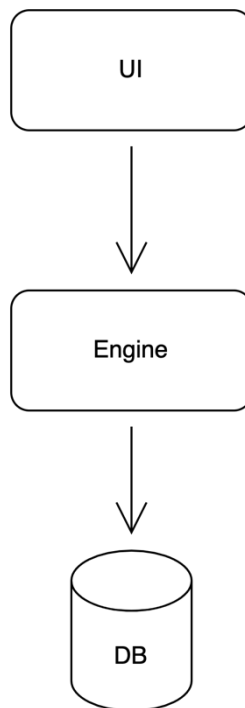


Projektni zadatak DRES Crypto Portfolio

Implementirati projekat koji simulira lični kripto portfolio. Portfolio predstavlja način da se prati koliku trenutnu vrednost korisnik ima u kriptovalutama.

Implementacija treba da sadrži 3 komponente:

1. Korisnicki interfejs (UI)
2. Servis za obradu zahteva i podataka (Engine)
3. Bazu podataka (DB)



Skica sistema

Korisnicki interfejs (UI)

Korisnicki interfejs je Flask web aplikacija koja treba da opsluzi korisnika u interakciji sa platnim prometom.

Akcije koje treba podrzati na korisnickom interfejsu su:

1. Registracija novog korisnika
2. Logovanje postojećeg korisnika
3. Izmena korisnickog profila
4. Pregled kriptovaluta koje korisnik ima i koliko ima u vlasništvu

5. Unos nove kupovine kriptovalute ili prodaje koja je obavljena nekog dana
6. Brisanje neke kriptovalute ili neke kupovine - prodaje
7. Prikaz po kriptovaluti profita ili gubitka i vrednost u toj kriptovaluti
8. Prikaz ukupne vrednosti svih kriptovaluta i ukupan profit ili gubitak

Korisnik se registruje unoseci:

1. Ime
2. Prezime
3. Adresa
4. Grad
5. Drzava
6. Broj telefona
7. Email
8. Lozinka

Korisnike se loguje putem:

- Email
- Lozinka

Novi korisnik dolazi na inicijalnu stranicu koja zapravo predstavlja portfolio. Inicijalno korisnik nema ništa u svom portfolio i ima mogućnost da unese informacije o nekoj obavljenoj kupovini kriptovalute ili prodaji, pri čemu prva stvar za neku kriptovalutu mora biti kupovina.

Korisnik unosi novu kupovinu ili prodaju pri čemu unosi kriptovalutu koju je kupio, datum i vreme kada je ta kriptovaluta kupljena kao i vrednost u dolarima koliko je kriptovaluta plaćena.

Svakim novim unosom ponovo se izračunava sama dobit za datu kriptovalutu kao i novo ukupno stanje na nivou portfolia.

Računanje novog iznosa portfolia je neophodno odraditi u zasebnom procesu nakon azuriranja kriptovalute.

Neophodno je omogućiti korisniku pretragu svih kriptovaluta. Informacije o vrednosti kriptovaluta u nekom momentu je neophodno dovlaci sa eksternih API-a poput kriptomenjačnica (poput Binance, CEX.io).

Servis za obradu zahteva i podataka (Engine)

Engine je servis implementiran kao flask API aplikacija. Engine ima svoje endpointe koje prikazuje eksternom svetu (UI aplikaciji) za koriscenje. UI deo poziva endpointe Engine-a radi obrade raznih zahteva i podataka. Pri tome samo Engine komunicira sa bazom, a UI sa Engine-om.

Baza podataka (DB)

Baza podataka je u komunikaciji sa Engine-om za svrhu skladistenja podataka o aplikaciji. U bazi se skladiste svi esencijalno bitni podaci za rad aplikacije.

Model baze kao i tip baze (NoSQL, SQL) je proizvoljan.

Nacina ocenjivanja

1. Aplikacija je funkcionalna i postoji Flask aplikacija – 51 poen
 - a. Aplikacija se sastoji od 1 aplikacije bez baze koja je potpuno funkcionalna
 2. Implementiran Engine kao posebna Flask aplikacija gde UI komunicira sa Engine-om putem API-a – 10 poena
 3. Implementirana je baza sa kojom komunicira Engine – 9 poena
 4. Koriscenje niti prilikom implementacije – 10 poena
 5. Koriscenje procesa prilikom implementacije – 10 poena
 6. Dockerizacija aplikacije i pokretanje na vise racunara (distribuiran sistem) – 10 poena
- Deploy aplikacije na Heroku – gratis 5 poena (moguće samo ako je svih 6 tacaka implementirano)