SVEUČILIŠTE U ZAGREBU

**FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA**

ZAVRŠNI RAD br. 1497

**WEB-APLIKACIJA ZA PLANIRANJE PUTOVANJA**

Jan Grbac

Zagreb, lipanj 2024.

SVEUČILIŠTE U ZAGREBU

**FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA**

ZAVRŠNI RAD br. 1497

**WEB-APLIKACIJA ZA PLANIRANJE PUTOVANJA**

Jan Grbac

Zagreb, lipanj 2024.

U okviru ovog završnog rada potrebno je razviti aplikaciju kojom korisnici mogu stvarati nove planove putovanja, spremiti postojeće planove pod svoj profil te ih podijeliti sa zajednicom. Pri planiranju puta, korisnici mogu koristiti interaktivnu kartu gdje odabiru ključna mjesta u svom putu. Web-aplikacija može korisnicima predložiti atrakcije za posjetiti u određenom mjestu, te se korisnici mogu informirati o prijevozu, hotelima, vremenskoj prognozi itd. Aplikacija bi trebala korisnicima omogućiti suradnju, odnosno da istovremeno više ljudi uređuje jedan plan. Aplikacija korisnicima daje i okvirnu cijenu određenih usluga koje su odabrali (hoteli, prijevoz). Također, korisnici mogu u svoj itinerar dodavati bilješke.

Sadržaj

[Uvod 1](#_Toc23263729)

[1. Naslov prvog poglavlja 2](#_Toc23263730)

[1.1. Prvo potpoglavlje 2](#_Toc23263731)

[1.2. Stilovi za tekst, naslove i podnaslove 2](#_Toc23263732)

[1.3. Stilovi za nabrajanje 2](#_Toc23263733)

[1.3.1. Stilovi za nabrajanje s točkama i crticama 3](#_Toc23263734)

[1.4. Slike 4](#_Toc23263735)

[1.5. Tablice 6](#_Toc23263736)

[1.6. Matematički izrazi 7](#_Toc23263737)

[1.7. Programski kôd 7](#_Toc23263738)

[Zaključak 9](#_Toc23263739)

[Literatura 10](#_Toc23263740)

[Sažetak 11](#_Toc23263741)

[Summary 12](#_Toc23263742)

[Skraćenice 13](#_Toc23263743)

[Privitak 14](#_Toc23263744)

# Uvod

# Korištene tehnologije

Pri izradi rada koristio sam nekoliko suvremenih i široko podržanih tehnologija koje omogućavaju visoku razinu funkcionalnosti i olakšavaju razvoj web-aplikacija. Korištene tehnologije općenito se mogu podijeliti u tri skupine: frontend (korisnička strana), backend (poslužitelj) te izvedba baze podataka (također na poslužitelju).

## Frontend

### React

Za korisničku stranu aplikacije kao glavnu tehnologiju odabrao sam React, jednu od najpopularnijih JavaScript knjižnica za izradu korisničkih sučelja. Razvijen od strane Facebooka, React omogućava izradu dinamičkih i responzivnih web stranica koristeći komponente koje se mogu ponovno koristiti.

Komponente su neovisne jedna o drugoj, koristeći prosljeđivanje podataka kroz hijerarhiju kako bi se ostvarila komunikacija. Glavni način međudjelovanja komponenti je korištenjem takozvanih React „hooks“, odnosno udica. Ispostavilo se da su najkorisnije udice upravo useState i useEffect udice.

useState se tipično koristi tako da se definira varijabla na razini komponente te funkcija koja postavlja vrijednost varijable, čija se vrijednost može proslijediti, no bitno, može se proslijediti i funkcija za postavljanje vrijednosti, čime komponente dublje u hijerarhiji mogu mijenjati vrijednost varijable. Nadalje, korištenje useState udica omogućava da se dio web-stranice gdje se „stateful“ varijabla koristi iznova generira pri promjeni vrijednosti te varijable.

useEffect udica se koristi za definiranje „callback“ procedura koje se obavljaju pri promjeni neke (ili više) varijabli sadržanih u „dependency array-u“ (polju ovisnosti). Polje može biti i prazno, pri čemu se procedura izvrši jednom pri učitavanju komponente. Kao varijable se uglavnom koriste upravo „stateful“ varijable definirane u useState udicama.

### TypeScript

TypeScript je korišten kao nadogradnja na JavaScript zbog svojih značajki poput statičkog tipiziranja, što doprinosi većoj pouzdanosti i održivosti koda. TypeScript omogućava otkrivanje potencijalnih grešaka već u fazi razvoja, što je značajno smanjilo broj grešaka pri testiranju aplikacije i uštedilo vremena na debuggiranje.

Statičko tipiziranje ispostavilo se korisno pri definiranju sučelja „Props“ u svakoj React komponenti, gdje se definira koje parametre ta komponenta zahtjeva od roditelja te kojeg su tipa. TypeScript također omogućava definiranje vlastitih tipova, što sam iskoristio za definiranje oblika objekata koje će korisnička strana zaprimiti pri pozivu poslužitelja, time omogućavajući lakšu integraciju s poslužiteljem.

## Backend

### Spring Boot

Za backend dio aplikacije korišten je Spring Boot, koji je dio ekosustava Spring Frameworka. Spring Boot pojednostavljuje razvoj, testiranje i implementaciju robusnih i skalabilnih backend aplikacija u Javi. Ova tehnologija pruža niz značajki koje ubrzavaju razvoj aplikacije, kao što su automatska konfiguracija i ugrađeni poslužitelji kao što su Tomcat i Jetty.

Spring je bio prirodan odabir zbog svoje široke podržanosti, robusne dokumentacija i izvedbe, lakoće korištenja, performansi te mnogih drugih poželjnih značajki.

Osim samog Spring Boota, jedan od glavnih dijelova koji sam koristio je upravo Spring Security framework, koji omogućava relativno lagano postavljanje sigurnosne infrastrukture za web-aplikaciju, kao što su filteri te uloge. Kao komplementarnu tehnologiju Spring Security-ju uveo sam JSON web token (JWT), koji se koristi u kolačićima za autentifikaciju i autorizaciju korisnika.

### SocketIO

Osim samog Spring frameworka, koristio sam i implementaciju WebSocketa u Javi, odnosno SocketIO. SocketIO se pokreće kao zasebna komponenta backenda na različitom portu od glavnog servera, te služu za facilitaciju komunikacije među korisnicima koji uređuju isto putovanje. U tu svrhu koristi se podjela na takozvane sobe (rooms) gdje korisnici koji uređuju isto putovanje uđu u istu sobu te šalju poruke na SocketIO server, koji prosljeđuje te poruke svim ostalim korisnicima, time ostvarujući komunikaciju i kolaboraciju u stvarnom vremenu.

## Baza podataka

### PostgreSQL

PostgreSQL je odabran kao sustav za upravljanje bazom podataka zbog svoje stabilnosti, performansi i podrške za napredne SQL značajke. PostgreSQL je objektno-relacijska baza podataka otvorenog koda koja omogućava efikasno upravljanje velikim količinama podataka te pruža robusne sigurnosne značajke i podršku za transakcije. Time se može smatrati nadogradnjom na jednostavni SQL. Baza je tijekom razvoja bila spremana lokalno te sam za pregled baze koristio alat pgAdmin 4.

# Zahtjevi sustava

## Generički zahtjevi aplikacije

* Mehanizam autentifikacije i autorizacije, temeljen na korisničkom imenu i lozinci
* Spremanje i izmjena korisničkih podataka
* Intuitivno i easy-to-use korisničko sučelje
* Odazivnost i responzivnost

## Specifični zahtjevi aplikacije

* Stvaranje i uređivanje planova putovanja
* Interaktivna karta za vizualiziranje planova
* Preporučivanje atrakcija i smještaja u okolini točaka puta
* Dijeljenje planova putovanja s zajednicom
* Real-time suradnja korisnika na jednom planu putovanja
* Vremenska prognoza za planove unutar nekog raspona datuma
* Troškovnik puta
* Bilješke za svaki dio puta

## Korisnički scenariji

# Arhitektura sustava

## Model arhitekture sustava

Web-aplikacija se služi arhitekturom sustava klijent-poslužitelj, u kojoj korisnikov web-preglednik šalje zahtjeve na server te prikazuje odgovarajuću web-stranicu. Ovaj model omogućuje skalabilnost, modularnost i fleksibilnost sustava. Klijentska strana je odgovorna za prezentaciju i interakciju s korisnikom, dok poslužiteljska strana rukuje logikom, pohranom podataka i integracijom s vanjskim servisima. Na ovaj način, sustav može lako rasti i prilagoditi se novim zahtjevima i funkcionalnostima.

## Oblik baze podataka

## Životni vijek zahtjeva

HTTP zahtjevi koji se generiraju kod klijenta šalju se na server, gdje ih zaprima neki od RESTful upravitelja (engl. controller) na jednoj od definiranih pristupnih točaka (engl. endpoint). API kojim se aplikacija koristi je interan i time poznat samo razvijatelju aplikacije. Zahtjevi uglavnom sadržavaju i JSON web token kojim korisnik dokazuje da ima autorizaciju da zahtjeva taj sadržaj, no postoje i pristupne točke koje su dostupne svima, neovisno o web tokenu. Nakon što upravitelj kroz servise i repozitorije izvrši svoju logiku, klijentu se vraća odgovor s HTTP status kodom i potencijalno s JSON objektom ako je klijent zatražio nekakve podatke.

# Implementacija sustava

## Struktura aplikacije

## Implementacija korisničke strane

## Implementacija serverske strane

## Integracija s bazom podataka

## Implementacija sigurnosnih značajki

# Korištenje sustava

# Zaključak

Na kraju rada piše se kratak zaključak, duljine do najviše jedne stranice.

# Literatura

# Sažetak

# Summary