**SVEUČILIŠTE U SPLITU**

**SVEUČILIŠNI ODJEL ZA STRUČNE STUDIJE**

Specijalistički diplomski stručni studij Primijenjenog računarstva

**Blockexplorer**

Kriptovalute

**Student:** David L. Tomasović **Datum:** 19.2.2022.

**Sadržaj**

[Opis projekta i korištene tehnologije 2](#__RefHeading___Toc179_3482498902)

[Backend 3](#__RefHeading___Toc181_3482498902)

[getTransactionInfo 4](#__RefHeading___Toc183_3482498902)

[GetBlockInfo 5](#__RefHeading___Toc185_3482498902)

[Server 5](#__RefHeading___Toc187_3482498902)

[Rute 6](#__RefHeading___Toc189_3482498902)

[Frontend 7](#__RefHeading___Toc191_3482498902)

[Fetch 7](#__RefHeading___Toc692_3606030947)

[Komponente 8](#__RefHeading___Toc694_3606030947)

[Input 8](#__RefHeading___Toc696_3606030947)

[SimpleKeyValPair 8](#__RefHeading___Toc698_3606030947)

[BlockResults 9](#__RefHeading___Toc700_3606030947)

[TransactionResults 9](#__RefHeading___Toc702_3606030947)

[FAB 10](#__RefHeading___Toc704_3606030947)

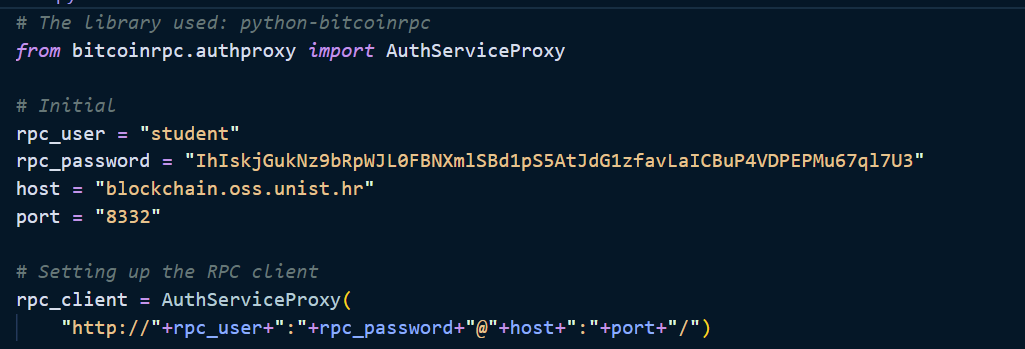
# Opis projekta i korištene tehnologije

U ovom projketnom zadatku, sam odarao napraviti aplikaciju Blockexplorer. Ovaj projektni zadatak se sastoji od 2 dijela, backend i frontend. Backend je napravljen koristeći Python kao programski jezik a za napravit API je korišten Flask modul, uz bitcoin-rpc za pristup na Bitcoin Core. Za frontend dio aplikacije je korišten React kao frontend framework za JavaScript programski jezik. Kako bih pripomoglo sa stiliziranjem aplikacije se koristio SASS (Stylistically Awesome Style Sheets), koji proširuje funkcionalnosti normalnog CSS-a i dopušta ugnježđenje css pravila.

# Backend

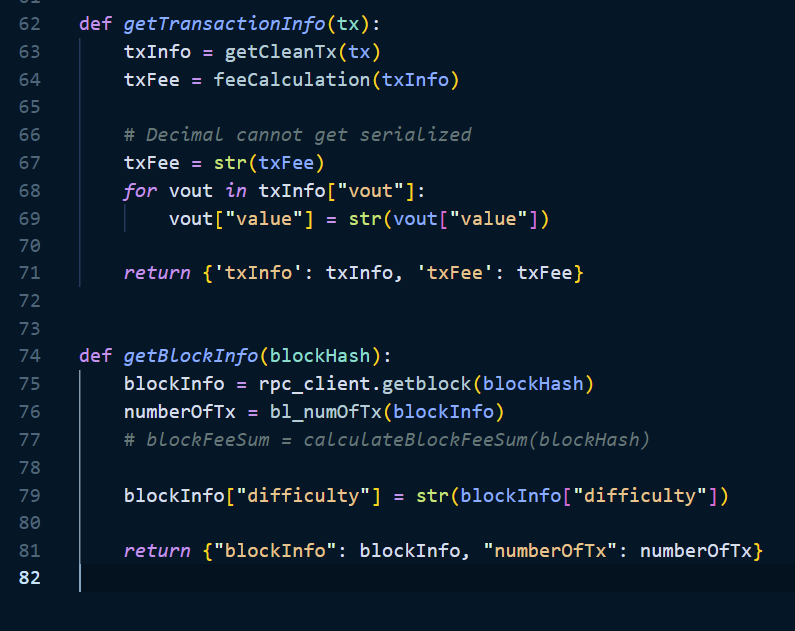
Za backend smo koristili Python sa Flask modulom za kreirat API pomoću kojeg će naš backend radit. Za pristup Bitcoin Core koristimo bitcoin-rpc modul za python.

Koristili smo node koji imamo instalirano u sveučilištu, pristupni podaci su vidljivi na slici Pristupni podaci

Figure 1: Pristupni podaci

Ove podatke poslije importamo u drugu datoteku unutar koje se nalaze sve funckije za pristup Bitcoin mreži.

Dvije glavne funkcije koje su se koristile za pristup blokovima(getBlockInfo) i tranzakcija (getTransactionInfo). Svaka od ovih funkcija se sastoji od manjih pod funckija. Ove dvije funckije možemo vidjeti na slici Block funkcije

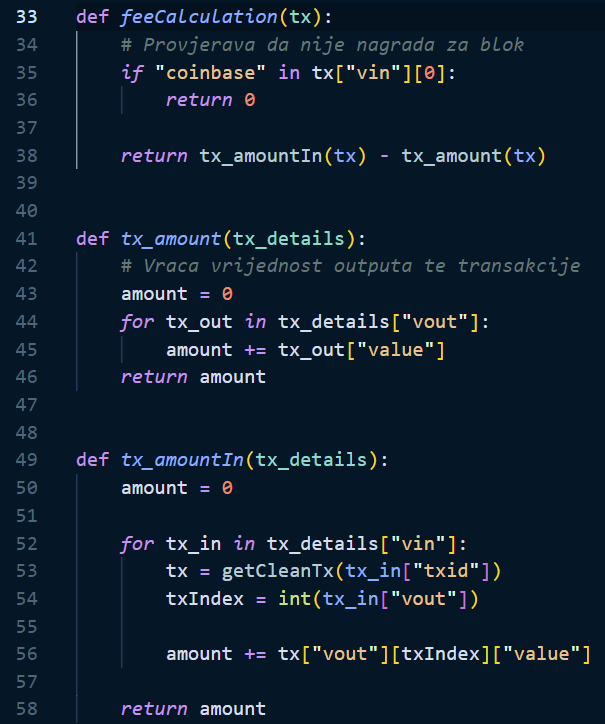
Figure 2: Block funkcije

## getTransactionInfo

Unutar funkcije prosljeđujemo hash tranzakcije, koju ćemo poslije dobiti sa front end dijela aplikacije.

Potrebno je prosljeđeni hash pretvoriti u nešto koristivo nama, tako da prosljeđujemo taj hash na funkciju GetCleanTx što komunicira sa bitcoin-rpc modulom koji pretraži bitcoin mrežu i povratna vrijednost bude JSON sa svim podacima o toj tranzakciji.

Zatim kako bih otkrili koji je fee na toj tranzakciji prosljeđujemo dobiveni JSON u funckiju feeCalculation. Funkciju možemo vidjeti na slici feeCalculation . Unutar koje provjeravamo ako je tranzakcija coinase, pa ju preskacemo. A fee se računa oduzimanjem ulazne vrijednosti i izlazne vrijednosti.

Figure 3: feeCalculation

## GetBlockInfo

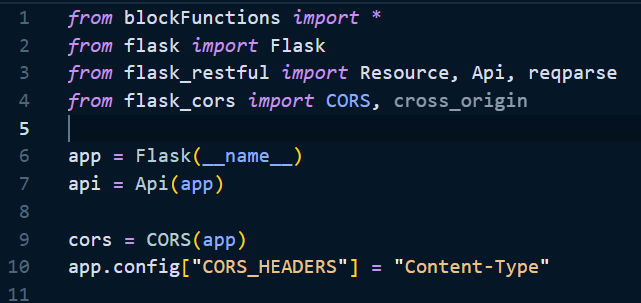
Unutar ove funkcije pomoću bitcoin-rpc modula prosljeđujemo blockhash i slično kao i prijašnja funkcija nam vraća JSON sa podacima tog bloka.

Provjeravat ćemo koliko tranzakcija ima unutar tog bloka i ujedno dobit informacije o samom bloku.

## Server

Za napraviti Server smo koristili Flask modul koji nam omogućava brzo i jednostavno konfiguriranje API-a koji nam je potreban kako bih koristili naš Python kod u JS-u.

Kako bi se kreirao API potrebno nam je prvo uključiti sve potrebne module u datoteku, možemo vidjeti na slici Server import. Potreban nam je bio CORS kako bih mogli sa različitih portova pristupit podacima, jer po početnim postavkama to ijje moguće zbog sigurnosnih napada.

Figure 4: Server import

### Rute

Imamo dvije glavne rute za dohvaćanje tranzakcija i dohvaćanje blokova. Kako bih prosljedili podatke u GET request prosljeđujemo podatak preko query parametra, koju definiramo (linija 18 i 34) i tako dohvaćamo vrijednosti i prosljeđujemo ih na našu getTransactionInfo ili getBlockInfo funkcijama.

Praktički cijeli request moramo zamotati u try...catch bloku kako u slučaju da bude nekakav error, kako bih sprijecili zatvaranje API-a.

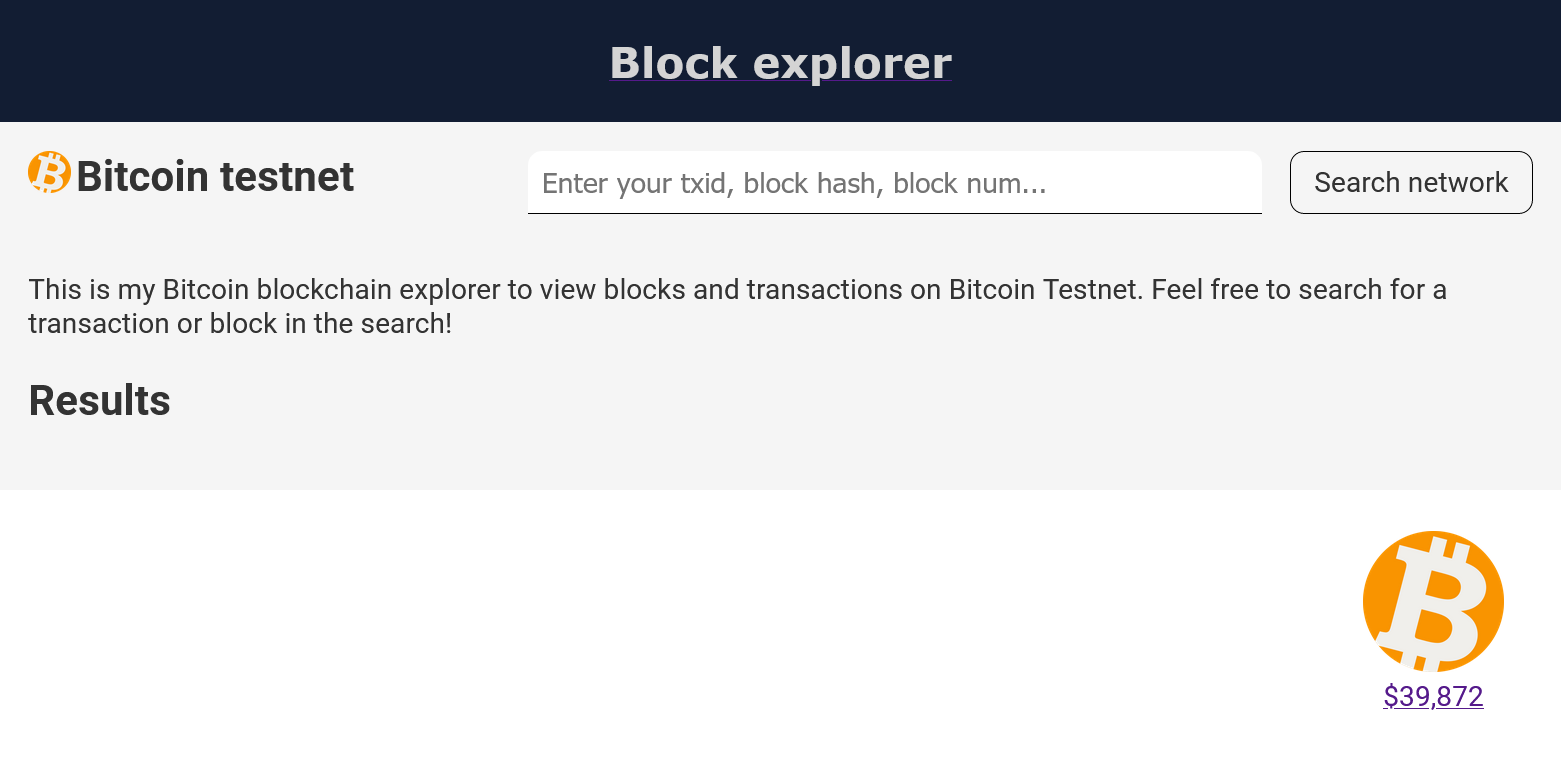
Možemo vidjeti sliku ruta na slici Server rute

Figure 5: Server rute

# Frontend

Za frontend smo koristili React framework. Prednosti korištenja Reacta preko nekog drugog frontend frameworka je to što koristi komponente koje se mogu ponovno iskoristiti. Početna stranica je prikazana na slici Početna stranica.

Neki od glavnih elemenata na ovoj stranici su cijena bitcoina u donjem desnom kutu, input field preko kojeg korisnici upisuju svoj blockhash ili tranzakcijski hash.

Figure 6: Početna stranica

## Fetch

Kako bih dohvatili podatke koje su podjeljene na API-u na backendu. JavaScript ima svoju integriranu fetch funkciju u koju se upisuje URL za postavljeni API. Dohvaćanje blokova i tranzakcija je prikazano na slici Fetch funkcije.

Nakon dohvaćanja podataka koristeći useState hook-ove (funkcionalnost koja „nacrta” cijeli zaslon iz početka sa novim podacima) postavljamo te vrijednosti.

Figure 7: Fetch funkcije

## Komponente

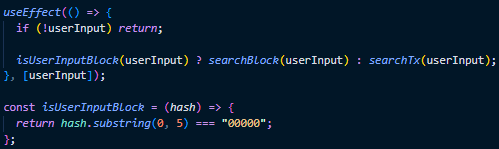
Komponente u Reactu su „med i mlijeko” React-a. To su manji dijelovi aplikacije koji se mogu nanovo koristit na drugim mjestima. Oni mogu sadržavati svoje varijable, funkcije i izgled. A kada se u komponentu želi prosljediti vrijednost, to zovemo „**prop**”.

Neke od glavnih komponenti koje su korištene u ovoj aplikaciji su:

* Input
* SimpleKeyValPair
* BlockResults
* TransactionResults
* FAB

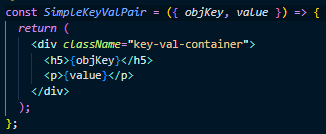
### Input

Prilikom upisivanja nekog teksta unutar input-a unos se ubacuje u useState. Koristeći useEffect (React hook koji izvrsava neki kod ako dođe do promjene varijable koja je prosljeđena kao parametar) provjerava se unos korisnika kako bih se otkrilo ako je to blok ili tranzakcijski hash. Na slici Korinički unos provjera je prikazana funkcija koja to provjerava. Nakon što se otkrije ako je to pretraživanje bloka ili tranzakcije se izvrši prijašnje naveden fetch

Figure 8: Korinički unos provjera

### SimpleKeyValPair

Ova komponenta, iako najjednostavnija se najviše koristi u aplikaciji. Cijelu komponentu možemo vidjeti na slici SimpleKeyValPair komponenta. Kao svoje props dobije vrijednost koja će biti prikazana sa lijeve strane i vrijednost koja se prikaze sa desne strane.

Figure 9: SimpleKeyValPair komponenta

### BlockResults

Ako korisnik je upisao hash bloka, onda će se pokrenuti funkcija koja će pretražiti bitcoin-rpc za blok sa tim hashom i vratit povratnu vrijednost za taj blok.

Cijelu komponentu možemo vidjeti na slici Block result komponenta. Sama komponta nije pre velika, već se izlistaju svi ključevi u prosljeđenom JSON-u i prikaže pomoću SimpleKeyValPair komponte.

Unutar komponente je moguće kliknuti na tranzakciju kako bih se učitala iduća komponenta.

Figure 10: Block result komponenta

### TransactionResults

Slična kao i prijašnja komponenta, ova komponenta se poziva prilikom upisa tranzakcije u input polje. Ova komponenta je malo veća nego blok funkcija ali uključuje jako slične elemente.

Unutar komponente isto koristimo SimpleKeyValuePair komponentu za prikaz podataka, no unutar ove komponente imamo neke podatke iz JSON-a koje bi htjeli na malo drugačiji način prikazati, npr. htjeli bi da su neki podaci stavljeni sa malo više razmaka, da je klikablilno itd.

Sliku prikaz izlaznih tranzakcija unutar ove komponente možemo vidjeti Izlazne tranzakcije, TransactionsResults. Ovdje prolazimo kroz JSON i odabiremo vrijednosti koje želimo prikazati i koristimo komponentu CleanKeyVal koja je identična kao i SimpleKeyVal samo uz drukčije stilove.

.

Figure 11: Izlazne tranzakcije, TransactionsResults

### FAB

Ova komponenta se nalazi na samom donjem desnom kutu aplikacije, i ona je zadužena za prikaz cijene Bitcoin-a. Kako bih dobili trenutnu cijenu Bitcoin-a koristimo besplatni, javno dostupni API od coingecko.com . Prikaz cijele komponente možemo vidjeti na slici FAB komponenta.

Unutar same komponente imamo fetch funkciju koja dobavlja trenutnu cijenu i koristeći useState ubacuje vrijednost ispod ikone Bitcoina.

Klikom na tu ikonu smo odvedeni na stranicu coingecko gdje vidimo više detalja o bitcoinu.

Figure 12: FAB komponenta