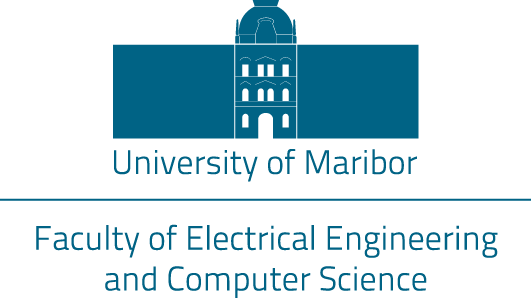
**YourVoice - Projektna dokumentacija**



**Razvoj in upravljanje programskih sistemov**

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

**Vzdrževalci:** Andraž Podpečan, Rok Žerdoner, Rok Porekar(skupina Boop)

Vsebina

[1. Opis projekta 4](#_Toc182266885)

[2. Zaledni del 5](#_Toc182266886)

[2.1. User API 5](#_Toc182266887)

[2.1.1. Pregled 5](#_Toc182266888)

[2.1.2. Namestitev in konfiguracija 5](#_Toc182266889)

[2.1.3. Uporabi API-ja 6](#_Toc182266890)

[2.1.3.1. Pridobitev seznama uporabnikov 6](#_Toc182266891)

[2.1.3.2. Pridobitev enega uporabnika 6](#_Toc182266892)

[2.1.3.3. Registracija uporabnika 7](#_Toc182266893)

[2.1.3.4. Prijava uporabnika 8](#_Toc182266894)

[2.1.3.5. Posodobitev uporabnikovih podatkov 9](#_Toc182266895)

[2.1.3.6. Brisanje uporabnika 10](#_Toc182266896)

[2.2. Post API 10](#_Toc182266897)

[2.2.1. Upravljanje objav 10](#_Toc182266898)

[2.2.1.1. Pridobitev seznama objav 10](#_Toc182266899)

[2.2.1.2. Pridobitev posamezne objave 11](#_Toc182266900)

[2.2.1.3. Kreiranje objave 12](#_Toc182266901)

[2.2.1.4. Posodabljanje objave 12](#_Toc182266902)

[2.2.1.5. Brisanje objave 13](#_Toc182266903)

[2.2.2. Upravljanje komentarjev 13](#_Toc182266904)

[2.2.2.1. Dodaj komentar k objavi 13](#_Toc182266905)

[2.2.2.2. Brisanje komentarja iz objave 14](#_Toc182266906)

[3. Čelni del 14](#_Toc182266907)

[3.1. Tehnične Zahteve 15](#_Toc182266908)

[3.2. Struktura Datotek 15](#_Toc182266909)

[3.2.1. App.tsx 15](#_Toc182266910)

[3.2.2. Routes (routes.tsx) 16](#_Toc182266911)

[3.2.3. UserContext (userContext.ts) 16](#_Toc182266912)

[3.3. Glavne Komponente 17](#_Toc182266913)

[3.3.1. Home.tsx 17](#_Toc182266914)

[3.3.2. Posts.tsx 17](#_Toc182266915)

[3.3.3. PostDetail.tsx 17](#_Toc182266916)

[3.3.4. AddPostModal.tsx 18](#_Toc182266917)

[3.3.5. Register.tsx in Login.tsx 18](#_Toc182266918)

[3.3.6. Profile.tsx 19](#_Toc182266919)

[3.3.7. Header.tsx 19](#_Toc182266920)

[3.3.8. Footer.tsx 19](#_Toc182266921)

[3.4. Ostalo 20](#_Toc182266922)

[3.4.1. ProtectedRoute.tsx 20](#_Toc182266923)

[3.4.2. Upravljanje Uporabniškega Stanja (UserContext.ts) 20](#_Toc182266924)

[3.5. Uporabniška Izkušnja in varnost 20](#_Toc182266925)

# 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.

OPIS VZDRŽEVALCI:

Poleg vsebine lahko uporabniki nalagajo tudi slike v objavah, ocenjujejo v obliki »všečkaj« gumba, filtrirajo glede na kategorije, čas in ime objave, objave se jim dinamično (sproti) prikazujejo s pomočjo oštevilčenja strani ter jim je omogočena tudi izbira avatarja.

# 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

ZALEDNI DEL VZDRŽEVALCI:

Uporabljene knjižnice:

* multer: za nalaganje slik (datotek), ki se pridobijo s pomočjo zahtevka na node.js strežnik, na katerem se nahaja zaledni del

## 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.

PREGLED VZDRŽEVALCI:

API prav tako omogoča, da lahko uporabniki naložijo svojo profilno sliko (avatar) in jo posodobijo oziroma izbrišejo, hkrati pa je sedaj spreminjanje uporabniških nastavitev izdelano tako, da lahko uporabnik spreminja svoj opis ter avatarja brez vnosa gesla, za ostale (njegovo uporabniško ime, elektronski naslov in novo geslo) nastavitve pa le, če je pravilno vnesel tudi svoje trenutno (staro) geslo. Poleg vsega tega, pa je v trenutni implementaciji dodano tudi preverjanje uporabniških vnosov pri registraciji in posodobitvi računa (elektronsko naslov in uporabniško ime potrebujeta biti unikatna, elektronski naslov mora biti veljaven, uporabniško ime dolgo vsaj 3 znake, geslo potrebuje biti dolgo vsaj 8 znakov z vsaj eno števko, veliko črko ter posebnim simbolom). Dodani so tudi opisi napak v rezultatih zahtevkov.

### 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"  //VZDRŽEVALCI  »avatar«: {  »imageName«: »32442331avatar123.jpg«  »imageUrl«: »/images/avatar123.jpg«  }, // avatar je lahko tudi prazen objekt  }  ] |

#### 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"  //VZDRŽEVALCI  »avatar«: {  »imageName«: »32442331avatar123.jpg«  »imageUrl«: »/images/avatar123.jpg«  }, // avatar je lahko tudi prazen objekt  } |

**Primer odziva pri napaki:**

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

#### Registracija uporabnika

**Pot: POST /user**

Ustvari novega uporabnika z vnesenimi parametri.

REGISTRACIJA VZDRŽEVALCI

Pri vnosu se preveri, če je elektronski naslov še neuporabljen in v veljavnem formatu (prejšnja verzija je imela nekatere pomanjkljivosti pri tem), če je uporabniško ime dolgo vsaj 3 znake in še ne uporabljeno ter, če ima geslo vsaj 8 znakov, vsaj eno veliko črko, številko ter poseben znak.

**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  }  }  } |

VZDRŽEVALCI: Dodani opisi napak, ki se pojavijo

|  |
| --- |
| {  "message": "Email ni v veljavnem formatu. Če ste se zatipkali, prosimo poskusite znova vnesti veljaven email"  // ALI  "message": "Geslo mora biti dolgo med 8 in 30 znakov in vsebovati vsaj eno veliko črko in vsaj eno številko!!!"  // ALI  "message": " Napaka strežnika - Težava pri shranjevanju gesla. Prosimo poskusite ponovno čez nekaj trenutkov"  // ALI ….  } |

#### 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

POSODOBITEV VZDRŽEVALCI:

Največja posodobitev pri uprabniškem delu API-ja, je bil pri posodabljanju uporabniških nastavitev. API za posodobitve sedaj omogoča nalaganje, spreminjanje in odstranjevanje avatarja, pri čemer je omejitev na 1 sliko v velikosti max 5MB in tipa .pmg, .jpeg ali .gif. Funkcija omogoča spreminjanje biografije in avatarja brez potrebe po geslu uporabnika, pri spreminjanju imena, e-naslova ali gesla, pa je potreebno podati tudi uporabnikovo predhodno geslo. V primeru napak se sedaj vračajo v zahtevku rezultata tudi opisi napak.

**Primer odziva ob uspehu:**

|  |
| --- |
| {  "message": "User updated successfully"  }  **// VZDRŽEVALCI:**  {  "message": "Uporabniške nastavitve uspešno posodobljene"  } |

**Primer odziva ob napaki:**

|  |
| --- |
| {  "message": "Geslo mora biti dolgo vsaj 8 znakov in vsebovati vsaj eno veliko črko in eno številko."  }  // VZDRŽEVALCI:  {  "message": "Email ni v veljavnem formatu. Če ste se zatipkali, prosimo poskusite znova vnesti veljaven email"  // ALI  "message": "Geslo mora biti dolgo med 8 in 30 znakov in vsebovati vsaj eno veliko črko in vsaj eno številko!!!"  // ALI  "message": " Napaka strežnika - Težava pri shranjevanju gesla. Prosimo poskusite ponovno čez nekaj trenutkov"  // ALI ….  } |

#### 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.

// POST VZDRŽEVALCI

API dodatno omogoča shranjevanje, posodabljanje in brisanje slik znotraj objav, pridobivanje objav glede na podane parametre (filtriranje glede na ime, datum in kategorijo) ter shranjevanje »všečkov« za posamezno objavo s pomočjo transakcij.

### 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":[],  "userId": "userId",  " updatedAt": "2024-01-01T12:00:00Z"  "createdAt": "2024-01-01T12:00:00Z"  // VZDRŽEVALCI  »images«: [  {  »dateCreated«: "2024-01-01T12:00:00Z",  »imageName«: »483248283ImageTest,  »imageUrl«: »images/483248283ImageTest.png,  } ], lahko tudi prazen seznam, ali pa ta seznam vsebuje do 10 slik  }  ] |

#### 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":[],  "userId": "userId",  " updatedAt": "2024-01-01T12:00:00Z"  "createdAt": "2024-01-01T12:00:00Z"  // VZDRŽEVALCI  »images«: [  {  »dateCreated«: "2024-01-01T12:00:00Z",  »imageName«: »483248283ImageTest,  »imageUrl«: »images/483248283ImageTest.png,  } ], lahko tudi prazen seznam, ali pa ta seznam vsebuje do 10 slik  }  }  ] |

**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).

// VZDRŽEVALCI

Zahtevek lahko prav tako dobi v svojem telesu seznam slik, v katerim je za vsako sliko zabeležen datum kreiranja, njeno ime in pot znotraj strežnika, kjer se bo fizično shranila. Slike se vse shranjujejo znotraj direktorija public/images. Za potrebe shranjevanja se je ustvarila pomožna funkcija imenovana »createUploader«, ki poskrbi za pravilno shranjevanje (komunicira s knjižnico muter).

**Primer odziva:**

|  |
| --- |
| [  {  "title": "Primer objave",  "content": "Vsebina objave",  "category": "Kategorija objave",  "userId": "userId",  // VZDRŽEVALCI  »images«: [  {  »dateCreated«: "2024-01-01T12:00:00Z",  »imageName«: »483248283ImageTest,  »imageUrl«: »images/483248283ImageTest.png,  }  ], lahko tudi prazen seznam, ali pa ta seznam vsebuje do 10 slik  }  ] |

**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).

VZDRŽEVALCI:

Podobno kot pri nalaganju je tudi tukaj dodatno omogočeno nalaganja slik preko pomožne funkcije »create uploader«. Sami funkciji se je dodala tudi možnost brisanja datotek tako iz baze, kot tudi iz mape »images«, saj v primeru, da znotraj zahtevka ni slike, ki bi morala biti se privzame, da želi uporabnik jo izbrisati. Zaradi varčevanja s shrambo je slika ne samo virtualno, temveč tudi fizično izbrisana iz strežnika.

**Primer odziva:**

|  |
| --- |
| [  {  "title": "Primer objave",  "content": "Vsebina objave",  "category": "Kategorija objave",  "userId": "userId",  // VZDRŽEVALCI  »images«: [  {  »dateCreated«: "2024-01-01T12:00:00Z",  »imageName«: »483248283ImageTest,  »imageUrl«: »images/483248283ImageTest.png,  }  ], lahko tudi prazen seznam, ali pa ta seznam vsebuje do 10 slik  }  }  ] |

**Primer odziva pri napaki:**

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

#### Brisanje objave

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

Izbriše določeno objavo na podlagi njenega ID-ja.

VZDRŽEVALCI:

Izbrišejo se tudi vse slike, ki so povezane s to objavo.

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

**Primer odziva pri napaki:**

|  |
| --- |
| {  "message": "Error when deleting 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",  }  ],  "userId": "userId",  " updatedAt": "2024-01-01T12:00:00Z"  "createdAt": "2024-01-01T12:00:00Z"  }  ] |

#### Brisanje komentarja iz objave

**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."  } |

# 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.

ČELNI DEL VZDRŽEVALCI

Na čelnem delu je uporabnikom omogočeno tudi filtriranje objav glede na ime, čas ali njihovo kategorijo. Zaradi optimizacijskih razlogov se objave nalagajo po straneh (sprotno nalaganje), uporabnikom pa je omogočeno objavljanje tako v obliki teksta, kot tudi slik (pri čemere se omejeni na največ 10 slik). Poleg komentiranja, lahko uporabniki tudi »všečkajo« objave, ki so jim všeč, ali »ne všečkajo« tistih, ki jim niso. Objave se prikazujejo urejeno glede na čas ali ime, pri čemer lahko uporabnik to nastavljivo izbira. Nalaganje objav, komentiranje in »všečkanje« je dovoljeno le prijavljenim uporabnikom, ki lahko tudi kasneje znotraj zavihka profil urejajo le tega (lahko imajo tudi naloženo profilno sliko). Uporabniki se med sabo sedaj lahko delijo glede na vloge, pri čemer lahko tisti, ki so »administrativne narave« blokirajo, omejijo delovanje drugim uporabnikom (komentiranje, ali ustvarjanje objav) ter tudi v celoti odstranijo.

Poleg vsega navedenega, pa so uporabnikom vidne tudi modalna okna, ki jih ozaveščajo o njihovih akcijah ali napakah pri izvedbi le teh.

## 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

VZDRŽEVALCI

* **knjižnice in orodja:**
  + Chakra Icon – za prikazovanje ikon (del Chakra UI knjižnice)

## 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.

VZDRŽEVALCI

* **Funkcionalnosti**
  + Pri vsaki objavi se prikaže prva shranjena slika, ki pripada posamezni objavi (če ta ne obstaja, se prikaže objava brez slike)
  + V modalnem okno je dodana možnost nalaganja do največ 10 slik na posamezno objavo. Te so tudi prikazane znotraj modalnega okna, in jih lahko kasneje uporabnik tudi odstrani

### 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.

VZDRŽEVALCI:

* **Funkcionalnosti**
  + Znotraj te strani se prikazujejo tudi vse slike, ki so vezane na njo v obliki razreda ImageSlider. Torej uporabnik lahko s pomočjo pomožnih gumbov preklaplja med prikazi različnih slik, ki pripadajo tej objavi. Če obstaja samo ena slika, so ti gumbi onemogočeni.

### 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.

VZDRŽEVALCI

* **Funkcionalnosti**
  + Dodana funkcionalnost za dodajanje slik
  + Ob rezultatu se prikaže lastno implementirano modalno okno, ki prikaže rezultate zahtevkov s strani strežnika

### 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.

VZDRŽEVALCI

* **Funkcionalnosti**
  + Implementirana verifikacija vnosnih polj pri registraciji. Preden se pošlje zahteva na strežnik, se preveri, če nobeno polje, ki je zahtevano ni prazno, uporabniško ime dolgo vsaj 3 znake, geslo in elektronski naslov pa v pravilnem formatu
  + Ko se pridobi odgovor s strani strežnika, se to sporočilo prikaže v obliki lastnega modalnega okna

### 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.

VZDRŽEVALCI

* **Funkcionalnosti**
  + Dodana funkcionalnost spreminjanja uporabniških informacij v obliki modalnega okna
  + Nalaganje in prikaz avatarja
  + Spreminjanje avatarja in opisa brez podajanja gesla
  + Če uporabnik poda geslo, lahko spremeni tudi svoje ime, email ali posodobi geslo
  + Prikaz rezultatov zahtevkov sprememb v obliki lastnega modalnega okna

### 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.

**VZDRŽEVALCI**

### CustomModalProps.tsx

* **Opis**: Modalno okno za prikaz obvestil uporabnikom, kateremu lahko spreminjamo barvo, tekst ali čas prikazovanja
* **Dostop**: Javno dostopni.
* **Funkcionalnosti**:
  + Podajanje naslova, besedila, barve, časa trajanja ter njegove vidnosti
* **Uporabljene knjižnice**: Chakra UI

### FileUpload.tsx

* **Opis**: Pomožna komponenta za nalaganje in prikaz slik znotraj oken za dodajanje ali spreminjanje podatkov.
* **Dostop**: Javno dostopni.
* **Funkcionalnosti**:
  + Uporabnik lahko z njegovo pomočjo izbere željene datotek iz svoje lokalne shrambe
  + Uporabnik lahko odstrani slike, ki so naložene na strežniku
  + Uporabniku se prikaže pomanjšana verzija datoteke znotraj okna
* **Uporabljene knjižnice**: Chakra UI

### ImageSlideShow.tsx

* **Opis**: Komponenta, ki prikaže prejete slike v obliki t.i. »slide show« projekcije
* **Dostop**: Javno dostopni.
* **Funkcionalnosti**:
  + Kot argument se poda polje slik, ki se po vrsti nato dodajo v to diaprojekcijo
  + Gumb za naslednjo ali prejšnjo sliko. Če sta podani manj kot 2 sliki, sta gumba onemogočena
* **Uporabljene knjižnice**: Chakra UI

## 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.

VZDRŽEVALCI

* **Prijaznost uporabniškega vmesnika**:
  + Lastno implementirano okno, ki prikaže napake, ko se le te zgodijo
  + Implementacija »slide show« projekcije na strani podrobno o objavi
  + Filtriranje objav glede na ime, čas ali kategorijo
  + Ocenjevanje s klikom na gumb »všečkaj«
  + Sortiranje objav glede na različne parametre, ki jih uporabnik lahko nastavi
* **Upravljanje napak**:
  + Modalno okno, ki prikaže uporabniku, kaj je bila težava
  + Verifikacija vnosnih polj pri registraciji

# Dopolnjevanje projekta

Ta odsek je namenjen strnitvi vseh posodobitev, ki jih je naša »Beep« ekipa postorila tekom teh treh šprintov. **Vse spremembe so bile omenjene tudi znotraj prejšnjih poglavji, pri čemer se nahajaj znotraj naslovji VZDRŽEVALCI, ki se lahko nahajajo v poljubnem poglavju**.

## Opis posodobitev

* Nalaganje slik tako objavam kot tudi profilom (avatarji)
* Filtriranje objav glede na kategorije
* Sortiranje prikaza objav glede na čas in ime
* Prikazovanje objav znotraj oštevilčenih strani in navigiranje znotraj njih
* Posodobitev UI na čelnem delu
* Dodajanje gumbov za »všečkanje« in »ne všečkanje«
* Prikazovanje slik v obliki »slide show« projekcije
* Spreminjanje uporabniških informacij
* Dodajanje funkcionalnosti administracije, ki omogoča brisanje objav in komentarjev ter blokiranje uporabniških računov
* Transakcije pri spreminjanju podatkov znotraj baze
* Validacija uporabniških podatkov znotraj registracije in znotraj uporabniških vnosnih polj
* Modalno okno, ki prikazuje obvestila (predvsem uspešno opravljene akcije, ali napak)

## Testiranja