**Dokumentacija aplikacije Psarna Albus**

**Predmet: Zanesljivost in testiranje IS**

**Avtor: Anže Kozole**

To je spletna aplikacija katera je narejena iz mikrostoritev. Da ni večjih komplikacij sem odstranil integracijo z Docker-jem ker sem imel tam določene težave.

Do repositorya se lahko dostopa tukaj: <https://github.com/anzeko123/ZTIS-PsarnaAlbus>

Za vse endpointe in preizkus endpointov so zraven v repository tudi pripravljeni postman klici.

**Struktura aplikacije:**

Narejene so 3 mikrostoritve na principu REST:

1. Registracija uporabnikov ki zajema: prikaz vseh uporabnikov, registracija, prijava, odjava, izbris uporabnika, rezervacija psa

* Narejeno je v Node.js z uporabo MongoDB baze katero sem jaz vzpostavil

1. Rezerviranje psov ki zajema: rezervacija psa, izbris psa, prikaz vseh psov

* Narejeno je v SpringBoot z uporabo MySQL podatkovne baze za shranjevanje pdoatkov
* Podatkovna baza ima ime: *p-albus-rezervacija*

1. Podatkovna baza ima ime Rezerviranje termina za ogled ki zajema: izpis vseh rezerviranih terminov, rezervacija termina, izbris termina

* Narejeno je v Quarkus z uporabo MySQL podatkovne baze za shranjevanje podatkov
* Podatkovna baza ima ime: *p-albus-reservation-date*

Frontend je narejen v React.js

Obstaja tudi prehodna storitev kjer so združene vse mikrostoritve tako da frontend kliče samo eno storitev. Ta prehodna storitev pa kliče vse ostale mikrostoritve.

**Podatkovne baze**

Za zagon je potrebno vzpostaviti tudi lokalne MySQL baze. Narediti je potrebno dve bazi katere vsebujejo eno tabelo. Vse podatke table sem izvozil in so na voljo v repositoryu

1. ***p-albus-rezervacija***– tabela:***pes***
2. ***p-albus-reservation-date***– tabela:***reservationdate***

**Vzpostavitev projekta**

Celotni projekt se lahko odpre v Visual Studio Code. Za vsako mikrostoritev je potrebno imeti ločen terminal.

Pri vsaki Node.js storitvi je potrebno ob prvem zagonu izvesti ukaz ***npm install.*** S tem se bodo namestili vsi paketi kateri so uporabljeni za delovanje storitev.

Prvo je potrebno zagnati vse mikrostoritve.

1. Register - Node.js

* Dostop do mape v terminalu: ***cd “microservices\Register”***
* Zagon: ***npm start***

1. Reservation – SpringBoot

* Dostop do mape v terminalu: ***cd “microservices\Reservation”***
* Zagon: ***./mvnw spring-boot:run***

1. ReservationDate – Quarkus

* Dostop do mape v terminalu: ***cd “microservices\ReservationDate”***
* Zagon: ***./mvnw quarkus:dev***

Ko se te zaženejo lahko zaženemo prehodno storitev:

1. backend\_for\_frontend – Node.js

* Dostop do mape v terminalu: ***cd “backend\_for\_frontend”***
* Zagon: ***npm start***

Ko je vse zagnano se lahko zažene frontend

1. Frontend

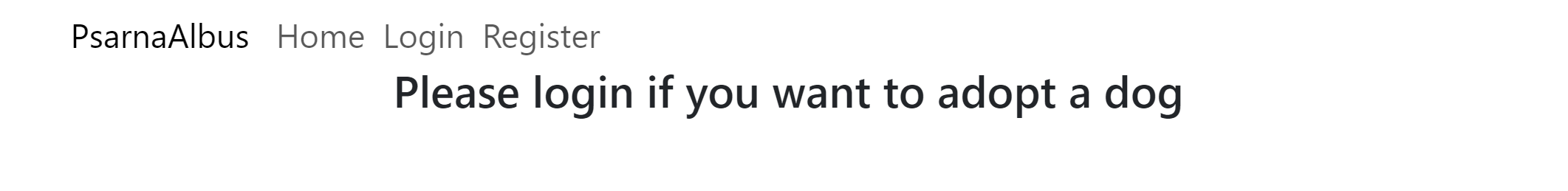
* Dostop do mape v terminalu: ***cd “frontend\register\_reserve\_dog”***
* Zagon: ***npm start***

**Funkcionalnosti**

1. Registracija – uporabnik se lahko registrira kjer je zahtevano vpisati uporabniško ime, epošto, geslo in ponovno geslo.
2. Prijava – uporabnik se lahko prijavi v aplikacijo, kjer je zahtevano uporabniško ime in geslo. Ob prijavi se uporabniški podatki shranijo v lokalno shrambo kateri so potem uporabljeni za rezervacijo psa.
3. Odjava – uporabnik se lahko ob prenehanju aplikacije odjavi
4. Izbris računa – uporabnik lahko tudi popolnoma izbriše račun
5. Ogled psov – Na začetni strani si lahko uporabnik ogleda vse pse kateri so na voljo. Nekateri so že rezervirani za oddajo kateri so tudi označeni.
6. Rezervacija psa – Uporabnik lahko izbere katerega koli psa bo rezerviral in ta se bo zapisal v uporabnikove podatke in pod njegovo id številko
7. Uporabnik si lahko tudi rezervira termin za ogled Psarne Albus. Ta vpiše vse podatke v obrazec in vnesen termin bob il zabeležen pod uporabnika
8. Uporabnik lahko ob želji da spremeni svoj datum vnese novi datum v obrazec za rezervacijo termina in zapisal se bo novi datum (zaradi napake v aplikacije se prejšnji datum ne bo izbrisal).
9. Če je termin že zaseden uporabniku sporoči da je termin že zaseden

**Frontend**

1. **Začetna stran brez prijave**



1. **Začetna stran ob prijavi**

A screenshot of a computer

Description automatically generated

1. **Prijavna stran**

A screenshot of a login screen

Description automatically generated

1. **Registracija**

A screenshot of a login form

Description automatically generated

1. **Rezerviranje obiska**

A close-up of a login form

Description automatically generated