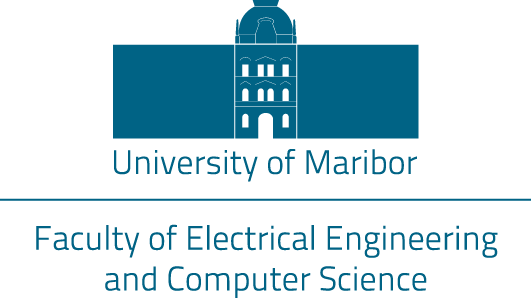
**YourVoice - Projektna dokumentacija**



**Razvoj in upravljanje programskih sistemov  
v2**

**Avtorji:**Matic Lukežič, Sabin Paurč, Leon Tikvič

Vsebina

[1. Opis projekta 4](#_Toc188291658)

[2. Zaledni del 5](#_Toc188291659)

[2.1. User API 5](#_Toc188291660)

[2.1.1. Pregled 5](#_Toc188291661)

[2.1.2. Namestitev in konfiguracija 5](#_Toc188291662)

[2.1.3. Uporabi API-ja 6](#_Toc188291663)

[2.1.3.1. Pridobitev seznama uporabnikov 6](#_Toc188291664)

[2.1.3.2. Pridobitev enega uporabnika 6](#_Toc188291665)

[2.1.3.3. Registracija uporabnika 7](#_Toc188291666)

[2.1.3.4. Prijava uporabnika 8](#_Toc188291667)

[2.1.3.5. Posodobitev uporabnikovih podatkov 9](#_Toc188291668)

[2.1.3.6. Brisanje uporabnika 10](#_Toc188291669)

[2.2. Post API 10](#_Toc188291670)

[2.2.1. Upravljanje objav 10](#_Toc188291671)

[2.2.1.1. Pridobitev seznama objav 10](#_Toc188291672)

[2.2.1.2. Pridobitev posamezne objave 11](#_Toc188291673)

[2.2.1.3. Kreiranje objave 12](#_Toc188291674)

[2.2.1.4. Posodabljanje objave 13](#_Toc188291675)

[2.2.1.5. Brisanje objave 13](#_Toc188291676)

[2.2.1.6. Označevanje objave kot arhivirano 14](#_Toc188291677)

[2.2.1.7. Prekinitev arhiviranja objave 14](#_Toc188291678)

[2.2.2. Upravljanje komentarjev 15](#_Toc188291679)

[2.2.2.1. Dodaj komentar k objavi 15](#_Toc188291680)

[2.2.2.2. Posodabljanje komentarja/ 16](#_Toc188291681)

[2.2.2.3. Brisanje komentarja 16](#_Toc188291682)

[2.2.2.4. Odzivanje z čustvenčeki 17](#_Toc188291683)

[2.2.2.5. Odstranitev odziva z čustvenčki 17](#_Toc188291684)

[2.3. Category API 18](#_Toc188291685)

[2.3.1. Iskanje objav po kategoriji 18](#_Toc188291686)

[3. Čelni del 19](#_Toc188291687)

[3.1. Tehnične Zahteve 19](#_Toc188291688)

[3.2. Struktura Datotek 20](#_Toc188291689)

[3.2.1. App.tsx 20](#_Toc188291690)

[3.2.2. Routes (routes.tsx) 20](#_Toc188291691)

[3.2.3. UserContext (userContext.ts) 21](#_Toc188291692)

[3.3. Glavne Komponente 21](#_Toc188291693)

[3.3.1. Home.tsx 21](#_Toc188291694)

[3.3.2. Posts.tsx 21](#_Toc188291695)

[3.3.3. PostDetail.tsx 22](#_Toc188291696)

[3.3.4. AddPostModal.tsx 22](#_Toc188291697)

[3.3.5. Register.tsx in Login.tsx 23](#_Toc188291698)

[3.3.6. Profile.tsx 23](#_Toc188291699)

[3.3.7. Header.tsx 23](#_Toc188291700)

[3.3.8. Footer.tsx 24](#_Toc188291701)

[3.4. Ostalo 24](#_Toc188291702)

[3.4.1. ProtectedRoute.tsx 24](#_Toc188291703)

[3.4.2. Upravljanje Uporabniškega Stanja (UserContext.ts) 24](#_Toc188291704)

[3.5. Uporabniška Izkušnja in varnost 25](#_Toc188291705)

# 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": "",  "isBanned": false,  "createdAt": "2024-01-01T12:00:00Z",  "avatar": "/avatars/bear.png,  }  ] |

#### 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": "",  "isBanned": false,  "createdAt": "2024-01-01T12:00:00Z"  "avatar": "/avatars/bear.png  } |

**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:**

* **username(opcijsko):** uporabniško ime uporabnika
* **email(opcijsko):** email uporabnika
* **bio (opcijsko):** opis uporabnika
* **avatar (opcijsko):** URL slike za avatar

**Primer odziva ob uspehu:**

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

#### 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",  "category": "Kategorija objave",  "comments":[],  "image": " …nHYIjaipRIzPnhOO0oHBI1BEPZvIAvfZlei+yPhuQFLj/YDN3B+LUUrzIXHyw7PlbepO1kwxffdrPMUPeGZus38AtAZTE7revuEkyBpb/bR3zihmz36LgGO0bwSq/HN/uqa/HEg3P1Wnv6QW/Llg7ZVT7W1jYTTTSB1FDNLzszPjWSGG8kwOwLLEZol2AKCKAc4jM2Eb2Mr04gU/6+DFI3k4FcIuh9jnZ9Q/DXEDpb647D5AOY4ufu9J3gQWcuMxWYGpT+KNT4A99jW51mRYMtFsBbo+oTQmvs3+cXCg1L8wnGjnvvMI4DSOK3+SbqRc0o8BZD2EgcOBmnlPh15dv8EPgmHTnoqUjYg…",  "userId": "userId",  "ratio":2,  "archived":false,  " 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",  "category": "Kategorija objave",  "comments":[],  "image": " …nHYIjaipRIzPnhOO0oHBI1BEPZvIAvfZlei+yPhuQFLj/YDN3B+LUUrzIXHyw7PlbepO1kwxffdrPMUPeGZus38AtAZTE7revuEkyBpb/bR3zihmz36LgGO0bwSq/HN/uqa/HEg3P1Wnv6QW/Llg7ZVT7W1jYTTTSB1FDNLzszPjWSGG8kwOwLLEZol2AKCKAc4jM2Eb2Mr04gU/6+DFI3k4FcIuh9jnZ9Q/DXEDpb647D5AOY4ufu9J3gQWcuMxWYGpT+KNT4A99jW51mRYMtFsBbo+oTQmvs3+cXCg1L8wnGjnvvMI4DSOK3+SbqRc0o8BZD2EgcOBmnlPh15dv8EPgmHTnoqUjYg…",  "userId": "userId",  "ratio":2,  "archived":false,  " updatedAt": "2024-01-01T12:00:00Z"  "createdAt": "2024-01-01T12:00:00Z"  }  ] |

**Primer odziva pri napaki:**

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

#### 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",  "category": "Kategorija objave",  "userId": "userId",  }  ] |

**Primer odziva pri napaki:**

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

#### 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",  "category": "Kategorija objave",  "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."  } |

#### Označevanje objave kot arhivirano

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

Ta metoda omogoča označevanje objave kot arhivirane. Arhivirana objava ne bo več prikazana v glavnem seznamu objav.

**Primer odziva:**

|  |
| --- |
| {  "\_id": "postId",  "title": "Primer objave",  "content": "Vsebina objave",  "category": "Kategorija objave",  "userId": "userId",  "archived": true,  "createdAt": "2024-01-01T12:00:00Z",  "updatedAt": "2024-01-01T12:00:00Z"  } |

**Primer odziva pri napaki:**

|  |
| --- |
| {  "message": "Error when archiving post."  } |

#### Prekinitev arhiviranja objave

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

Ta metoda omogoča odstranitev oznake arhiviranja z objave in jo ponovno prikaže v glavnem seznamu objav.

**Primer odziva:**

|  |
| --- |
| {  "\_id": "postId",  "title": "Primer objave",  "content": "Vsebina objave",  "category": "Kategorija objave",  "userId": "userId",  "archived": false,  "createdAt": "2024-01-01T12:00:00Z",  "updatedAt": "2024-01-01T12:00:00Z"  } |

**Primer odziva pri napaki:**

|  |
| --- |
| {  "message": "Error when unarchiving post."  } |

### Upravljanje komentarjev

#### Dodaj komentar k objavi

**Pot: POST /post/:id/coment**

Doda nov komentar k določeni objavi na podlagi njenega ID-ja. Obvezni polji sta userId in vsebina (content).

**Primer odziva:**

|  |
| --- |
| [  {  "\_id": "postId",  "upvotes": 0,  "downvotes": 0,  "title": "Primer objave",  "content": "Vsebina objave",  "category": "Kategorija objave",  "comments": [  {  "\_id": "commentId",  "content": "Primer komentarja",  "userId": "userId",  "image": "slika.jpg"  }  ],  "userId": "userId",  "ratio":2,  "archived":false,  " updatedAt": "2024-01-01T12:00:00Z"  "createdAt": "2024-01-01T12:00:00Z"  }  ] |

#### Posodabljanje komentarja/

Pot: Put /post/:id/comment/:commentId

Posodabljanje obstoječega komentarja na podlagi ID-ja komentarja.

**Primer odziva:**

|  |
| --- |
| {  "\_id": "commentId",  "content": "Posodobljen komentar",  "userId": "userId",  "image": "novaslika.jpg"  } |

**Primer odziva pri napak:**

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

#### Brisanje komentarja

**Pot: DELETE /post/:id/coment/:comentId**

Opis: Odstrani določen komentar iz objave na podlagi ID-ja komentarja.

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

**Primer odziva pri napaki:**

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

#### Odzivanje z čustvenčeki

**Pot: POST /post/:id/reaction**

Omogoča uporabnikom, da ob objavi reagirajo z enim izmed vnaprej določenih emojijev. Uporabnik lahko spremeni ali odstrani reakcijo.

**Primer odziva:**

|  |
| --- |
| {  "reaction": "❤️",  "userId": "userId",  "createdAt": "2024-01-01T12:00:00Z"  } |

Primer odziva ob napaki:

|  |
| --- |
| {  "message": "Error when adding reaction."  } |

#### Odstranitev odziva z čustvenčki

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

Odstrani reakcijo z emoji na objavo, če je bila dodeljena uporabniku. Uspešen odziv: Brez vsebine, HTTP statusna koda 204.

**Primer odziva pri napaki:**

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

## Category API

### Iskanje objav po kategoriji

**Pot: GET /:category**

Ta metoda omogoča iskanje objav v določeni kategoriji. Kategorija je podana kot parameter v URL-ju. Iskanje je neobčutljivo na velikost črk in išče po vseh objavah, ki ustrezajo kategoriji.

**Primer odziva:**

|  |
| --- |
| [  {  "\_id": "postId1",  "userId": {  "username": "user1",  "avatar": "http://example.com/avatar1.jpg"  },  "category": "technology",  "content": "Objava o novih tehnologijah.",  "createdAt": "2025-01-20T12:30:00Z",  "archived": false  },  {  "\_id": "postId2",  "userId": {  "username": "user2",  "avatar": "http://example.com/avatar2.jpg"  },  "category": "technology",  "content": "Nadaljnji prispevek o napredku v tehnologiji.",  "createdAt": "2025-01-19T10:15:00Z",  "archived": false  }  ] |

**Primer odziva ob napaki:**

|  |
| --- |
| {  "message": "Error when retrieving posts by category."  } |

# 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**: s

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.