

DISTRIBUIRANI RAČUNARSKI SISTEMI

ONLAJN KUPOVINA

- PR79/2020 Jovan Vukosavljević
- PR82/2020 Nikola Lazarević
- PR74/2020 Vanja Mihajlović
- PR75/2020 Dajana Radović

Opis projektnog zadatka

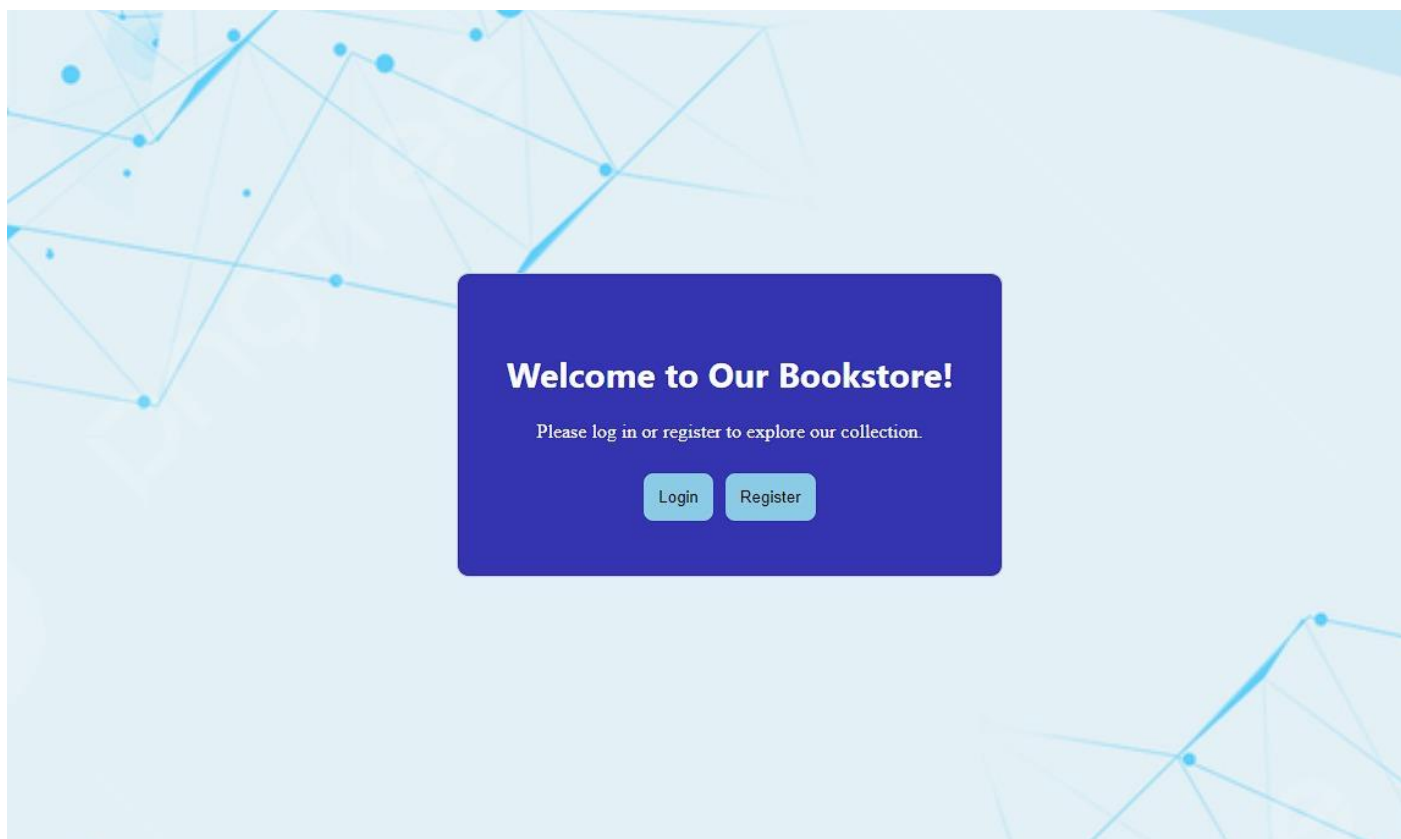
Projektni zadatak predstavlja onlajn aplikaciju koja služi za kupovinu proizvoda. Imamo mogućnost logovanja kao regularan korisnik i ako administrator.

Inicijalno se prikazuje Login ekran koji nakon ulogovanja redirektuje na odgovarajući prozor u zavisnosti da li je korisnik kupac ili administrator.

Signup prozor omogućava kreiranje novog naloga (email i password moraju biti jedinstveni za svaki nalog).

Administrator ima mogućnost kreiranja novog proizvoda, povećanja količine istog i verifikacije korisničkog naloga. Takođe može da prati uživo aktivnost korisnika – ko je šta kupio.

Korisnik vidi proizvode na stanju i ima mogućnost kupovine. Verifikuje svoj nalog tako što unosi broj svoje kartice (nema mogućnost kupovine ukoliko nije verifikovan), ima mogućnost da menja polja svog profila. Dostupan mu je pregled svog profila i ima mogućnost pregleda stanja u različitim valutama.



Slika 1 – početni ekran

User General

[Store](#)[Profile](#)[Verification](#)[Payments](#)[Amount](#)[Conversion](#)[History](#)[Logout](#)

Profile

Name:

Dajana

Surname:

Radovic

Address:

Adresa

City:

Grad

Country:

Zemlja

Telephone Number:

423432432

Email:

nikola@gmail.com

Password:

....

Save

Slika 2 – korisnikov pregled

The image shows a web application interface for an admin panel. At the top, there's a dark blue header with the text "Admin General" in white. Below the header, there's a row of five light blue buttons: "Add Product", "Product Info", "Tracking", "Verification", and "Logout". The main content area is a light blue box with the title "Add New Product" in bold. Inside this box, there are four input fields: "Name:" (a text input), "Price:" (a text input with "0" inside), "Currency:" (a dropdown menu with a downward arrow), and "Quantity:" (a text input with "0" inside). At the bottom of the form, there's a rounded rectangular button labeled "Add Product". The background of the entire page is a light blue gradient with a network-like pattern of dots and lines.

Slika 3 – adminov pregled

Opis polja:

Korisnik

- **Store** – pregled svih proizvoda na stanju koje korisnik može da kupi.
- **Profile** – pregled polja profila i mogućnost izmene.
- **Verification** – korisnik unosi broj kartice
- **Payments** – Korisnikov pregled računa u svim valutama
- **Amount** – Dodavanje novca na račun u proizvoljnoj valuti
- **Conversion** – Konverzija stanja računa iz jedne valute u drugu
- **Hisory** – Pregled kupljenih proizvoda
- **Logout**

Administrator:

- **Add product** — Dodavanje novog proizvoda
- **Product info** — informacije o proizvodima na stanju. Mogućnost povećanja količine proizvoda na stanju
- **Tracking** — uživo praćenje kupljenih proizvoda
- **Logout**

Korišćene tehnologije

Front-end:

Korišćen je Angular za izradu front-enda. Za svaki prozor i opciju je napravljena posebna komponenta sa svojim html kodom, css opisom izgleda, typescript fajlom za definisanje i poziv funkcija. Za menjanje prozora se koristi rutiranje.

Back-end:

Rađen u Python-u., Flask aplikacija. Projekat radi po principu REST API-ja. Server je stateless što znači da ne čuva stanje korisnika i zbog toga koristi tokene pri svakom korisničkom zahtevu kako bi mogao da identifikuje korisnika i proveri njegova prava pristupa. Back-end omogućuje povezivanje korisnika sa bazom podataka i koristi HTTP metode kako bi dostavio tražene informacije iz te baze ili ažurirao istu.

Obezbeđen je rad sa multiprocesiranjem koji poboljšava performance aplikacije.

Baza podataka:

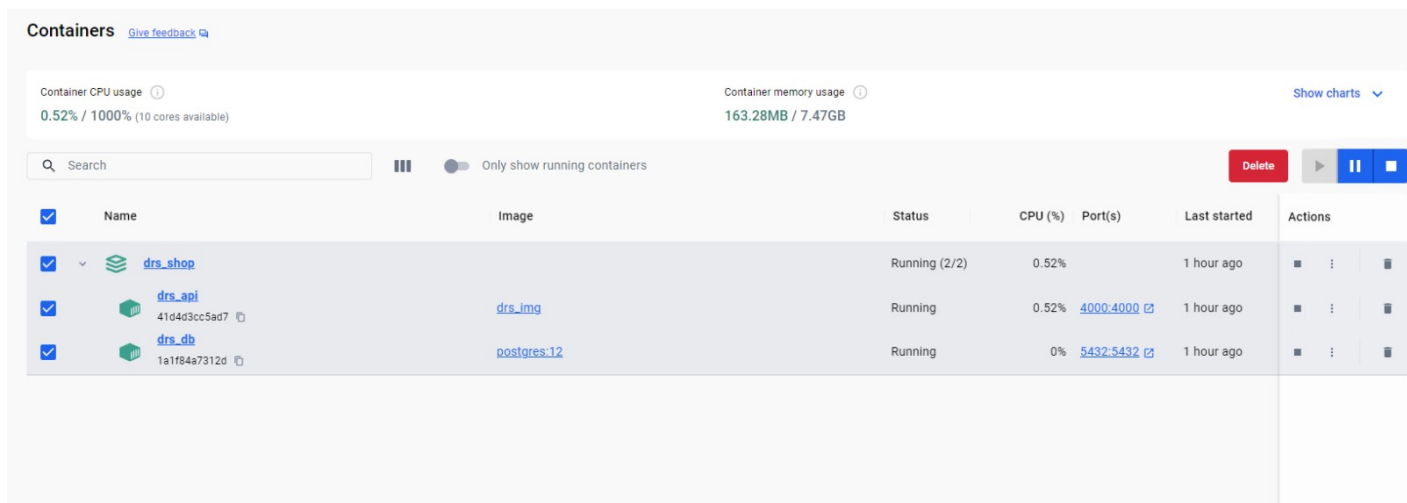
PostgreSQL

Docker:

Kako bismo mogli koristiti aplikaciju na različitim računarima koristili smo Docker. On uz pomoć Docker slika kreira kontejnere koji sadrže sve neophodne instalacije i sam kod aplikacije i obezbeđuje nesmetano izvršavanje programa.

Drs_api i drs_dp predstavljaju servise koji su zaduženi za izvršavanje baze podataka i API-ja. Drs_api zavisi od drs_db i zato se drs_db mora pokrenuti prvo.

Koristeći Dockerfile definišemo korake za kreiranje Docker kontejnera.



Slika 4- Pokretanje Docker kontejnera

Okruženja:

Visual Studio Code – Front-end

PyCharm – Front-end, Back-end