

UNIVERZA V MARIBORU
FAKULTETA ZA ELEKTROTEHNIKO,
RAČUNALNIŠTVO IN INFORMATIKO

DOKUMENTACIJA
ODLOČITEV

Tehnična dokumentacija

Filip Polner
Saša Vinčič
Gašper Lukacs

BlockPermit
Maribor, april
2023

1 ARHITEKTURNE ODLOČITVE.....	2
1.1 Arhitektura aplikacije.....	2
1.1.1. Next.js	2
1.1.2. Prisma	2
1.1.3. TypeScript.....	3
1.1.4. Tailwind.....	3
1.1.5. Supabase	3
1.1.6. MongoDB	4
1.1.7. Atlas	4
1.1.8. Mailgun	4
1.1.9. Vercel	4
1.1.10. Ethereum Blockchain	5
1.1.11. Solidity.....	5
1.1.12. Ganache.....	5
1.1.13. Hardhat	5
1.1.14. Metamask.....	5
1.1.15. Postman	5
1.1.16 Ethers.js.....	6
1.1.17. Github.....	6
1.1.18. Discord	6
1.1.19. Jira	6
2 ORGANIZACIJA DELA	7

1 ARHITEKTURNE ODLOČITVE

1.1 Arhitektura aplikacije

Aplikacijo smo zgradili s pomočjo naslednjih tehnologij:

- Next.js
- Prisma
- Typescript
- TailwindCSS
- Supabase
- MongoDB
- Mailgun
- Vercel
- Ethereum blockchain
- Solidity
- Ganache
- Hardhat
- Metamask
- Postman
- Ether.js
- Github
- Discord
- Jira

1.1.1. Next.js

Next.js je ogrodje za razvoj spletnih aplikacij z uporabo Reacta. Zagotavlja izjemno zmogljivost in hitrost nalaganja strani, saj predrenderira strani na strežniku. Prav tako ponuja enostavno upravljanje stanja aplikacije in dobro podporo za SEO (optimizacija za iskalnike), kar sicer ni bilo tako relevantno za naš "usecase". Poleg vseh teh prednosti pride z ogromno vgrajenimi orodji za routing, vzdrževanje uporabnikove seje ipd.

1.1.2. Prisma

Prisma je ORM (objektno-relacijsko preslikovanje) in sklad podatkovne baze za razvoj spletnih aplikacij. Prisma olajša delo z bazami podatkov, saj avtomatsko generira tipizirane modele podatkov in zagotavlja preprost dostop do podatkov. Izbrali smo zaradi enostavne uporabe, ki jo zagotavlja.

1.1.3. TypeScript

TypeScript: TypeScript je nadmnožica JavaScripta, ki podpira statično tipizacijo. To pomeni, da morajo spremenljivke, funkcije in objekti imeti določen tip. Prednosti tega so večja jasnost, manj napak pri programiranju in izboljšana berljivost in vzdrževanje kode. Zaradi teh značilnosti je TypeScript še posebej uporaben pri velikih in kompleksnih projektih. Tukaj je bil "no brainer".

1.1.4. Tailwind

Tailwind CSS je nizko nivojski okvir za CSS, ki omogoča gradnjo prilagojenih oblik z uporabo pripomočkov. Vsak pripomoček služi kot atomska CSS razreda, kar omogoča večjo fleksibilnost pri oblikovanju. Z Tailwind CSS lahko zlahka ustvarite edinstvene in odzivne oblike. Uporabili smo ga predvsem, zato da smo se izognili nepotrebnim količinam CSS datotek.

1.1.5. Supabase

Supabase je odprtokodna alternativa Firebase, ki vključuje značilnosti, kot so avtentikacija, shranjevanje podatkov in poizvedbe v realnem času, ter omogoča enostavno in hitro izgradnjo spletnih aplikacij. Supabase podpira PostgreSQL bazo, ki je znana po svoji robustnosti in zmogljivosti, kar ga naredi odlično izbiro za zapletene projekte. Uporabili smo ga zaradi zelo enostavnega StorageAPI-ja, shranjevanje dokumentov in sama nastavitvev klienta je izjemno enostavna.

1.1.6. MongoDB

MongoDB je odprtokodna, dokumentno usmerjena baza podatkov, ki je zasnovana za enostavnost razvoja in razširljivost. Omogoča shranjevanje podatkov v fleksibilnih, JSON podobnih dokumentih, kar pomeni, da se polja lahko razlikujejo med dokumenti in struktura podatkov lahko sčasoma spremeni. Izbrali smo jo saj smo imeli dosti nepovezanih podatkov, večina podatkov pa se je tako pridobila direktno iz blockchaina.

1.1.7. Atlas

MongoDB Atlas je oblachna storitev, ki samodejno obvladuje infrastrukturo, setup in administracijo baze podatkov. Omogoča razvijalcem, da se osredotočijo na pisanje kode, namesto na upravljanje baze podatkov. Poleg tega ponuja varnostne, varnostne kopije in obnovitvene funkcije, kar zagotavlja stabilnost in zanesljivost aplikacij. Super za shranjevanje in dostopanje do podatkov v bazi.

1.1.8. Mailgun

Mailgun je orodje za avtomatizacijo e-pošte, ki omogoča razvijalcem pošiljanje, prejemanje in sledenje e-pošti znotraj svojih aplikacij. Omogoča tudi napredne funkcionalnosti, kot so preverjanje pristnosti e-pošte, sledenje dogodkom in analitiko. Zaradi svoje zanesljivosti in učinkovitosti je odlična izbira za aplikacije, ki potrebujejo robustne e-poštnih storitev. Uporabili smo jih ker imajo enostaven API za pošiljanje e-pošte.

1.1.9. Vercel

Vercel je oblachna platforma za uvajanje frontend in strežniških JavaScript aplikacij. Podpira Next.js, omogoča hitro uvajanje in optimizacijo za najboljše delovanje spletnih strani. Vercel se osredotoča na razvijalčevo izkušnjo z zagotavljanjem funkcij, kot so samodejno skaliranje, CDN, SSL in še več. Uporabili smo jih saj je tukaj najlažje deployati Next.js aplikacijo.

1.1.10. Ethereum Blockchain

Ethereum je decentralizirana platforma, ki omogoča razvijalcem ustvarjanje in uvajanje pametnih pogodb. To so samodejno izvršljivi programi, ki tečejo na Ethereum blockchainu brez potrebe po tretji strani. Ethereum omogoča izdelavo decentraliziranih aplikacij (dApps) in kripto valut, kar odpira nove možnosti za transparentne, nepreklicne in varne digitalne transakcije. Ethereum smo uporabili za shranjevanje in ohranjanje integritete naših podatkov. Podatki so vedno transparentni in preverljivi na blockchainu.

1.1.11. Solidity

Solidity je objektno usmerjen programski jezik, ki je zasnovan za pisanje pametnih pogodb na Ethereum blockchainu. Solidity omogoča razvijalcem, da napišejo programe, ki manipulirajo z digitalno vrednostjo, izvajajo transakcije in interagirajo z drugimi pametnimi pogodbami, kar je bistvenega pomena za izgradnjo dAppov in drugih projektov na Ethereumu.

1.1.12. Ganache

Ganache je osebno blockchain okolje za Ethereum, ki je namenjeno testiranju in razvoju. Omogoča razvijalcem, da uvajajo pametne pogodbe, izvajajo transakcije, pregledajo stanje blockchaina in simulirajo delovanje v realnem okolju. To je bistveno orodje za razvijalce, ki delajo z Ethereumom, saj zagotavlja hitro in varno testiranje. Ganache smo uporabili za gostovanje ter testiranje našega blockchaina na lokalnem okolju.

1.1.13. Hardhat

Hardhat je razvojno okolje za Ethereum, ki olajša razvijalcem delo s pametnimi pogodbami. Ponuja vrsto funkcij, kot so upravljanje z lokalnimi in testnimi omrežji, avtomatizirano testiranje, debugiranje in možnost prilagajanja okolja. Ta prilagodljivost in vsestranskost naredi Hardhat odlično izbiro za razvoj kompleksnih Ethereum projektov. Hardhat smo uporabili za izvajanje skript in prilagoditev na blockchainu.

1.1.14. Metamask

MetaMask je brskalnik in mobilna razširitev, ki deluje kot digitalna denarnica za Ethereum in omogoča interakcijo z decentraliziranimi aplikacijami (dApps) neposredno iz brskalnika. Uporabniki lahko hranijo Ether in druge žetone na Ethereumu, izvajajo transakcije in upravljajo svoje digitalne identitete. Za razvijalce je MetaMask pomembno orodje, ki omogoča njihovim uporabnikom enostavno interakcijo z njihovimi dApps. Preko Metamask-a uporabnik komunicira z našo aplikacijo.

1.1.15. Postman

Postman je platforma za testiranje API-jev, ki omogoča razvijalcem sestavljanje, testiranje in spremljanje API-jev. Ponuja funkcije, kot so ustvarjanje zahtevkov, avtomatizacija testov in dokumentacija, kar pomaga pri hitrem in učinkovitem razvoju API-jev.

1.1.16 Ethers.js

Ether.js je knjižnica, ki omogoča interakcijo z Ethereum blockchainom. Ponuja vrsto funkcij, ki omogočajo delo s pametnimi pogodbami, izvajanje transakcij in branje podatkov iz blockchaine. Ether.js je znan po svoji varnosti in prenosljivosti, kar ga naredi odlično izbiro za razvijalce Ethereum.

1.1.17. Github

Github je platforma za souporabo kode, ki temelji na sistemu za nadzor različic Git. Omogoča razvijalcem, da sodelujejo na projektih, sledijo spremembam, uvajajo novo kodo in pregledujejo kodo drugih. Github je ključnega pomena za sodelovanje in učinkovito upravljanje projektov.

1.1.18. Discord

Discord je platforma za komunikacijo, ki omogoča takojšnje sporočanje, glasovno in video komunikacijo ter deljenje datotek. Razvijalci pogosto uporabljajo Discord za komunikacijo v skupini, sodelovanje na projektih in gradnjo skupnosti. Discord podpira integracije z drugimi orodji, kot so Github in Jira, kar omogoča boljše sodelovanje in produktivnost.

1.1.19. Jira

Jira je orodje za sledenje napak in upravljanje projektov, ki je priljubljeno v razvojni skupnosti. Omogoča sledenje nalog, napak in mejnikov ter omogoča vizualizacijo napredka preko kanbana ali Scrum plošč. Jira je ključnega pomena za upravljanje in organizacijo projektov, zlasti pri delu v večjih skupinah. Jira smo uporabili za upravljanje našega projekta.

2 ORGANIZACIJA DELA

V naši skupini smo poskušali doseči čimbolj učinkovito komunikacijo in jasno razdelitev odgovornosti. Ključno orodje za našo medsebojno komunikacijo je bil Discord, kjer smo hitro in učinkovito delili informacije, razpravljali o težavah in se dogovarjali za sestanke. Za sledenje nalog in napredka projekta smo uporabljali orodje Jira, ki nam je omogočalo vizualno predstavitev nalog, njihovega stanja in kdo je zanje odgovoren.

Odgovornosti so bile jasno razdeljene med člani skupine. Vsak od nas je prevzel vodilno vlogo v določenih aspektih projekta, kar je omogočilo, da je vsak član lahko izkoristil svoje znanje in veščine. Prav tako smo imeli enega člana, ki je bil določen za koordinatorja skupine in je skrbel za to, da smo ostali na pravi poti in sledili rokom.

Razdelitev dela je potekala na način, da smo na začetku projekta določili ključne mejnike in nato razdelili naloge med člane glede na njihove spretnosti in interese. Ta pristop nam je omogočil, da smo bili vsi vključeni in motivirani za doseganje ciljev projekta.

Imeli smo redne sestanke z našim mentorjem, kjer smo pregledali napredek, razpravljali o izzivih in prejeli dragocene povratne informacije. Ti sestanki so bili ključni za naš napredek in uspešnost projekta.

Prav tako smo vzpostavili redne sestanke s stranko, kjer smo predstavili naš napredek, razpravljali o njenih povratnih informacijah in potrdili, da smo na pravi poti za izpolnitev njenih pričakovanj.

Za vodenje našega projekta smo uporabili pristop Waterfall, kar pomeni, da smo projekt izvajali v zaporednih fazah od začetne analize do končne implementacije in vzdrževanja. Ta pristop nam je omogočil, da smo načrtovali, kaj potrebujemo storiti in kdaj, ter ohranili jasen pregled nad celotnim potekom projekta.

Za sledenje spremembam kode in sodelovanje pri razvoju kode smo uporabljali Git. Ta sistem za nadzor različic nam je omogočil, da smo hitro in enostavno sodelovali na isti kodi, ob tem pa ohranili sledljivost sprememb in zagotovili, da se napake lahko hitro odkrijejo in popravijo.