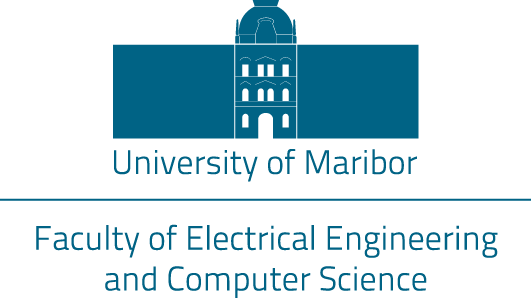
**YourVoice - Projektna dokumentacija**



**Razvoj in upravljanje programskih sistemov**

**Avtorji:** Andraž Oset, Žak Ternik, Siniša Vučetić

Vsebina

[1. Opis projekta 5](#_Toc188297947)

[2. Zaledni del 6](#_Toc188297948)

[2.1. User API 6](#_Toc188297949)

[2.1.1. Pregled 6](#_Toc188297950)

[2.1.2. Namestitev in konfiguracija 6](#_Toc188297951)

[2.1.3. Uporabi API-ja 7](#_Toc188297952)

[2.1.3.1. Pridobitev seznama uporabnikov 7](#_Toc188297953)

[2.1.3.2. Pridobitev enega uporabnika 7](#_Toc188297954)

[2.1.3.3. Registracija uporabnika 8](#_Toc188297955)

[2.1.3.4. Prijava uporabnika 9](#_Toc188297956)

[2.1.3.5. Posodobitev uporabnikovih podatkov 10](#_Toc188297957)

[2.1.3.6. Brisanje uporabnika 11](#_Toc188297958)

[2.2. Post API 11](#_Toc188297959)

[2.2.1. Upravljanje objav 11](#_Toc188297960)

[2.2.1.1. Pridobitev seznama objav 11](#_Toc188297961)

[2.2.1.2. Pridobitev posamezne objave 12](#_Toc188297962)

[2.2.1.3. Pridobitev objav glede na forum 13](#_Toc188297963)

[2.2.1.4. Kreiranje objave 13](#_Toc188297964)

[2.2.1.5. Posodabljanje objave 14](#_Toc188297965)

[2.2.1.6. Brisanje objave 15](#_Toc188297966)

[2.3. Comment API 15](#_Toc188297967)

[2.3.1. Upravljanje komentarjev 15](#_Toc188297968)

[2.3.1.1. Dodaj komentar k objavi 15](#_Toc188297969)

[2.3.1.2. Brisanje komentarja iz objave 16](#_Toc188297970)

[2.3.1.3. Upvote komentarja 16](#_Toc188297971)

[2.3.1.4. Downvote komentarja 16](#_Toc188297972)

[2.4. Forum API 17](#_Toc188297973)

[2.4.1. Upravljanje forumov 17](#_Toc188297974)

[2.4.1.1. Pridobivanje seznama forumov 17](#_Toc188297975)

[2.4.1.2. Pridobivanje foruma glede na id 17](#_Toc188297976)

[2.4.1.3. Pridobivanje možnih značk za forum 17](#_Toc188297977)

[2.4.1.4. Dodajanje foruma 18](#_Toc188297978)

[2.4.1.5. Posodabljanje foruma 18](#_Toc188297979)

[2.4.1.6. Brisanje foruma 19](#_Toc188297980)

[2.5. Poll API 19](#_Toc188297981)

[2.5.1. Upravljanje Polls 19](#_Toc188297982)

[2.5.1.1. Dodaj Poll na objavo 19](#_Toc188297983)

[2.5.1.2. Pridobivanje Polls za objavo 19](#_Toc188297984)

[2.5.1.3. Oddaj glas za Poll 20](#_Toc188297985)

[3. Čelni del 20](#_Toc188297986)

[3.1. Tehnične Zahteve 21](#_Toc188297987)

[3.2. Struktura Datotek 21](#_Toc188297988)

[3.2.1. App.tsx 21](#_Toc188297989)

[3.2.2. Routes (routes.tsx) 22](#_Toc188297990)

[3.2.3. UserContext (userContext.ts) 22](#_Toc188297991)

[3.3. Glavne Komponente 22](#_Toc188297992)

[3.3.1. Home.tsx 22](#_Toc188297993)

[3.3.2. Posts.tsx 23](#_Toc188297994)

[3.3.3. PostDetail.tsx 23](#_Toc188297995)

[3.3.4. AddPostModal.tsx 24](#_Toc188297996)

[3.3.5. Register.tsx in Login.tsx 24](#_Toc188297997)

[3.3.6. Profile.tsx 24](#_Toc188297998)

[3.3.7. Header.tsx 25](#_Toc188297999)

[3.3.8. Footer.tsx 25](#_Toc188298000)

[3.4. Ostalo 25](#_Toc188298001)

[3.4.1. ProtectedRoute.tsx 25](#_Toc188298002)

[3.4.2. Upravljanje Uporabniškega Stanja (UserContext.ts) 26](#_Toc188298003)

[3.5. Uporabniška Izkušnja in varnost 26](#_Toc188298004)

# Opis projekta

YourVoice je interaktivni forum, zasnovan za izmenjavo informacij in povezovanje uporabnikov. Aplikacija omogoča prijavljenim uporabnikom, da objavljajo svoje vsebine, komentirajo, ocenjujejo objave drugih ter urejajo svoj profil. Prijavljeni uporabniki lahko tudi aktivno sodelujejo pri predlogih in ocenjevanju izboljšav spletnega foruma. Neprijavljeni uporabniki lahko brskajo po objavah, vendar brez možnosti interakcije. Napredne funkcionalnosti vključujejo filtriranje in sortiranje objav, napredno iskanje ter ostale možnosti. Aplikacija vključuje tudi posebne funkcionalnosti ter pravice za moderatorje in administratorje. Za razvoj je uporabljen MERN sklad.

# Zaledni del

API privzeto deluje na **vratih 3000**, vendar se lahko v nastavitvah po potrebi poljubno konfigurira.

Uporabljene knjižnice:

* express: za upravljanje strežnika in definiranje poti
* mongoose: za komunikacijo z MongoDB
* bcrypt: za hashiranje gesel
* express-session: za upravljanje sej uporabnikov

## User API

### Pregled

User API omogoča upravljanje uporabnikov v aplikaciji. API vključuje funkcionalnosti za registracijo, prijavo, ogled profila, posodabljanje uporabniških podatkov ter brisanje uporabnikov. Vključuje tudi zaščito z uporabo hashiranja gesel in preverjanje moči gesel.

### Namestitev in konfiguracija

1. Kloniramo projekt iz git-a:

|  |
| --- |
| <https://github.com/zakternik/YourVoice.git>  cd YourVoice  npm install |

1. Nastavitev podatkovne baze: V datoteki dbdata.js zagotovite pravilne podatke za povezavo z MongoDB.
2. Zagon strežnika:

|  |
| --- |
| npm start |

### Uporabi API-ja

#### Pridobitev seznama uporabnikov

**Pot: GET /user**

Pridobi seznam vseh uporabnikov, pri čemer so gesla izključena iz pridobljenih atributov.

**Primer odziva:**

|  |
| --- |
| [  {  "\_id": "userId",  "username": "user123",  "email": "user@example.com",  "role": "user",  "bio": "",  "avatar": "",  "isBanned": false,  "createdAt": "2024-01-01T12:00:00Z"  }  ] |

#### Pridobitev enega uporabnika

**Pot: GET /user/:id**

Pridobi podatke o enem uporabniku po ID-ju.

**Primer odziva ob uspehu:**

|  |
| --- |
| {  "\_id": "userId",  "username": "user123",  "email": "user@example.com",  "role": "user",  "bio": "",  "avatar": "",  "isBanned": false,  "createdAt": "2024-01-01T12:00:00Z"  } |

**Primer odziva pri napaki:**

|  |
| --- |
| {  "message": "No such User"  } |

#### Registracija uporabnika

**Pot: POST /user**

Ustvari novega uporabnika z vnesenimi parametri.

**Parametri:**

* **username (obvezno, unique):** uporabniško ime
* **email (obvezno, unique):** veljaven email naslov
* **password (obvezno):** mora biti dolgo vsaj 8 znakov, z vsaj eno veliko črko in eno številko

**Primer odziva ob uspehu:**

|  |
| --- |
| {  "\_id": "userId",  "username": "user123",  "email": "user@example.com",  "role": "user"  } |

**Primer odziva ob napaki:**

|  |
| --- |
| {  "message": "Username already taken. Please choose another.",  "error": {  "code": 11000,  "keyPattern": {  "username": 1  }  }  } |

#### Prijava uporabnika

**Pot: POST /user/login**

Preveri uporabniške podatke in ustvari sejo, če so pravilni.

**Parametri:**

* **username (obvezno):**
* **password (obvezno):**

**Primer odziva ob uspehu:**

|  |
| --- |
| {  "\_id": "userId",  "username": "user123",  "email": "user@example.com",  "role": "user"  } |

#### Posodobitev uporabnikovih podatkov

**Pot: PUT /user/:id**

Posodobi podatke o uporabniku, vključno z geslom, biografijo in avatarjem.

**Parametri:**

* **password (opcijsko):** novo geslo (mora biti validirano)
* **bio (opcijsko):** opis uporabnika
* **avatar (opcijsko):** URL slike za avatar

**Primer odziva ob uspehu:**

|  |
| --- |
| {  "message": "User updated successfully"  } |

**Primer odziva ob napaki:**

|  |
| --- |
| {  "message": "Geslo mora biti dolgo vsaj 8 znakov in vsebovati vsaj eno veliko črko in eno številko."  } |

#### Brisanje uporabnika

**Pot: DELETE /user/:id**

Izbriše uporabnika iz podatkovne baze.

**Parametri:**

* **id (obvezno):** ID uporabnika

**Primer odziva ob uspehu:**

|  |
| --- |
| **// Ni vsebine, HTTP status koda: 204 No Content** |

**Primer odziva ob napaki:**

|  |
| --- |
| {  "message": "No such User"  } |

## Post API

API omogoča upravljanje objav in komentarjev v vaši aplikaciji, z vključitvijo osnovnih operacij (ustvarjanje, posodabljanje, brisanje) za objave in komentarje.

### Upravljanje objav

#### Pridobitev seznama objav

**Pot: GET /post**

Pridobi seznam vseh objav.

**Primer odziva:**

|  |
| --- |
| [  {  "\_id": "postId",  "upvotes": 0,  "downvotes": 0,  "title": "Primer objave",  "content": "Vsebina objave",  "tags": [],  "comments":[],  "userId": "userId",  "forumId": " forumId ",  " updatedAt": "2024-01-01T12:00:00Z"  "createdAt": "2024-01-01T12:00:00Z"  }  ] |

#### Pridobitev posamezne objave

**Pot: GET /post/:id**

Pridobi podrobnosti določene objave na podlagi njenega ID-ja, vključno s pripadajočimi komentarji.

**Primer odziva:**

|  |
| --- |
| [  {  "\_id": "postId",  "upvotes": 0,  "downvotes": 0,  "title": "Primer objave",  "content": "Vsebina objave",  "tags": [],  "comments":[],  "userId": "userId",  "forumId": " forumId ",  " updatedAt": "2024-01-01T12:00:00Z"  "createdAt": "2024-01-01T12:00:00Z"  }  ] |

**Primer odziva pri napaki:**

|  |
| --- |
| {  "message": "No such Post"  } |

#### Pridobitev objav glede na forum

**Pot: GET /post/byForum/:forumId**

Vrne objave iz izbranega foruma.

**Primer odziva:**

|  |
| --- |
| [  {  "\_id": "postId",  "upvotes": 0,  "downvotes": 0,  "title": "Primer objave",  "content": "Vsebina objave",  "tags": [],  "comments":[],  "userId": "userId",  "forumId": " forumId ",  " updatedAt": "2024-01-01T12:00:00Z"  "createdAt": "2024-01-01T12:00:00Z"  }  ] |

#### Kreiranje objave

**Pot: POST /post**

Ustvari novo objavo s polji: naslov (title), vsebina (content), kategorija (category) in uporabnikov ID (userId).

**Primer odziva:**

|  |
| --- |
| [  {  "title": "Primer objave",  "content": "Vsebina objave",  "tags": [],  "userId": "userId",  "forumId": " forumId ",  }  ] |

**Primer odziva pri napaki:**

|  |
| --- |
| {  "message": "Error when creating Post."  } |

|  |
| --- |
| {  "message": "Post flagged inappropriate."  } |

#### Posodabljanje objave

**Pot: PUT /post/:id**

Posodobi objavo s polji: naslov (title), vsebina (content), kategorija (category) in uporabnikov ID (userId).

**Primer odziva:**

|  |
| --- |
| [  {  "title": "Primer objave",  "content": "Vsebina objave",  "tags": [],  "userId": "userId",  }  ] |

**Primer odziva pri napaki:**

|  |
| --- |
| {  "message": "Error when updating Post."  } |

#### Brisanje objave

**Pot: DELETE /post/:id**

Izbriše določeno objavo na podlagi njenega ID-ja.

**Uspešen odziv:** Brez vsebine, HTTP statusna koda 204.

**Primer odziva pri napaki:**

|  |
| --- |
| {  "message": "Error when deleting Post."  } |

## Comment API

### Upravljanje komentarjev

#### Dodaj komentar k objavi

**Pot: POST /comment**

Doda nov komentar k določeni objavi na podlagi njenega ID-ja. Obvezni polji sta userId, postId in vsebina (content). ParentId označuje prejšnji komentar v primeru odgovora na komentar.

**Primer zahtevka:**

|  |
| --- |
| [  {  "content": "Vsebina",  "userId": "userId",  "postId": " postId",  "parentId": " parentId ",  }  ] |

**Primer odziva ob uspehu:**

|  |
| --- |
| **// Ni vsebine, HTTP status koda: 204 No Content** |

#### Brisanje komentarja iz objave

**Pot: DELETE /comment/:id**

Opis: Zamenja vsebino komentarja z [DELETED] in smeri ustvarjalca komentarja na Ghost uporabnika. Tako se ohranijo odgovori na izbrisan komentar.

**Uspešen odziv:** Brez vsebine, HTTP statusna koda 204. Odziv ob napaki:

**Primer odziva pri napaki:**

|  |
| --- |
| {  "message": "No such comment."  } |

#### Upvote komentarja

**Pot: PUT /comment/upvote/:id**

Opis: Zabeleži upvote za trenutnega uporabnika na danem komentarju. Če je uporabnik že dal downvote, bo ga spremenilo v upvote.

**Uspešen odziv:** Brez vsebine, HTTP statusna koda 204.

#### Downvote komentarja

**Pot: PUT /comment/downvote/:id**

Opis: Zabeleži downvote za trenutnega uporabnika na danem komentarju. Če je uporabnik že dal upvote, bo ga spremenilo v downvote.

**Uspešen odziv:** Brez vsebine, HTTP statusna koda 204.

## Forum API

### Upravljanje forumov

#### Pridobivanje seznama forumov

**Pot: GET /forum**

Vrne listo forumov.

**Primer odziva:**

|  |
| --- |
| [  {  "title": "Naslov",  "adminId": "adminId",  "createdAt": "createdAt",  }  ] |

#### Pridobivanje foruma glede na id

**Pot: GET /forum/:id**

**Primer odziva:**

|  |
| --- |
| {  "title": "Naslov",  "adminId": "adminId",  "createdAt": "createdAt",  } |

#### Pridobivanje možnih značk za forum

**Pot: GET /forum/tags/:id**

**Primer odziva:**

|  |
| --- |
| [  {  "name": "Ime",  }  ] |

#### Dodajanje foruma

**Pot: POST /forum**

Kreira nov forum z podanim naslovom in značkami.

**Primer zahtevka:**

|  |
| --- |
| {  "title": "Naslov",  "admin": "adminId",  "tags": []  } |

**Primer odziva:**

|  |
| --- |
| **// Ni vsebine, HTTP status koda: 204 No Content** |

#### Posodabljanje foruma

**Pot: PUT /forum/:id**

Posodobi naslov ali pa značke.

**Primer zahtevka:**

|  |
| --- |
| {  "title": "Naslov",  "tags": []  } |

**Primer odziva:**

|  |
| --- |
| **// Ni vsebine, HTTP status koda: 204 No Content** |

#### Brisanje foruma

**Pot: DELETE /forum/:id**

**Primer odziva:**

|  |
| --- |
| **// Ni vsebine, HTTP status koda: 204 No Content** |

## Poll API

### Upravljanje Polls

#### Dodaj Poll na objavo

**Pot: POST /poll**

Ustvari novo anketo na objavi.

**Primer zahtevka:**

|  |
| --- |
| {  "question": "Vprasanje",  "options": [],  "userId": "userId",  "postId": "postId"  } |

**Primer odziva:**

|  |
| --- |
| **// Ni vsebine, HTTP status koda: 204 No Content** |

#### Pridobivanje Polls za objavo

**Pot: GET /poll/list/:postId**

Kreira nov forum z podanim naslovom in značkami.

**Primer zahtevka:**

|  |
| --- |
| {  "question": "Vprasanje",  "options": [],  "userId": "userId",  "postId": "postId",  "votedBy": [],  "createdAt": "createdAt"  } |

**Primer odziva:**

|  |
| --- |
| **// Ni vsebine, HTTP status koda: 204 No Content** |

#### Oddaj glas za Poll

**Pot: PUT /:pollId/vote**

Odda glas za trenutnega uporabnika.

**Primer zahtevka:**

|  |
| --- |
| {  "optionId": "optionId",  "userId": "userId",  } |

**Primer odziva:**

|  |
| --- |
| **// Ni vsebine, HTTP status koda: 204 No Content** |

# Čelni del

Čelni del spletne aplikacije YourVoice je implementiran z uporabo knjižnice **React** in predstavlja spletno platformo za interaktivni forum, kjer uporabniki ustvarjajo in komentirajo objave ter po potrebi upravljajo svoj profil. Aplikacija uporablja knjižnico **Chakra UI** za izgled strani, **Context API** za upravljanje stanja prijave, ter **react-router-dom** za usmerjanje in zaščito poti. Čelni del aplikacije vsebuje komponente za prikaz objav, podroben prikaz posamezne objave, urejanje objav, dodajanje komentarjev in upravljanje uporabniškega profila.

## Tehnične Zahteve

* **Razvojna knjižnica**: React
* **Upravljanje stanja**: Context API
* **Vizualna knjižnica**: Chakra UI
* **Usmerjanje**: react-router-dom
* **Knjižnice in orodja**:
  + react-router-dom za usmerjanje in zaščitene poti
  + Chakra UI za oblikovanje komponent
  + FontAwesome za ikone
  + TypeScript za tipizacijo komponent in podatkov

## Struktura Datotek

### App.tsx

* **Opis:** Glavna komponenta, ki upravlja usmerjanje in inicializacijo.
* **Funkcionalnosti:**
  + **Usmerjanje:** Konfiguracija javnih in zaščitenih poti, ki so dostopne le prijavljenim uporabnikom.
  + **Globalna prijava uporabnika:** Z UserContext je omogočen dostop do podatkov o prijavi vsem komponentam aplikacije.
* **Povezane komponente:**
  + **Header in Footer:** Prikazani na vsaki strani aplikacije.
  + **Home:** Začetna stran.
  + **Posts in PostDetail:** Prikaz objav in podrobnosti posamezne objave.
  + **ProtectedRoute:** Komponenta za zaščito poti, dostopnih le prijavljenim uporabnikom.

### Routes (routes.tsx)

* **Opis:** Konfiguracija poti (strani) aplikacije, ki razdeli poti na javne (publicRoutes) in zaščitene (protectedRoutes).
* **Poti:**
  + **publicRoutes:** vključujejo poti (strani) "Domov", "Prijava", "Registracija" in "Objave".
  + **protectedRoutes:** vključujejo poti (strani) "Profil" in "Odjava", ki so vidne samo prijavljenim uporabnikom.

### UserContext (userContext.ts)

* **Opis:** Kontekst za globalno upravljanje podatkov prijavljenega uporabnika.
* **Funkcionalnosti:**
  + Hranjenje podatkov o uporabniku (user) in funkcija za posodabljanje (setUserContext), ki omogoča nastavitev ali odstranitev prijave.
  + Omogoča centralizirano prijavno stanje aplikacije.

## Glavne Komponente

### Home.tsx

* **Opis**: Začetna stran aplikacije, kjer so opisane osnovne informacije o spletni aplikaciji.
* **Dostop**: Javno dostopna.
* **Funkcionalnosti**:
  + Prikazuje uvodni opis aplikacije, njene funkcionalnosti in gumb za preusmeritev na stran **Objave**.
* **Uporabljene knjižnice**: Chakra UI za oblikovanje strani, Font Awesome za ikone.

### Posts.tsx

* **Opis**: Stran za prikaz seznama objav.
* **Dostop**: Javno dostopna.
* **Funkcionalnosti**:
  + Prikazuje seznam vseh objav z možnostjo prikaza podrobnosti za vsako objavo.
  + Prijavljeni uporabniki lahko dodajajo nove objave, urejajo in brišejo svoje objave.
  + Modalno okno za dodajanje ali urejanje objave (**AddPostModal**).
* **Uporabljene knjižnice**: Chakra UI, useDisclosure za modalno okno, UserContext za preverjanje prijavnega stanja.

### PostDetail.tsx

* **Opis**: Prikaz podrobnosti posamezne objave, vključno s komentarji.
* **Dostop**: Javno dostopna.
* **Funkcionalnosti**:
  + Prikazuje naslov, vsebino, kategorijo, avtorja in komentarje objave.
  + Uporabniki lahko dodajo komentar (prijava je obvezna) ali izbrišejo svoje komentarje.
  + Modalno okno za dodajanje komentarjev.
* **Uporabljene knjižnice**: Chakra UI za oblikovanje in modalna okna, useParams in useNavigate za pridobivanje parametrov URL in navigacijo.

### AddPostModal.tsx

* **Opis**: Modalno okno za dodajanje ali urejanje objave.
* **Dostop**: Samo prijavljeni uporabniki.
* **Funkcionalnosti**:
  + Če uporabnik ureja svojo objavo, se mu vnosna polja napolnijo z trenutno obstoječimi podatki o objavi.
  + Pošlje POST ali PUT zahtevek na strežnik za ustvarjanje ali posodobitev objave.
* **Uporabljene knjižnice**: Chakra UI za modalno okno, useRef za začetno fokusiranje na polju z naslovom.

### Register.tsx in Login.tsx

* **Opis**: Strani za prijavo in registracijo uporabnikov.
* **Dostop**: Javno dostopni.
* **Funkcionalnosti**:
  + Uporabniki vnesejo svoje podatke (uporabniško ime, geslo, e-poštni naslov za registracijo) ter oddajo zahtevek na strežnik.
  + Prikaz sporočil o uspehu ali napaki z uporabo useToast.
* **Uporabljene knjižnice**: Chakra UI, UserContext za nastavitev stanja ob uspešni prijavi ali registraciji.

### Profile.tsx

* **Opis**: Stran za ogled profilnih podatkov uporabnika.
* **Dostop**: Samo prijavljeni uporabniki
* **Funkcionalnosti**:
  + Prikazuje podatke o uporabniku (uporabniško ime, e-poštni naslov, datum registracije).
  + Če uporabnik ni prijavljen, je preusmerjen na stran za prijavo.
* **Uporabljene knjižnice**: Chakra UI za oblikovanje, UserContext za dostop do podatkov uporabnika.

### Header.tsx

* **Opis**: Glava aplikacije, kjer se prikazuje navigacijski meni.
* **Funkcionalnosti**:
  + Prikazuje poti glede na prijavno stanje uporabnika: zaščitene poti za prijavljene uporabnike in javne poti za obiskovalce.
* **Uporabljene knjižnice**: react-router-dom za navigacijo, UserContext za preverjanje stanja prijave.

### Footer.tsx

* **Opis**: Spletno dno aplikacije.
* **Funkcionalnosti**:
  + Prikazuje informacije o aplikaciji in članih ekipe ter povezave do družbenih omrežij.
* **Uporabljene knjižnice**: Ikone za socialna omrežja z uporabo SVG ikon.

## Ostalo

### ProtectedRoute.tsx

* **Opis**: Komponenta za zaščito poti, dostopnih le prijavljenim uporabnikom.
* **Funkcionalnosti**:
  + Če je uporabnik prijavljen (user je nastavljen), prikaže ustrezno komponento.
  + Če uporabnik ni prijavljen, ga preusmeri na stran za prijavo.

### Upravljanje Uporabniškega Stanja (UserContext.ts)

* **Opis**: UserContext zagotavlja globalno shranjevanje podatkov o prijavi in omogoča uporabo teh podatkov v celotni aplikaciji.
* **Funkcionalnosti**:
  + UserContext hrani stanje uporabnika (user) in funkcijo setUserContext za posodabljanje stanja prijave.
  + Uporablja se v vseh zaščitenih komponentah in pri določanju vidnosti poti v aplikaciji.

## Uporabniška Izkušnja in varnost

* **Zaščitene poti**:

ProtectedRoute komponenta zagotavlja, da je dostop do zaščitenih poti (strani) omogočen le prijavljenim uporabnikom.

* **Prijaznost uporabniškega vmesnika**:
  + Chakra UI za lepo in enostavno oblikovanje.
  + Modalna okna za dodajanje ali urejanje vsebin, kot so objave in komentarji, kar zagotavlja uporabniku nemoteno uporabniško izkušnjo, brez potrebe po osveževanju strani.
* **Upravljanje napak**:

Komponente, kot je na primer **PostDetail**, vsebujejo ustrezno ravnanje z napakami pri pridobivanju, dodajanju in brisanju komentarjev. Ob pojavi napake se uporabniku prikaže sporočilo, kar izboljša preglednost in zagotavlja boljšo uporabniško izkušnjo.