville, N. Tilni bilish darajasini baholash: an'anaviy va muqobil usullarni qiyosiy o'rganish. // Til sinovlari jurnali. – 2019. – №3. – B. 112–128.
4. Qosimov, D. Talabalarning kommunikativ kompetensiyasini baholashning samarali metodlari. // Pedagogika va psixologiya jurnali. – 2022. – №5. – B. 50–59.
5. Kumaravadivelu, B. Til o‘rganishda muqobil baholash. // TESOL choraklik. – 2012. – №4. – B. 75–89.

**4. Elektron manbalar**

1. O‘zbekiston Respublikasi Qonun hujjatlari ma’lumotlar bazasi: [www.lex.uz](http://www.lex.uz)
2. YUNESKO ta'lim va baholash resurslari: [www.unesco.org/education](http://www.unesco.org/education)
3. Ta'lim sifatini baholash bo‘yicha xalqaro tadqiqotlar: [www.oecd.org/education](http://www.oecd.org/education)
4. O‘zbekiston Respublikasi Oliy va o‘rta maxsus ta’lim vazirligi rasmiy veb-sayti: [www.edu.uz](http://www.edu.uz)
5. Jahon bankining ta’lim bo‘yicha tadqiqotlari: [www.worldbank.org/education](http://www.worldbank.org/education)
6. Ta’limda baholash va innovatsiyalar bo‘yicha ilmiy maqolalar: [www.sciencedirect.com](http://www.sciencedirect.com)
7. Tillar uchun umumiy Yevropa ma’lumot bazasi (CEFR): [www.coe.int](http://www.coe.int)

**ILOVALAR**

**Asosiy App.jsx componentasi:**

import './App.css'

import { BrowserRouter as Router, Routes, Route } from "react-router-dom";

import { ThemeProvider, useTheme } from "@/components/theme-provider"

import Home from "./pages/Home";

import { useEffect } from 'react';

import CreateVisitCard from './pages/VisitCard/CreateVisitCard';

import CreateInvitationCard from './pages/Invitation/CreateInvitation';

import CreateGreetingCard from './pages/GreetingCard/CreateGreetingCard';

function App() {

  const { theme } = useTheme();

  useEffect(() => {

    document.body.className = theme;

  }, [theme]);

  return (

    <>

      <ThemeProvider defaultTheme="dark" storageKey="vite-ui-theme">

        <Router>

          <Routes>

            <Route path="/" element={<Home />} />

            <Route path="/create-visit-card/:id" element={<CreateVisitCard />} />

            <Route path="/create-visit-card" element={<CreateVisitCard />} />

            <Route path="/create-invitation-card/:id" element={<CreateInvitationCard />} />

            <Route path="/create-invitation-card" element={<CreateInvitationCard />} />

            <Route path="/create-greeting-card/:id" element={<CreateGreetingCard />} />

            <Route path="/create-greeting-card" element={<CreateGreetingCard />} />

          </Routes>

        </Router>

      </ThemeProvider>

    </>

  )

}

export default App

**Home componentasi:**

import Services from "@/components/Services";

import Header from "@/components/Header";

import Footer from "@/components/Footer";

import Templates from "@/components/Templates";

export default function Home() {

  return (

    <>

      <Header/>

      <section

        id="home"

        className="dark:bg-gray-dark relative z-10 overflow-hidden  pb-16 pt-[120px] md:pb-[120px] md:pt-[150px] xl:pb-[160px] xl:pt-[180px] 2xl:pb-[200px] 2xl:pt-[210px]"

      >

        <div className="container ">

          <div className="mx-4 flex flex-wrap">

            <div className="w-full px-4">

              <div

                className="wow fadeInUp mx-auto max-w-[800px] text-center"

                data-wow-delay=".2s"

              >

                <h1 className="mb-5 text-3xl font-bold leading-tight text-body-color dark:text-body-color-dark sm:text-4xl sm:leading-tight md:text-5xl md:leading-tight">

                  Taklifnoma, tashrifnoma, tabriknoma — barchasi bir joyda!

                </h1>

                <p className="dark:text-body-color-dark mb-12 text-base !leading-relaxed text-body-color sm:text-lg md:text-xl">

                  Endi siz nafaqat professional tashrifnomalar, balki chiroyli taklifnomalar va o‘zgacha tabriknomalarni ham yaratishingiz mumkin. Siz oson yaratish, tez sozlash va darhol yuklab olish imkoniyati mavjud.

                  Uzingizga mos bo‘lgan shablonlari tanlang va yarating!

                </p>

                <div className="flex flex-col items-center justify-center space-y-4 sm:flex-row sm:space-x-4 sm:space-y-0">

                  <a

                    href="/#templates"

                    className="rounded-sm px-8 py-4 text-base text-white font-semibold bg-blue-700 hover:bg-blue-800 duration-300 ease-in-out hover:bg-blue/80"

                  >

                    🔥Shablonni tanlash

                  </a>

                  <a

                    href="/#services"

                    className="inline-block rounded-sm bg-black px-8 py-4 text-base font-semibold text-body-color dark:text-body-color-dark duration-300 ease-in-out hover:bg-black/90 dark:bg-white/10 dark:hover:bg-white/5"

                  >

                    ✏️ Dizayn qilishni boshlash

                  </a>

                </div>

              </div>

            </div>

          </div>

        </div>

        <div className="absolute right-0 top-0 z-[-1] opacity-30 lg:opacity-100">

          <svg

            width="450"

            height="556"

            viewBox="0 0 450 556"

            fill="none"

            xmlns="http://www.w3.org/2000/svg"

          >

            <circle

              cx="277"

              cy="63"

              r="225"

              fill="url(#paint0\_linear\_25:217)"

            />

            <circle

              cx="17.9997"

              cy="182"

              r="18"

              fill="url(#paint1\_radial\_25:217)"

            />

            <circle

              cx="76.9997"

              cy="288"

              r="34"

              fill="url(#paint2\_radial\_25:217)"

            />

            <circle

              cx="325.486"

              cy="302.87"

              r="180"

              transform="rotate(-37.6852 325.486 302.87)"

              fill="url(#paint3\_linear\_25:217)"

            />

            <circle

              opacity="0.8"

              cx="184.521"

              cy="315.521"

              r="132.862"

              transform="rotate(114.874 184.521 315.521)"

              stroke="url(#paint4\_linear\_25:217)"

            />

            <circle

              opacity="0.8"

              cx="356"

              cy="290"

              r="179.5"

              transform="rotate(-30 356 290)"

              stroke="url(#paint5\_linear\_25:217)"

            />

            <circle

              opacity="0.8"

              cx="191.659"

              cy="302.659"

              r="133.362"

              transform="rotate(133.319 191.659 302.659)"

              fill="url(#paint6\_linear\_25:217)"

            />

            <defs>

              <linearGradient

                id="paint0\_linear\_25:217"

                x1="-54.5003"

                y1="-178"

                x2="222"

                y2="288"

                gradientUnits="userSpaceOnUse"

              >

                <stop stopColor="#4A6CF7" />

                <stop offset="1" stopColor="#4A6CF7" stopOpacity="0" />

              </linearGradient>

              <radialGradient

                id="paint1\_radial\_25:217"

                cx="0"

                cy="0"

                r="1"

                gradientUnits="userSpaceOnUse"

                gradientTransform="translate(17.9997 182) rotate(90) scale(18)"

              >

                <stop offset="0.145833" stopColor="#4A6CF7" stopOpacity="0" />

                <stop offset="1" stopColor="#4A6CF7" stopOpacity="0.08" />

              </radialGradient>

              <radialGradient

                id="paint2\_radial\_25:217"

                cx="0"

                cy="0"

                r="1"

                gradientUnits="userSpaceOnUse"

                gradientTransform="translate(76.9997 288) rotate(90) scale(34)"

              >

                <stop offset="0.145833" stopColor="#4A6CF7" stopOpacity="0" />

                <stop offset="1" stopColor="#4A6CF7" stopOpacity="0.08" />

              </radialGradient>

              <linearGradient

                id="paint3\_linear\_25:217"

                x1="226.775"

                y1="-66.1548"

                x2="292.157"

                y2="351.421"

                gradientUnits="userSpaceOnUse"

              >

                <stop stopColor="#4A6CF7" />

                <stop offset="1" stopColor="#4A6CF7" stopOpacity="0" />

              </linearGradient>

              <linearGradient

                id="paint4\_linear\_25:217"

                x1="184.521"

                y1="182.159"

                x2="184.521"

                y2="448.882"

                gradientUnits="userSpaceOnUse"

              >

                <stop stopColor="#4A6CF7" />

                <stop offset="1" stopColor="white" stopOpacity="0" />

              </linearGradient>

              <linearGradient

                id="paint5\_linear\_25:217"

                x1="356"

                y1="110"

                x2="356"

                y2="470"

                gradientUnits="userSpaceOnUse"

              >

                <stop stopColor="#4A6CF7" />

                <stop offset="1" stopColor="white" stopOpacity="0" />

              </linearGradient>

              <linearGradient

                id="paint6\_linear\_25:217"

                x1="118.524"

                y1="29.2497"

                x2="166.965"

                y2="338.63"

                gradientUnits="userSpaceOnUse"

              >

                <stop stopColor="#4A6CF7" />

                <stop offset="1" stopColor="#4A6CF7" stopOpacity="0" />

              </linearGradient>

            </defs>

          </svg>

        </div>

        <div className="absolute bottom-0 left-0 z-[-1] opacity-30 lg:opacity-100">

          <svg

            width="364"

            height="201"

            viewBox="0 0 364 201"

            fill="none"

            xmlns="http://www.w3.org/2000/svg"

          >

            <path

              d="M5.88928 72.3303C33.6599 66.4798 101.397 64.9086 150.178 105.427C211.155 156.076 229.59 162.093 264.333 166.607C299.076 171.12 337.718 183.657 362.889 212.24"

              stroke="url(#paint0\_linear\_25:218)"

            />

            <path

              d="M-22.1107 72.3303C5.65989 66.4798 73.3965 64.9086 122.178 105.427C183.155 156.076 201.59 162.093 236.333 166.607C271.076 171.12 309.718 183.657 334.889 212.24"

              stroke="url(#paint1\_linear\_25:218)"

            />

            <path

              d="M-53.1107 72.3303C-25.3401 66.4798 42.3965 64.9086 91.1783 105.427C152.155 156.076 170.59 162.093 205.333 166.607C240.076 171.12 278.718 183.657 303.889 212.24"

              stroke="url(#paint2\_linear\_25:218)"

            />

            <path

              d="M-98.1618 65.0889C-68.1416 60.0601 4.73364 60.4882 56.0734 102.431C120.248 154.86 139.905 161.419 177.137 166.956C214.37 172.493 255.575 186.165 281.856 215.481"

              stroke="url(#paint3\_linear\_25:218)"

            />

            <circle

              opacity="0.8"

              cx="214.505"

              cy="60.5054"

              r="49.7205"

              transform="rotate(-13.421 214.505 60.5054)"

              stroke="url(#paint4\_linear\_25:218)"

            />

            <circle cx="220" cy="63" r="43" fill="url(#paint5\_radial\_25:218)" />

            <defs>

              <linearGradient

                id="paint0\_linear\_25:218"

                x1="184.389"

                y1="69.2405"

                x2="184.389"

                y2="212.24"

                gradientUnits="userSpaceOnUse"

              >

                <stop stopColor="#4A6CF7" stopOpacity="0" />

                <stop offset="1" stopColor="#4A6CF7" />

              </linearGradient>

              <linearGradient

                id="paint1\_linear\_25:218"

                x1="156.389"

                y1="69.2405"

                x2="156.389"

                y2="212.24"

                gradientUnits="userSpaceOnUse"

              >

                <stop stopColor="#4A6CF7" stopOpacity="0" />

                <stop offset="1" stopColor="#4A6CF7" />

              </linearGradient>

              <linearGradient

                id="paint2\_linear\_25:218"

                x1="125.389"

                y1="69.2405"

                x2="125.389"

                y2="212.24"

                gradientUnits="userSpaceOnUse"

              >

                <stop stopColor="#4A6CF7" stopOpacity="0" />

                <stop offset="1" stopColor="#4A6CF7" />

              </linearGradient>

              <linearGradient

                id="paint3\_linear\_25:218"

                x1="93.8507"

                y1="67.2674"

                x2="89.9278"

                y2="210.214"

                gradientUnits="userSpaceOnUse"

              >

                <stop stopColor="#4A6CF7" stopOpacity="0" />

                <stop offset="1" stopColor="#4A6CF7" />

              </linearGradient>

              <linearGradient

                id="paint4\_linear\_25:218"

                x1="214.505"

                y1="10.2849"

                x2="212.684"

                y2="99.5816"

                gradientUnits="userSpaceOnUse"

              >

                <stop stopColor="#4A6CF7" />

                <stop offset="1" stopColor="#4A6CF7" stopOpacity="0" />

              </linearGradient>

              <radialGradient

                id="paint5\_radial\_25:218"

                cx="0"

                cy="0"

                r="1"

                gradientUnits="userSpaceOnUse"

                gradientTransform="translate(220 63) rotate(90) scale(43)"

              >

                <stop offset="0.145833" stopColor="white" stopOpacity="0" />

                <stop offset="1" stopColor="white" stopOpacity="0.08" />

              </radialGradient>

            </defs>

          </svg>

        </div>

      </section>

      <Services />

      <Templates />

      <Footer/>

    </>

  )

}

**Tashrifnoma yaratish componentasi:**

import { use, useEffect, useRef, useState } from "react";

import { BreadcrumbMenu } from "@/components/Header/BreadcrumbMenu";

import { ModeToggle } from "@/components/mode-toggle";

import { Tabs, TabsContent, TabsList, TabsTrigger } from "@/components/ui/tabs"

import { DownloadIcon, MixIcon, TextIcon, TransparencyGridIcon, Pencil2Icon } from "@radix-ui/react-icons"

import { ArrowLeftIcon, ArrowRightIcon, BanIcon, ImageIcon, Redo, Trash, Trash2Icon, Undo, XIcon } from "lucide-react";

import { useTheme } from "@/components/theme-provider"

import {

  ResizableHandle,

  ResizablePanel,

  ResizablePanelGroup,

} from "@/components/ui/resizable"

import { Input } from "@/components/ui/input";

import {

  Drawer,

  DrawerTrigger,

} from "@/components/ui/drawer"

import { FabricJSCanvas, useFabricJSEditor } from "fabricjs-react";

import { FabricImage } from "fabric";

import { Rnd } from "react-rnd";

import { Button } from "@/components/ui/button";

import { Card, CardContent } from "@/components/ui/card";

import { Label } from "@/components/ui/label";

import { BackgroundColors, BackgroundImages } from "./Backgrounds";

import { AddImages, AddShape, AddText } from "./AddImages";

import { Icons, IconRenderer } from "./Icons";

import { useParams } from "react-router-dom";

import { CardElementType } from "@/types/CardElement"

import ElementSetting from "./ElementSetting";

import jsPDF from "jspdf";

// eslint-disable-next-line @typescript-eslint/ban-ts-comment

// @ts-expect-error

import domtoimage from 'dom-to-image-more';

import FileSaver from 'file-saver';

export default function CreateVisitCard() {

  const { id } = useParams();

  const { theme } = useTheme();

  const [elements, setElements] = useState<CardElementType[]>([]);

  const [cardBackground, setCardBackground] = useState("#fff")

  const [mobileResponsive, setMobileResponsive] = useState(false);

  const [openLeftPanel, setOpenLeftPanel] = useState(true);

  const addElementHandler = (type: "text" | "image" | "icon", value: string) => {

    setElements([

      ...elements,

      {

        id: crypto.randomUUID(),

        type,

        content: type === "text" ? (value as string) : type === "image" ? "/images/img\_for\_cardImg/img1.png" : value,

        x: 50,

        y: 50,

        color: "#000000",

        fontSize: type == "text" ? 16 : 7,

        fontFamily: "Arial",

        fontWeight: "normal",

        fontStyle: "normal",

        textDecoration: "none",

        resizble: type === "image" ? true : false

      }

    ]);

  };

  //selected element for edit

  const [selectedElement, setSelectedElement] = useState<CardElementType>({ id: "", type: "text", content: "", x: 0, y: 0 });

  const updateElement = (id: string, newProps: { x: number; y: number; }) => {

    const updatedElements = elements.map((el) =>

      el.id === id ? { ...el, ...newProps } : el

    );

    setElements(updatedElements);

    saveHistory(updatedElements);

  };

  const selectelementHandler = (el: CardElementType) => {

    setSelectedElement({ ...el });

  }

  const history = useRef<string[]>([]);

  const historyIndex = useRef(-1);

  const saveHistory = (newState: CardElementType[]) => {

    if (historyIndex.current < history.current.length - 1) {

      history.current = history.current.slice(0, historyIndex.current + 1);

    }

    history.current.push(JSON.stringify(newState));

    historyIndex.current++;

  };

  const undo = () => {

    if (historyIndex.current > 0) {

      historyIndex.current--;

      setElements(JSON.parse(history.current[historyIndex.current]));

    }

  };

  const redo = () => {

    if (historyIndex.current < history.current.length - 1) {

      historyIndex.current++;

      setElements(JSON.parse(history.current[historyIndex.current]));

    }

  };

  const editElementDone = () => {

    const updatedElements = elements.map((el) => {

      if (el.id === selectedElement.id) {

        return { ...selectedElement }

      } else {

        return { ...el }

      }

    })

    setElements(updatedElements);

  };

  const downloadCardAsImage = async () => {

    editor?.canvas.renderAll();

    domtoimage.toBlob(document.getElementById('visit-card-preview')).then(function (blob: string | Blob) {

      FileSaver.saveAs(blob, "my-visit-card.png");

    });

    console.log(elements);

  };

  const downloadCardAsPDF = async () => {

    editor?.canvas.renderAll();

    domtoimage.toCanvas(document.getElementById('visit-card-preview')).then(function (canvas: HTMLCanvasElement) {

      const pdf = new jsPDF("landscape", "px", [canvas.width, canvas.height]);

      pdf.save("visit-card.pdf");

    });

  };

  const renderCard = () => {

    editor?.canvas.renderAll();

  }

  const { editor, onReady } = useFabricJSEditor();

  const AddShapeHandler = (text: "rectangle" | "circle" | "line") => {

    switch (text) {

      case "rectangle":

        editor?.addRectangle();

        break;

      case "circle":

        editor?.addCircle();

        break;

      case "line":

        editor?.addLine();

        break;

    }

  };

  const onRemoveShape = (type: "selected" | "all") => {

    const resultOfConfirm = window.confirm("Are you sure you want to delete?");

    if (!resultOfConfirm) return;

    if (type === "selected") {

      editor?.deleteSelected();

    } else {

      editor?.deleteAll();

    }

  }

  const onAddImage = async (imgUrl: string) => {

    const image = await FabricImage.fromURL(imgUrl);

    image.scaleToWidth(100);

    image.scaleToHeight(100);

    editor?.canvas.add(image);

  };

  const clearCanvas = () => {

    const resultOfConfirm = window.confirm("Are you sure you want to delete?");

    if (!resultOfConfirm) return;

    editor?.canvas.clear();

    editor?.deleteAll();

    setElements([]);

  }

  const [openEditElem, setOpenEditElem] = useState(false)

  const onRemoveElement = (id: string) => {

    const resultOfConfirm = window.confirm("Are you sure you want to delete?");

    if (!resultOfConfirm) return;

    const updatedElements = elements.filter((el) => el.id !== id);

    setElements(updatedElements);

    saveHistory(updatedElements);

  }

  //keydown handler

  useEffect(() => {

    const handleKeyDown = (e: { ctrlKey: unknown; key: string; preventDefault: () => void; }) => {

      // Ctrl + Y

      if (e.ctrlKey && e.key === 'y') {

        e.preventDefault();

        redo();

      }//Ctrl + Z

      if (e.ctrlKey && e.key === 'z') {

        e.preventDefault();

        undo();

      }

      if (e.key === 'Delete') {

        e.preventDefault();

        const resultOfConfirm = window.confirm("Are you sure you want to delete?");

        if (!resultOfConfirm) return;

        editor?.deleteSelected();

      }

    };

    window.addEventListener("keydown", handleKeyDown);

    return () => window.removeEventListener("keydown", handleKeyDown);

  }, [editor]);

  useEffect(() => {

    if(!id || isNaN(parseInt(id))) return;

    fetch("/templates/visit\_card.json")

      .then(res => res.json())

      .then(data => {

        const selected = data.find((t: { id: number; }) => t.id === parseInt(id));

        if(selected === undefined) return

        setCardBackground(selected.bg);

        setElements(selected.elements);

      });

  }, [id])

  useEffect(() => {

    setMobileResponsive(window.innerWidth < 768);

    const handleResize = () => {

      setMobileResponsive(window.innerWidth < 768);

    }

    window.addEventListener("resize", handleResize);

  }, [])

  return (

    <>

      {/\* header \*/}

      <div className="header border-b">

        <div className="container mx-auto p-2">

          <div className="flex md:justify-between md:flex-row flex-col items-center">

            <h1 className="text-xl font-bold text-body-color dark:text-body-color-dark md:text-3xl">Tashrif qog'oz</h1>

            <div className="justify-center mr-4 items-center mt-2">

              <BreadcrumbMenu currentPage="Tashrif qog'oz yaratish" />

            </div>

            <div className="absolute right-2 w-full md:w-auto flex justify-end md:relative">

              <ModeToggle />

            </div>

          </div>

        </div>

      </div>

      {/\* main workspace \*/}

      <div className="w-full flex" style={{ height: "calc(100vh - 50px)" }}>

        <ResizablePanelGroup direction="horizontal">

          <ResizablePanel minSize={25} maxSize={25} defaultSize={25} style={openLeftPanel? {} : { left: "0" }}>

            {/\* menu \*/}

            <Tabs defaultValue="addtext" className={`w-full md:relative h-full absolute z-100 bg-gray-800 ${openLeftPanel? "left-0" : "-left-110"} transition-all duration-300`}>

              <Button className={`absolute top-2 right-2 p-3 bg-red-500 ${openLeftPanel? "right-20": "right-10"}`} onClick={() => setOpenLeftPanel(false)}>{(mobileResponsive && openLeftPanel)? <ArrowRightIcon /> : <XIcon />}</Button>

              <div className="flex pt-2 md:pt-0">

                <TabsList className="md:w-20 min-h-max md:h-full gap-4 flex-col my-2 mx-0 md:p-0 p-2">

                  <TabsTrigger value="addtext" className="w-full h-full md:p-2 py-4 flex-col data-[state=active]:bg-blue-500 data-[state=active]:text-white rounded-md"><TextIcon /> <span className="md:block hidden">Text</span></TabsTrigger>

                  <TabsTrigger value="addicon" className="w-full h-full md:p-2 py-4 flex-col data-[state=active]:bg-blue-500 data-[state=active]:text-white rounded-md"><MixIcon /> <span className="md:block hidden">Icon</span></TabsTrigger>

                  <TabsTrigger value="addbg" className="w-full h-full md:p-2 py-4 flex-col data-[state=active]:bg-blue-500 data-[state=active]:text-white rounded-md"><TransparencyGridIcon /> <span className="md:block hidden">Background</span></TabsTrigger>

                  <TabsTrigger value="addimg" className="w-full h-full md:p-2 py-4 flex-col data-[state=active]:bg-blue-500 data-[state=active]:text-white rounded-md"><ImageIcon /> <span className="md:block hidden">Image</span></TabsTrigger>

                  <TabsTrigger value="export" className="w-full h-full md:p-2 py-4 flex-col data-[state=active]:bg-blue-500 data-[state=active]:text-white rounded-md"><DownloadIcon /> <span className="md:block hidden">Download</span></TabsTrigger>

                </TabsList>

                <div className="flex-1 p-2 ">

                  {/\* text \*/}

                  <TabsContent value="addtext" className="flex flex-col justify-center px-3">

                    <h4 className="text-2xl text-center">Yangi matn qo'shish</h4>

                    <AddText addElementHandler={addElementHandler} />

                  </TabsContent>

                  {/\* icon \*/}

                  <TabsContent value="addicon" className="flex flex-col justify-center px-3">

                    <Icons addElementHandler={addElementHandler} />

                  </TabsContent>

                  {/\* bg \*/}

                  <TabsContent value="addbg" className="flex flex-col justify-center px-3">

                    <h4 className="text-2xl text-center">Orqa fonni almashtirish</h4>

                    <Label className="mt-2">Rang tanlash</Label>

                    <Card className="my-2">

                      <CardContent className="flex flex-row flex-wrap justify-center gap-2 scroll-auto" >

                        <BackgroundColors setCardBackground={setCardBackground} />

                      </CardContent>

                    </Card>

                    <Label className="mt-2">Rasm tanlash</Label>

                    <Card className="my-2 h-85" style={{ overflowY: "auto", scrollbarWidth: "thin" }}>

                      <CardContent className="flex flex-row flex-wrap justify-center gap-2 scroll-auto" >

                        <BackgroundImages setCardBackground={setCardBackground} />

                      </CardContent>

                    </Card>

                  </TabsContent>

                  {/\* addimg \*/}

                  <TabsContent value="addimg" className="flex flex-col justify-center px-3">

                    <h4 className="text-2xl text-center">Rasm qo'shish</h4>

                    <Label className="mt-2">rasm tanlash</Label>

                    <Card className="my-2">

                      <CardContent className="flex flex-row flex-wrap justify-center gap-2 scroll-auto" >

                        <AddImages onAddImage={onAddImage} />

                      </CardContent>

                    </Card>

                    <Label className="mt-2">shakil tanlash</Label>

                    <Card className="my-2">

                      <CardContent className="flex flex-row flex-wrap justify-center gap-2 scroll-auto" >

                        <AddShape AddShapeHandler={AddShapeHandler} />

                      </CardContent>

                    </Card>

                  </TabsContent>

                  {/\* dowmload \*/}

                  <TabsContent value="export" className="flex flex-col justify-center px-3">

                    <h4 className="text-2xl text-center">Yuklab olish</h4>

                    <Button onClick={downloadCardAsImage} className="p-3 m-2 bg-blue-500 rounded flex"><DownloadIcon />Rasm sifatida yuklab olish</Button>

                    <Button onClick={downloadCardAsPDF} hidden className="p-3 m-2 bg-blue-500 rounded flex"><DownloadIcon />Download card as a PDF</Button>

                    <Button onClick={renderCard} hidden className="p-3 m-2 bg-blue-500 rounded flex"><DownloadIcon />render</Button>

                  </TabsContent>

                </div>

              </div>

            </Tabs>

          </ResizablePanel>

          <ResizableHandle className="cursor-ew-resize bg-black" />

          {/\* playground \*/}

          <ResizablePanel>

            <div className="w-full h-full flex justify-center items-center py-10 bg-gray-300 dark:bg-gray-500">

              <Card id="visit-card-preview" className="relative rounded-lg p-2 border border-gray-300 overflow-hidden w-full h-full" style={{ width: "577.612px", height: "324.800px", background: cardBackground.startsWith("url") ? `${cardBackground} center/cover no-repeat` : cardBackground }}>

                <FabricJSCanvas className="absolute top-0 left-0 w-[577px] h-full" onReady={onReady} />

                {elements.map((el) => (

                  <Rnd

                    key={el.id}

                    onDragStop={(\_, d) => updateElement(el.id, { x: d.x, y: d.y })}

                    default={{ x: el.x, y: el.y, width: "auto", height: "auto" }}

                    bounds="parent"

                    enableResizing={el.resizble}

                    className="absolute w-[577.612px] h-[324.800px]"

                    style={{ zIndex: 2, position: "absolute" }}

                  >

                    {el.type === "text" && <span style={{

                      border: "none",

                      outline: "none",

                      textDecoration: "none",

                      background: "transparent",

                      color: el.color, fontSize: el.fontSize, fontFamily: el.fontFamily, fontWeight: el.fontWeight, fontStyle: el.fontStyle, backgroundColor: "transparent", width: "auto", height: "auto"

                    }}

                    className="border-none">{el.content}</span>}

                    {el.type === "image" && <img src={el.content} alt="Image" className="w-12 h-12" />}

                    {el.type === "icon" && <IconRenderer iconName={el.content} color={el.color} size={el.fontSize} className="bg-transparent border-none" />}

                  </Rnd>

                ))}

              </Card>

            </div>

          </ResizablePanel>

          <ResizableHandle className="cursor-ew-resize bg-black" />

          {/\* left panet shows all texts \*/}

          <ResizablePanel minSize={17} maxSize={17} defaultSize={17} className="flex-col" >

            <Drawer open={openEditElem} onOpenChange={setOpenEditElem}>

              <div className="flex-col h-100">

                <h4 className="text-2xl text-center">Boshqarish</h4>

                <div className="flex-col border-b">

                  <div className="flex justify-between">

                    <div className="flex">

                      <Button onClick={undo} title="Undo (Ctrl + Z)" className="p-1 m-1 bg-blue-400"><Undo /></Button>

                      <Button onClick={redo} title="Redo (Ctrl + Y)" className="p-1 m-1 bg-blue-400"><Redo /></Button>

                    </div>

                    <div className="flex">

                      <Button className="my-2 mx-1 bg-red-500" onClick={() => onRemoveShape("selected")} title="Remove selected shape or image"><BanIcon /></Button>

                      <Button className="my-2 mx-1 bg-red-500" onClick={() => onRemoveShape("all")} title="Remove all shape and image"><Trash /></Button>

                      <Button onClick={clearCanvas} className="my-2 bg-red-500" title="Clear All elements"><Trash2Icon /></Button>

                    </div>

                  </div>

                </div>

                <div className="flex-col p-2 justify-center">

                  {

                    elements.map((el, index) => (

                      <div className="flex w-full my-2" key={index}>

                        <Label>

                          {

                            el.type == "text" ? "matin" : "Icon"

                          }

                        </Label>

                        {

                          el.type === "text" ?

                            <Input type="text" className="mx-2" disabled value={el.content} /> :

                            <div className="my-2 mx-4">

                              <IconRenderer iconName={el.content} color={theme == "dark" ? "#fff" : "#000"} size={5} className={""} />

                            </div>

                        }

                        <DrawerTrigger className="hover:cursor-pointer"><Pencil2Icon className="text-sky-500" onClick={() => selectelementHandler(el)} /></DrawerTrigger>

                        <Button className="hover:cursor-pointer" onClick={() => onRemoveElement(el.id)}><Trash2Icon className="text-red-800" /></Button>

                      </div>

                    ))

                  }

                </div>

              </div>

              {/\* Drawer window for open element settings \*/}

              <ElementSetting editElementDone={editElementDone} selectedElement={selectedElement} setSelectedElement={setSelectedElement} openEditElem={openEditElem} />

            </Drawer>

          </ResizablePanel>

        </ResizablePanelGroup>

      </div>

    </>

  );

}

**WEB SAYT KO’RINISHI**

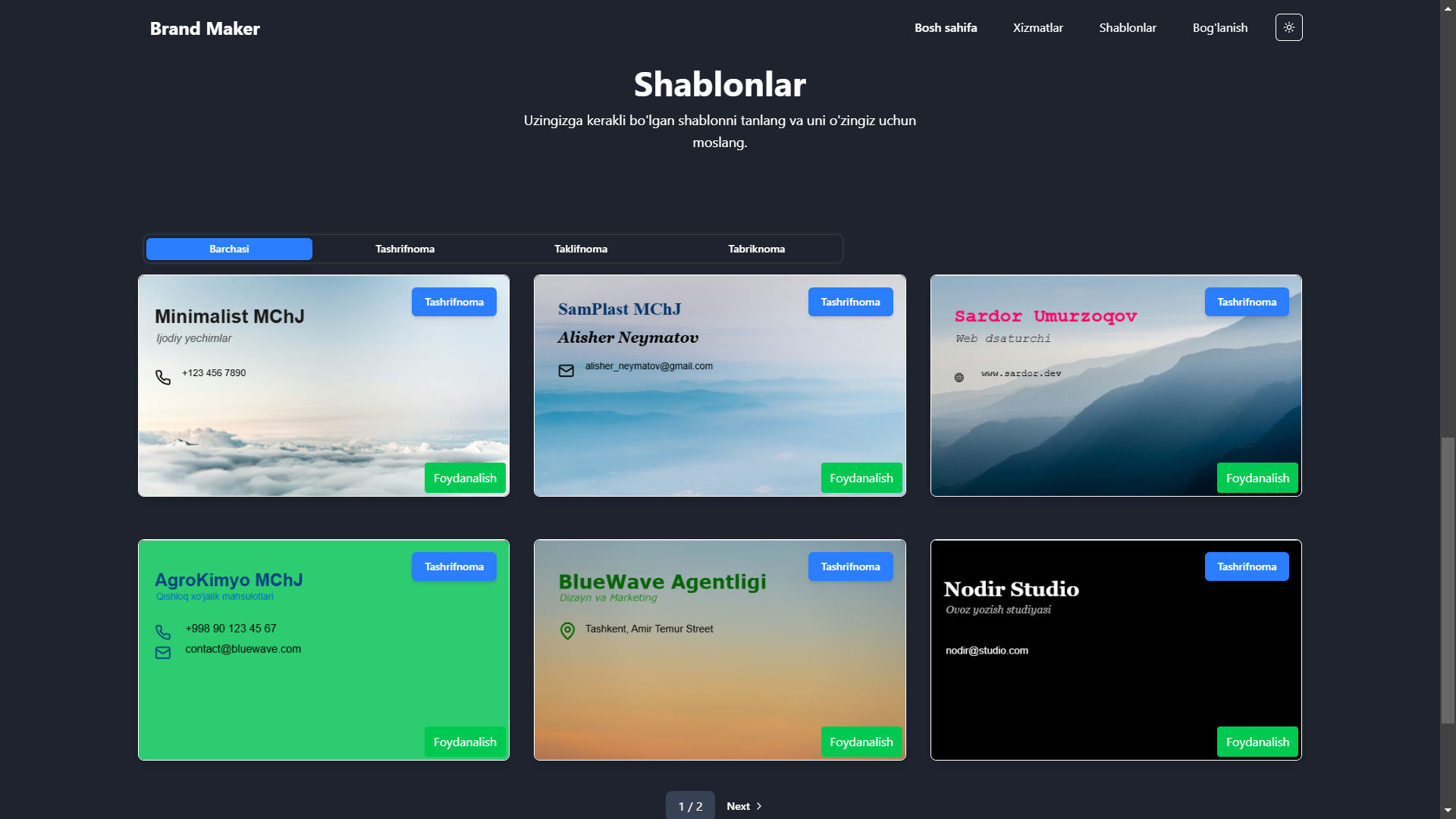
Bosh sahifa

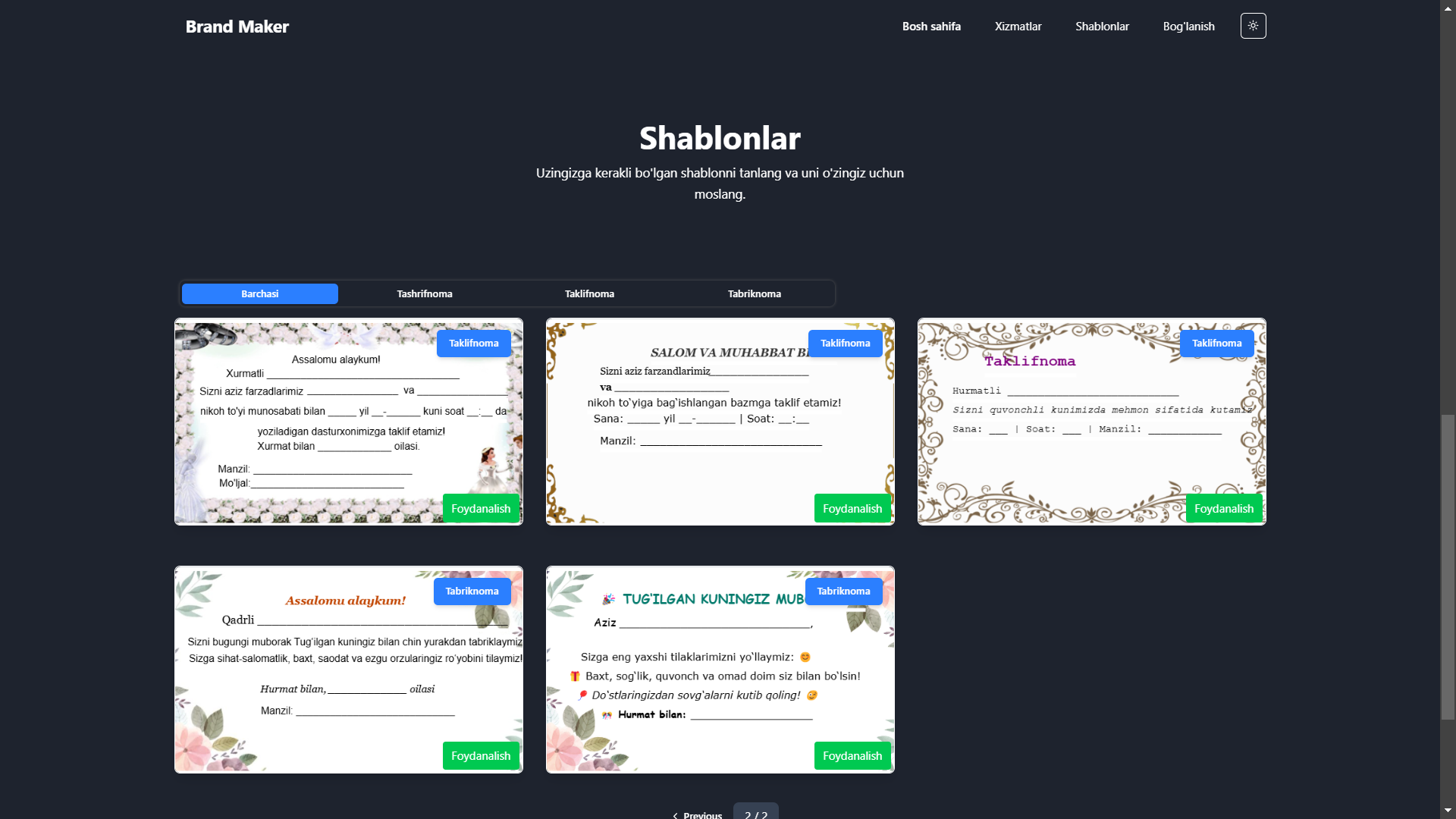
****

Xizmatlar

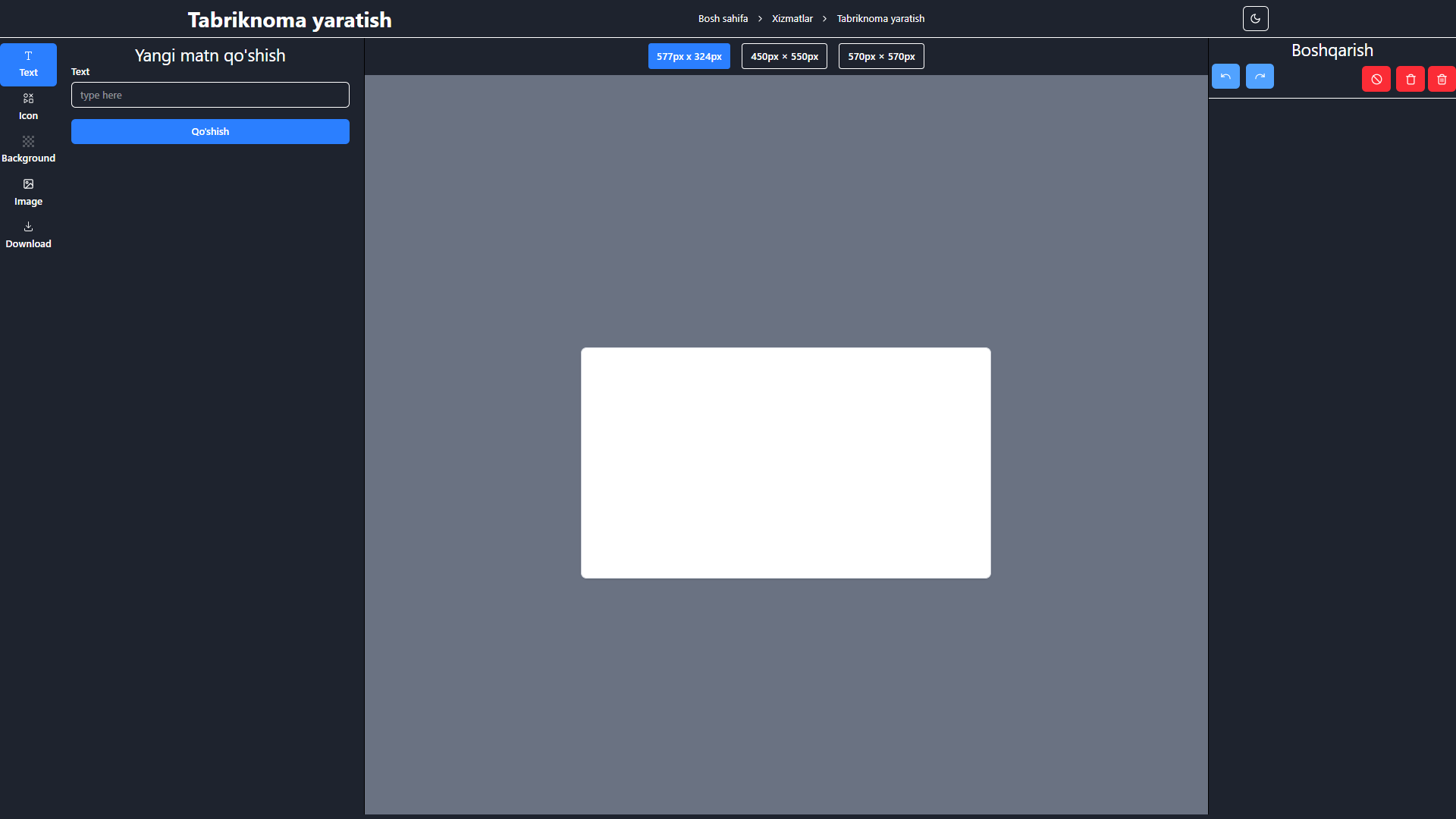
****

Shablonlar

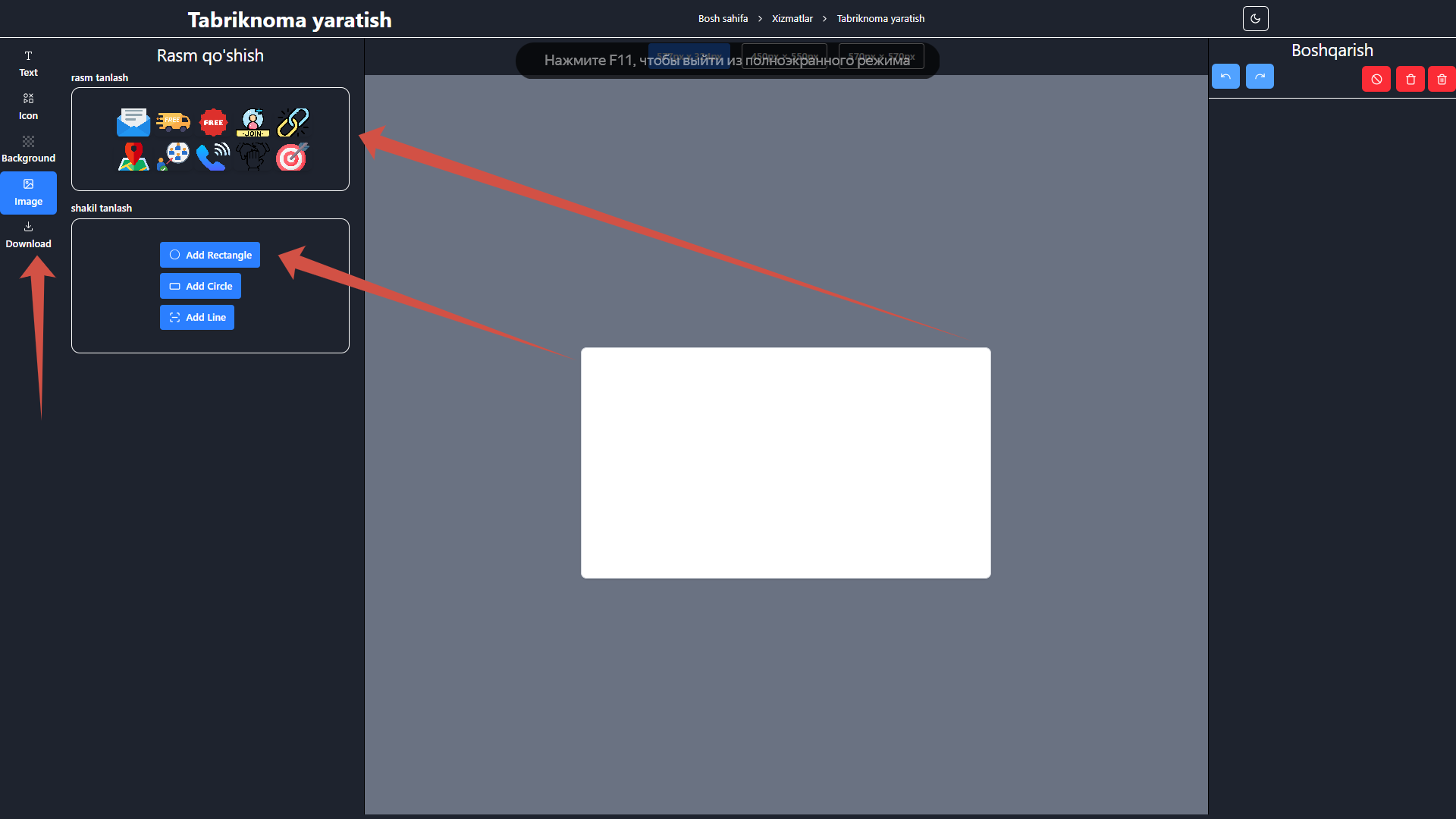
****

****

Tabriknoma yaratish

****

Bu yerda tabriknomaga rasm, text va iconlar qo’shsak bo’ladi

****