

Izrada web aplikacije za upravljanje projektima	Verzija: <1.0>
Tehnička dokumentacija	Datum: 18.01.2022.

# **ProjControl** **Tehnička dokumentacija** **Verzija <1.0>**

**Studentski tim:**  
 Filip Marčec  
 Marko Okreša  
 Filip Cindrić  
 Luka Cavalli  
 Mihael Cugovčan  
 Dominik Matijaca

**Nastavnik:** Krešimir Fertalj

Izrada web aplikacije za upravljanje projektima	Verzija: <1.0>
Tehnička dokumentacija	Datum: 18.01.2022.

## Sadržaj

1.	Opis razvijenog proizvoda	3
1.1	Motivacija i cilj razvijanja proizvoda	3
1.2	Opis razvijenog proizvoda	3
1.3	Slični projekti	3
1.3.1	YouTrack	3
1.3.2	Monday	5
1.3.3	JIRA	6
2.	Tehničke značajke	8
2.1	Korištene tehnologije	8
2.2	Struktura repozitorija i sadržaj pojedinih dijelova	9
2.3	Dijagram komponenti	17
2.4	Dijagram klasa	18
2.5	Baza podataka	20
3.	Upute za korištenje	24
3.1	funkcionalnost – registracija	24
3.2	funkcionalnost – prijava	25
3.3	funkcionalnost – prikaz svih projekata	26
		26
3.4	funkcionalnost – dashboard projekta	26
		27
3.5	funkcionalnost – pregled aktivnih zadataka	28
3.6	funkcionalnost – pregled ostalih zadataka	30
3.7	funkcionalnost – pregled zaposlenika	31
3.8	funkcionalnost – dodavanje novog ili uklanjanje postojećeg zaposlenika na projektu	31
3.9	funkcionalnost – pregled grafova	32
3.10	funkcionalnost – dodavanje novog projekta	33
3.11	funkcionalnost – dodavanje novog zadatka	34
3.12	funkcionalnost – pregled zadataka na trenutnom projektu	34
4.	Literatura	35

Izrada web aplikacije za upravljanje projektima	Verzija: <1.0>
Tehnička dokumentacija	Datum: 18.01.2022.

# Tehnička dokumentacija

## 1. Opis razvijenog proizvoda

### 1.1 Motivacija i cilj razvijanja proizvoda

Prilikom vođenja projekta, voditelj projekta nailazi na mnoge prepreke. Mnoge organizacije često preambiciozno postavljaju zadane rokove isporuke, ne uzimajući pritom u obzir realne uvjete poput mogućih kašnjenja, nepredviđenih događaja i slično. Izradom web aplikacije za upravljanje projektom želi se postići racionalno planiranje projekta. Voditelju projekta želi se omogućiti izdavanje novih zadataka i postavljanje kontrolnih točaka, pregled dosad odrađenih zadataka te cjelokupan napredak na projektu. Smatramo da će se korištenjem web aplikacije za upravljanje projektima povećati efikasnost, minimizirati rizici te optimizirati raspoređenost zaposlenih na projektu.

Cilj projekta je izraditi web aplikaciju za upravljanje projektima koja će omogućiti lakše upravljanje projektima, bolju preglednosti i povećanu efikasnost.

### 1.2 Opis razvijenog proizvoda

Rezultat projekta jest web aplikacija „ProjControl“ te upute za njeno korištenje. Web aplikaciji se pristupa preko linka : [projcontrol.herokuapp.com](http://projcontrol.herokuapp.com) . Potrebno je izvršiti registraciju i potom se može započeti s radom. Aplikacija omogućava :

- kreiranje novih projekata
- dodavanje zadataka na postojeće projekte
- pregled svih zadataka te njihovo filtriranje i sortiranje
- dodavanje zaposlenika na projekt
- pregled statistike zaposlenika
- pregled grafova o radu na projektu
- uređivanje podataka o zadacima i o projektima
- pregled vlastitog rada i vlastitih zadataka

Tehnička dokumentacija sadrži detaljniji pregled tehničke strane projekta – arhitekture, strukture repozitorija, dijagrami baze podataka, dijagram komponenti i dijagram klasa, te naravno korisničke upute za korištenje aplikacije.

### 1.3 Slični projekti

Analizirali smo 3 aplikacije čija je zadaća upravljanje projektima.

#### 1.3.1 *YouTrack*

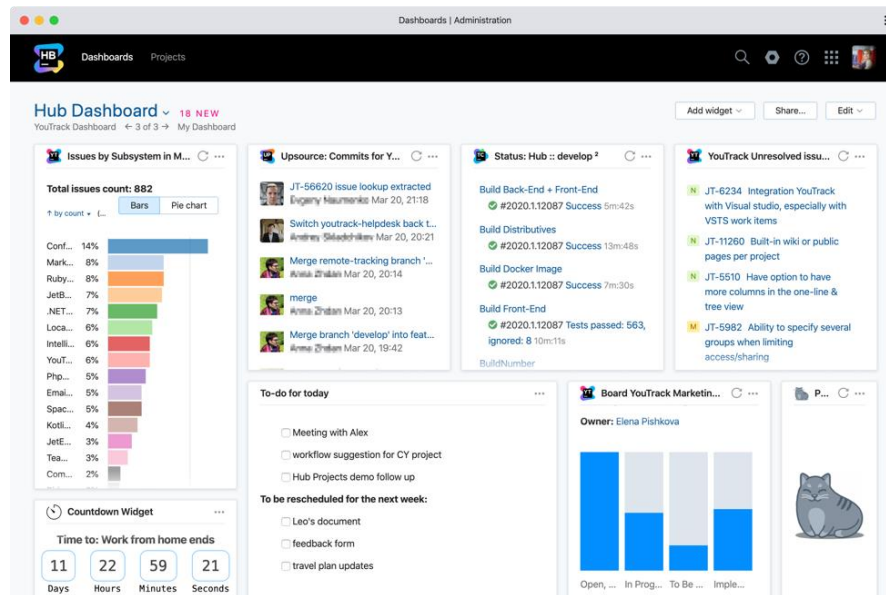
YouTrack(JetBrains) je alat za upravljanje projektima koji se može prilagoditi korisničkim procesima kako bi mu pomogao u isporuci svojih proizvoda.

Popis funkcionalnosti:

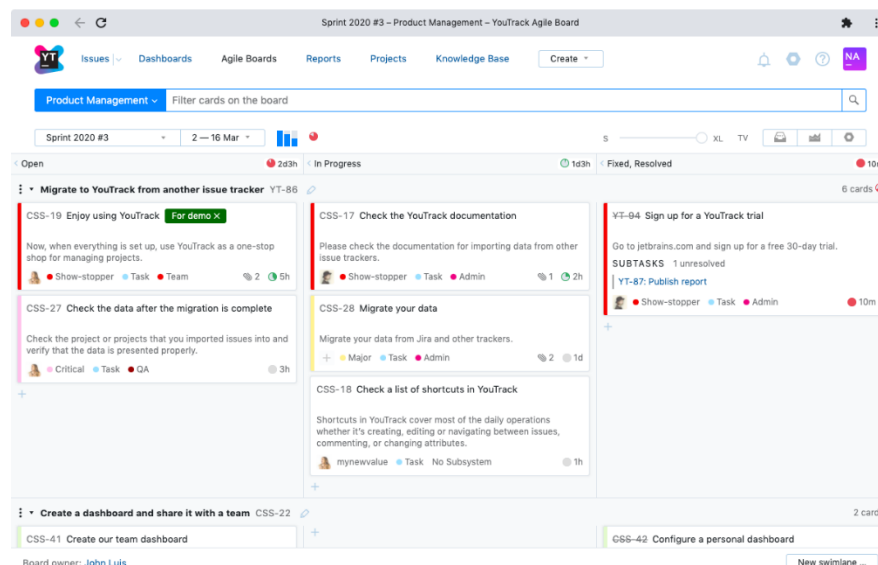
- praćenje projekata i zadataka
- korištenje agilne ploče

Izrada web aplikacije za upravljanje projektima	Verzija: <1.0>
Tehnička dokumentacija	Datum: 18.01.2022.

- planiranje sprintova i izdanja
- vođenje baze znanja
- rad s izvješćima i nadzornim pločama
- davanje prioriteta zadacima
- otvaranje i zatvaranje problema
- prikaz napretka u vremenskim tablicama



Slika 1. Korisničko sučelje aplikacije YouTrack



Slika 2. Prikaz popisa zadatka u aplikaciji YouTrack

Izrada web aplikacije za upravljanje projektima	Verzija: <1.0>
Tehnička dokumentacija	Datum: 18.01.2022.

### 1.3.2 Monday

Monday.com je prilagodljiva web i mobilna platforma koja olakšava upravljanje radom. Dizajnirana je da pomaže organizacijama i timovima povećati efektivnost i produktivnost u radu.

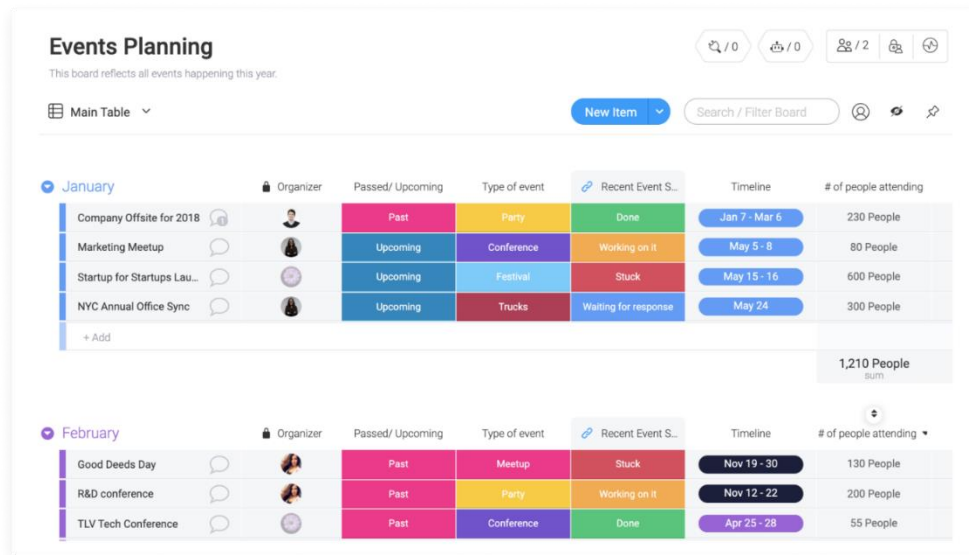
Popis funkcionalnosti:

- praćenje projekta i tijeka rada
- korištenje agilne ploče
- planiranje sprintova i izdanja
- rad s izvješćima i nadzornim pločama
- davanje prioriteta zadacima
- otvaranje i zatvaranje problema
- komunikacija između članova tima
- korištenje unaprijed pripremljenih predložaka
- slanje obavijesti članovima tima
- s *third-party* aplikacijama



Slika 3. Prikaz korisničkog sučelja u aplikaciji Monday.com

Izrada web aplikacije za upravljanje projektima	Verzija: <1.0>
Tehnička dokumentacija	Datum: 18.01.2022.



**Events Planning**  
This board reflects all events happening this year.

Main Table New item Search / Filter Board

January	Organizer	Passed/ Upcoming	Type of event	Recent Event S...	Timeline	# of people attending
Company Offsite for 2018		Past	Party	Done	Jan 7 - Mar 6	230 People
Marketing Meetup		Upcoming	Conference	Working on it	May 5 - 8	80 People
Startup for Startups Lau...		Upcoming	Festival	Stuck	May 15 - 16	600 People
NYC Annual Office Sync		Upcoming	Trucks	Waiting for response	May 24	300 People
+ Add						
						1,210 People Sum

February	Organizer	Passed/ Upcoming	Type of event	Recent Event S...	Timeline	# of people attending
Good Deeds Day		Past	Meetup	Stuck	Nov 19 - 30	130 People
R&D conference		Past	Party	Working on it	Nov 12 - 22	200 People
TLV Tech Conference		Past	Conference	Done	Apr 25 - 28	55 People

Slika 4. Prikaz planiranja događaja u aplikaciji Monday.com

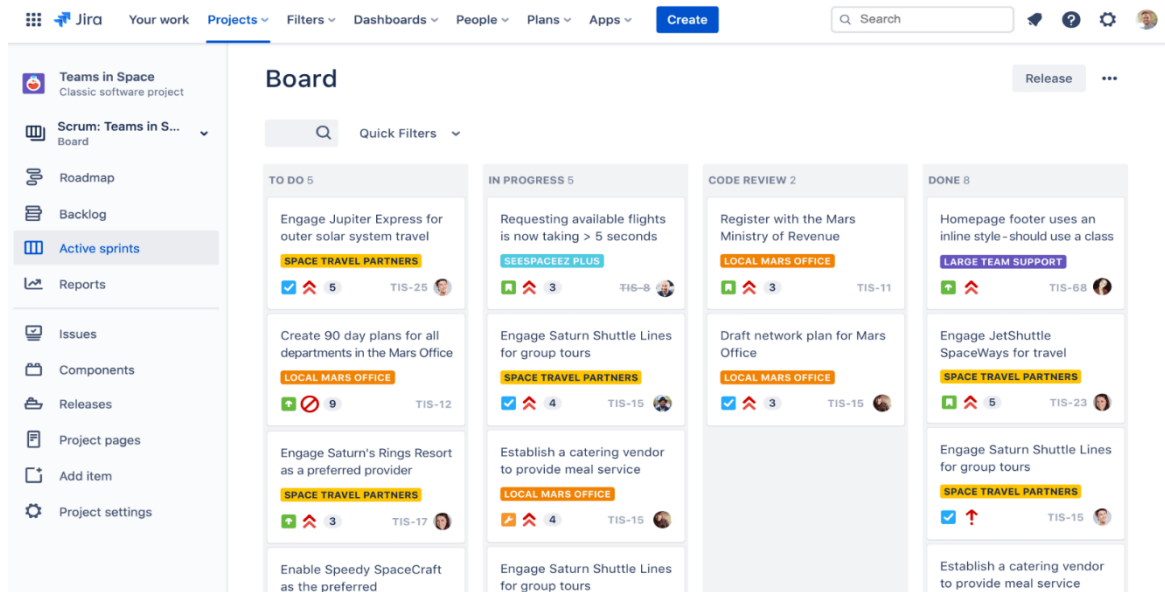
### 1.3.3 JIRA

JIRA je popularan softver za upravljanje projektima, kojim se služe agilni timovi za razvoj softvera.

Popis funkcionalnosti:

- praćenje projekta i tijeka rada
- korištenje agilne ploče
- planiranje sprintova i izdanja
- korištenje kanban ploča
- rad s izvješćima i nadzornim pločama
- davanje prioriteta zadacima
- otvaranje i zatvaranje problema
- komunikacija između članova tima

Izrada web aplikacije za upravljanje projektima	Verzija: <1.0>
Tehnička dokumentacija	Datum: 18.01.2022.



Slika 5. Korisničko sučelje aplikacije JIRA

Izrada web aplikacije za upravljanje projektima	Verzija: <1.0>
Tehnička dokumentacija	Datum: 18.01.2022.

## 2. Tehničke značajke

### 2.1 Korištene tehnologije

**ReactJS**[1] – biblioteka bazirana na JavaScript-u za razvoj i kreiranje korisničkih sučelja, te jednostavno preusmjerivanje na klijentskoj aplikaciji.

**ChartJS**[2] – dodatak za JavaScript za kreiranje raznih dijagrama i grafikona.

**NodeJS**[3] – backend JavaScript okolina izvođenja koja služi za izvođenje JavaScript koda izvan web preglednika.

**ExpressJS**[4] – biblioteka za NodeJS za upravljanje zahtjevima na backend-u, može se koristiti kao REST API za upravljanje CRUD operacijama.

**PostgreSQL**[5] – sustav za upravljanje bazama podataka otvorenog koda.

**Node-postgres**[6] – biblioteka za NodeJS za pristup i upravljanje PostgreSQL bazama podataka.

**Nodemailer**[7] – biblioteka za NodeJS koja omogućuje slanje e-poruka korisniku iz backend-a aplikacije.

Za pokretanje aplikacije lokalno potrebno je imati instaliran npm, te instalirati potrebne node module pisanjem komande „npm install“. Zatim je potrebno posebno pokrenuti frontend (koji se nalazi u mapi client) pisanjem komande „npm start“ i backend (koji se nalazi u mapi server) pisanjem komande „node server.js“.

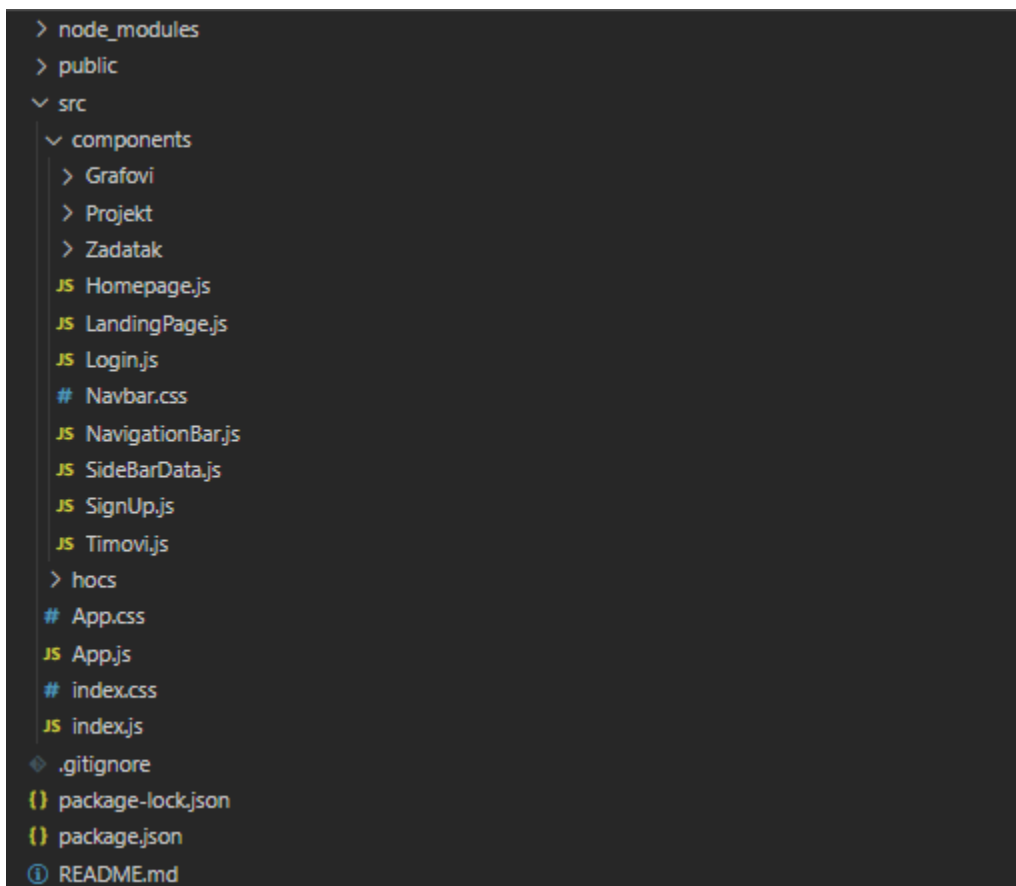
Aplikacija je također dostupna na web-adresi <https://projcontrol.herokuapp.com>



Izrada web aplikacije za upravljanje projektima	Verzija: <1.0>
Tehnička dokumentacija	Datum: 18.01.2022.

## 2.2 Struktura repozitorija i sadržaj pojedinih dijelova

Repozitorij se sastoji od 2 glavna dijela, „client“ i „server“. U dijelu „client“ nalazi se se frontend dio aplikacije.



Slika 6. Repozitorij „client“

### Mapa „components“

Mapa „components“ sadrži sve JavaScript datoteke u kojima se nalaze metode za koje šalju zahtjev za dohvatom podataka ili šalju podatke, te pripadni HTML kod koji prikazuje te podatke korisniku. Ona još u sebi sadrži tri mape : „Grafovi“, „Projekt“ te „Zadatak“ u kojima su grupirane datoteke koje su vezane uz jedno područje.

Slika 7. prikazuje kako izgleda tipična datoteka u mapi „components“, odnosno njen dio u kojem se dohvacaju i organiziraju podaci, a slika 8. prikazuje kako izgleda HTML kod za njihov prikaz.

Izrada web aplikacije za upravljanje projektima	Verzija: <1.0>
Tehnička dokumentacija	Datum: 18.01.2022.

```

JS Login.js X
src > components > JS Login.js > [e] Login > [e] onSubmitForm
1  import React, { Fragment, useState } from "react";
2  import { toast } from "react-toastify";
3  import { useHistory } from 'react-router-dom';
4
5  const Login = ({ setAuth }) => {
6      const [inputs, setInputs] = useState({
7          email: "",
8          password: ""
9      });
10
11     let history = useHistory();
12
13     const { email, password } = inputs;
14
15     const onChange = e =>
16     setInputs({ ...inputs, [e.target.name]: e.target.value });
17
18     const onSubmitForm = async e => {
19         e.preventDefault();
20         try {
21             const body = { email, password };
22             const response = await fetch(
23                 "http://localhost:5000/auth/login",
24                 {
25                     method: "POST",
26                     mode: "cors",
27                     headers: {
28                         "Content-type": "application/json"
29                     },
30                     body: JSON.stringify(body)
31                 }
32             );
33
34             const parseRes = await response.json();
35
36             if (parseRes.jwtToken) {
37                 localStorage.setItem("token", parseRes.jwtToken);
38                 localStorage.setItem("user", JSON.stringify(parseRes.data));
39                 setAuth(true);
40                 toast.success("Uspjesna prijava!");
41                 history.push('/landingpage')
42                 window.location.reload(false);
43             } else {
44                 setAuth(false);
45                 toast.error(parseRes);
46             }
47         } catch (err) {
48             console.error(err.message);
49         }
50     };

```

Slika 7. Zahtjev za dohvat poataka

Izrada web aplikacije za upravljanje projektima	Verzija: <1.0>
Tehnička dokumentacija	Datum: 18.01.2022.

```

return (
  <Fragment>
    <div className='signIN'>
      <div className = 'login-naslov'>
        <h1>DOBRODOŠLI NATRAG!</h1>
        <p>Unesite svoje podatke za prijavu:</p>
      </div>
      <div className = 'form-box'>
        <form onSubmit={onSubmitForm} className='formtest'>
          <div className='email-form'>
            <input
              type="text"
              name="email"
              value={email}
              onChange={e => onChange(e)}
              className="form-control-email"
              placeholder = 'Upišite email adresu*'
            />
          </div>
          <div className='pass-form'>
            <input
              type="password"
              name="password"
              value={password}
              onChange={e => onChange(e)}
              className="form-control"
              placeholder = 'Upišite lozinku'
            />
            <button className='anew btn btn-2 navlinkother btn-noborder' type='submit' >Prijava</button>
          </div>
        </form>
      </div>
      <br />
      <div className = 'ostaloLogin'>
        <div className='sign-up-option'>
          Nemaš račun? <a href="/signup" className= 'a6 btn-6'>Registriraj se</a>
        </div>
        <div className='pass-reset-option'>
          Zaboravljena lozinka? <a href="/reset-password" className= 'a6 btn-6'>Obnovi lozinku</a>
        </div>
      </div>
    </div>
  </Fragment>
);

```

Slika 8. HTML kod za prikaz podataka i forme za login

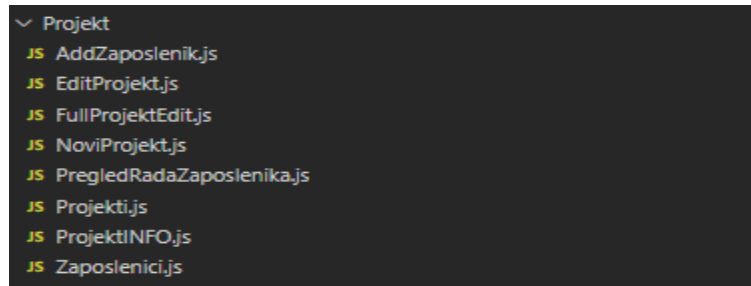
### Mapa „Grafovi“

U mapi grafovi definirana je datoteka „Chart.js“ u kojoj je izvedeno prikazivanje *burndown charta* za pojedini projekt. Za ostvarenje te funkcionalnosti korišten je paket „react-chartjs-2“ koji omogućuje jednostavno definiranje podataka za prikaz i uređivanje izgleda grafa.

Izrada web aplikacije za upravljanje projektima	Verzija: <1.0>
Tehnička dokumentacija	Datum: 18.01.2022.

### Mapa „Projekt“

U mapi „Projekt“ nalaze se sve JavaScript datoteke koji su povezane s prikazom podataka o projektu. Na slici 9. prikazan je njen sadržaj.



Slika 9. Prikaz sadržaja mape „Projekt“

### Mapa „Zadatak“

U mapi „Zadatak“ nalaze se sve JavaScript datoteke koji su povezane s prikazom podataka o projektu. Na slici 9. prikazan je njen sadržaj.

### App.js

App.js glavna je datoteka frontend dijela jer su u njoj definiran *router*(*react-router-dom*) koji određuje koja JavaScript datoteka se mora pokrenuti kako bi se korisniku prikazao dobar prikaz. Slika 10. prikazuje kako taj *router* izgleda.

```

<Router>
  <Layout>
    <Switch>
      <Route exact path="/" render={props => !isAuthenticated ? (<Homepage {...props} />) : (<Redirect to="/landingpage" />)} />
      <Route exact path="/login" render={props => !isAuthenticated ? (<Login {...props} setAuth={setAuth} />) : (<Redirect to="/landingpage" />) } />
      <Route exact path="/signup" render={props => !isAuthenticated ? (<Redirect to="/landingpage"/>) : (<SignUp {...props} setAuth={setAuth} />) } />
      <Route exact path="/landingpage" render={props => isAuthenticated ? (<LandingPage {...props} setAuth={setAuth} />) : (<Redirect to="/login" />) } />
      <Route exact path="/chart/:pid" render={props => isAuthenticated ? (<Charts {...props} setAuth={setAuth} />) : (<Redirect to="/login" />) } />
      <Route exact path="/projekti" render={props => isAuthenticated ? (<Projekti {...props} setAuth={setAuth} />) : (<Redirect to="/login" />)} />
      <Route exact path="/timovi" render={props => isAuthenticated ? (<Timovi {...props} setAuth={setAuth} />) : (<Redirect to="/login" />)} />
      <Route exact path="/zadaci" render={props => isAuthenticated ? (<MojiZadaci {...props} setAuth={setAuth} />) : (<Redirect to="/login" />)} />
      <Route exact path="/noviprojekt" render={props => isAuthenticated ? (<NoviProjekt {...props} setAuth={setAuth} />) : (<Redirect to="/login" />)} />
      <Route exact path="/novizadatak" render={props => isAuthenticated ? (<NoviZadatak {...props} setAuth={setAuth} />) : (<Redirect to="/login" />)} />
      <Route exact path="/svizadaci/:id" render={props => isAuthenticated ? (<SviZadaci {...props} setAuth={setAuth} />) : (<Redirect to="/login" />)} />
      <Route exact path="/projekti/izmjena/:id" render={props => isAuthenticated ? (<EditProjekt {...props} setAuth={setAuth} />) : (<Redirect to="/login" />)} />
      <Route exact path="/zadaci/izmjena/:pid/:id" render={props => isAuthenticated ? (<EditTask {...props} setAuth={setAuth} />) : (<Redirect to="/login" />)} />
      <Route exact path="/zadaci/dodaj/:pid/:id" render={props => isAuthenticated ? (<AddUser {...props} setAuth={setAuth} />) : (<Redirect to="/login" />)} />
      <Route exact path="/userzadatak/izmjena/:id" render={props => isAuthenticated ? (<EditTaskUser {...props} setAuth={setAuth} />) : (<Redirect to="/login" />)} />
      <Route exact path="/projektinfo/:id" render={props => isAuthenticated ? (<ProjektINFO {...props} setAuth={setAuth} />) : (<Redirect to="/login" />)} />
      <Route exact path="/projektinfo/zaposlenici/:id" render={props => isAuthenticated ? (<Zaposlenici {...props} setAuth={setAuth} />) : (<Redirect to="/login" />)} />
      <Route exact path="/zadatak/finish/:id" render={props => isAuthenticated ? (<TaskFinish {...props} setAuth={setAuth} />) : (<Redirect to="/login" />)} />
      <Route exact path="/userzadatak/finish/:id" render={props => isAuthenticated ? (<TaskFinishUser {...props} setAuth={setAuth} />) : (<Redirect to="/login" />)} />
      <Route exact path="/addzaposlenik/:pid" render={props => isAuthenticated ? (<AddZaposlenik {...props} setAuth={setAuth} />) : (<Redirect to="/login" />)} />
      <Route exact path="/projekti/fullizmjena/:id" render={props => isAuthenticated ? (<FullProjektEdit {...props} setAuth={setAuth} />) : (<Redirect to="/login" />)} />
      <Route exact path="/ostalizadaci/:id" render={props => isAuthenticated ? (<OstaliZadaci {...props} setAuth={setAuth} />) : (<Redirect to="/login" />)} />
      <Route exact path="/voditelj-pregled/:id" render={props => isAuthenticated ? (<PregledRadaZaposlenika {...props} setAuth={setAuth} />) : (<Redirect to="/login" />)} />
    </Switch>
  </Layout>
</Router>

```

Slika 10. Prikaz definicije rutera i ruta u App.js

Izrada web aplikacije za upravljanje projektima	Verzija: <1.0>
Tehnička dokumentacija	Datum: 18.01.2022.

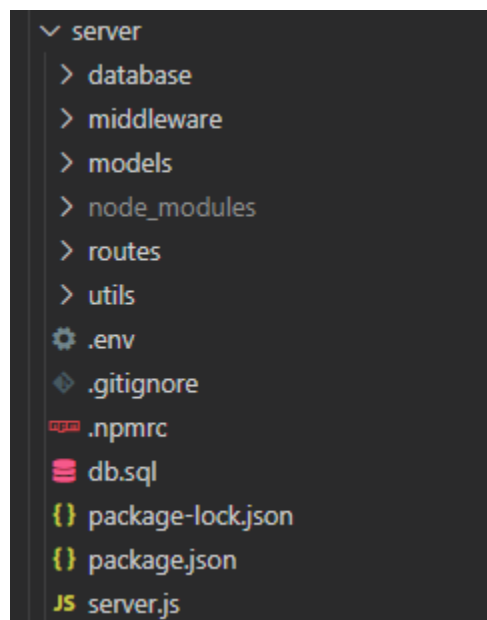
### Mapa „hocs“

U mapi „hocs“ nalazi se datoteka „Layout.js“ čiji je sadržaj prikazan na slici 11. te je njena uloga da navigacijska traga u aplikaciji uvijek bude na vrhu stranice te da se sve ostale komponente „renderaju“ ispod nje.

```
src > hocs > JS Layout.js > default
1  import React from "react";
2  import NavigationBar from '../components/NavigationBar';
3  const Layout = ({children}) => {
4      return (
5          <div>
6              <NavigationBar />
7              {children}
8          </div>
9      );
10 };
11
12 export default Layout;
```

Slika 11. Prikaz sadržaja datoteke Layout.js

U dijelu „server“ nalazi se se *backend* dio aplikacije. Slika 12. prikazuje sadržaj mape „server“.



Slika 12. Sadržaj mape "server"

Izrada web aplikacije za upravljanje projektima	Verzija: <1.0>
Tehnička dokumentacija	Datum: 18.01.2022.

### Mapa „database“

Mapa „database“ sadrži Javascript datoteke za spajanje aplikacije na bazu podataka i stvaranje tablice u bazi podataka za spremanje sesije prilikom prijavljivanja korisnika u aplikaciju. Slika 13. prikazuje datoteku **index.js** koja sadrži sve potrebne informacije pomoću kojih se aplikacija spaja na bazu podataka koja je hostana na poslužitelju Heroku.

```

8
9  const pool = new Pool({
10    user: 'xxlvjmifzfzzxf',
11    host: 'ec2-52-49-23-139.eu-west-1.compute.amazonaws.com',
12    database: 'dcnrusoio98o93',
13    password: 'a9808052bd0227925056275506dfb9d24b1525d73e01d7014adca668b7ecd208',
14    port: 5432,
15    ssl: {
16      rejectUnauthorized: false
17    }
18  });
19
20  module.exports = {
21    query: (text, params) => {
22      const start = Date.now();
23      //console.log('executing query');
24      //console.log(text);
25      //console.log(params);
26      return pool.query(text, params)
27        .then(res => {
28        //const duration = Date.now() - start;
29        //console.log('executed query', {text, params, duration, rows: res.rows});
30        return res;
31      }).catch(err => {
32        console.error(err);
33        throw err;
34      });
35    },
36    pool: pool
37  }
38
39

```

Slika 13. Index.js

Izrada web aplikacije za upravljanje projektima	Verzija: <1.0>
Tehnička dokumentacija	Datum: 18.01.2022.

### Mapa „middleware“

U mapi „middleware“ nalaze se *middleware* funkcije za validaciju podataka koje se pozivaju prije dodavanja novih podataka u bazu podataka. Slika 14. prikazuje datoteku validInfo.js, koja provjera ispravnost unesenih podataka prilikom pokušaja registracije novog korisnika u sustav.

```

app > server > middleware > JS validInfo.js > <unknown> > exports
1  module.exports = function (req, res, next) {
2      const { username, email, password, name, surname } = req.body;
3
4      // check a valid email has been entered by using a regex function
5      function validEmail(userEmail) {
6          return /^\\w+([\\.-]?\\w+)*@\\w+([\\.-]?\\w+)*(\\.\\w{2,3})+$/.test(userEmail);
7      }
8
9      if (req.path === "/signup") {
10         console.log(!email.length);
11         if (![username, email, password, name, surname].every(Boolean)) {
12             return res.json("Potrebno popuniti sva polja!");
13         } else if (!validEmail(email)) {
14             return res.json("Email nevaljan!");
15         }
16     } else if (req.path === "/login") {
17         if (![email, password].every(Boolean)) {
18             return res.json("Potrebno popuniti sva polja!");
19         } else if (!validEmail(email)) {
20             return res.json("Email nevaljan!");
21         }
22     }
23
24     next();
25 };
```

Slika 14. validInfo.js

Izrada web aplikacije za upravljanje projektima	Verzija: <1.0>
Tehnička dokumentacija	Datum: 18.01.2022.

### Mapa „models“

Mapa „models“ sadrži klase koje modeliraju konkretne entitete iz baze podataka i funkcije za dohvaćanje informacija o tim entitetima iz baze podataka, mijenjanje njihovih vrijednosti ili brisanje. Slika 15. prikazuje model zaposlenika i metode definirane za taj model.

```

app > server > models > JS Zaposlenik.js > <unknown> > Zaposlenik > fetchByUsername
1  const db = require('../database');
2
3  module.exports = class Zaposlenik {
4      constructor (korisnickoIme, lozinka, email, imeZaposlenika, prezimeZaposlenika, idUloge) {
5          this.korisnickoIme = korisnickoIme;
6          this.lozinka = lozinka;
7          this.email = email;
8          this.imeZaposlenika = imeZaposlenika;
9          this.prezimeZaposlenika = prezimeZaposlenika;
10         this.idUloge = idUloge
11     }
12
13     static async fetchByEmail(email) {
14         const results = (await db.query(
15             `SELECT * FROM Zaposlenik WHERE email = $1`,
16             [email]
17         )).rows;
18
19         return results.length == 0
20             ? new Zaposlenik()
21             : new Zaposlenik(
22                 results[0].korisnicko_ime, results[0].lozinka, results[0].email,
23                 results[0].imeZaposlenika, results[0].prezimeZaposlenika, results[0].idUloge
24             );
25     }
26     static async fetchByUsername(username) {
27         const results = (await db.query(
28             `SELECT * FROM Zaposlenik WHERE korisnickoIme = $1`,
29             [username]
30         )).rows;
31
32         return results.length == 0
33             ? new Zaposlenik()
34             : new Zaposlenik(
35                 results[0].korisnicko_ime, results[0].lozinka, results[0].email,
36                 results[0].imeZaposlenika, results[0].prezimeZaposlenika, results[0].idUloge
37             );
38     }
39 }

```

Slika 15. Model zaposlenika



Izrada web aplikacije za upravljanje projektima	Verzija: <1.0>
Tehnička dokumentacija	Datum: 18.01.2022.

### Mapa „Utils“

Mapa „Utils“ sadrži pomoćnu funkciju za generiranje JWT-a(json web token) koji se koristi za prijavljivanje u sustav i sesije. Na slici 16. prikazan je programski kod koji ostvaruje navedenu funkcionalnost.

```

app > server > utils > JS jwtGenerator.js > ...
1  const jwt = require("jsonwebtoken");
2  require("dotenv").config();
3
4  function jwtGenerator(user_id) {
5      const payload = {
6          user: {
7              id: user_id
8          }
9      };
10
11     return jwt.sign(payload, `${process.env.JWT_KEY}`, { expiresIn: "1h" });
12 }
13
14 module.exports = jwtGenerator;

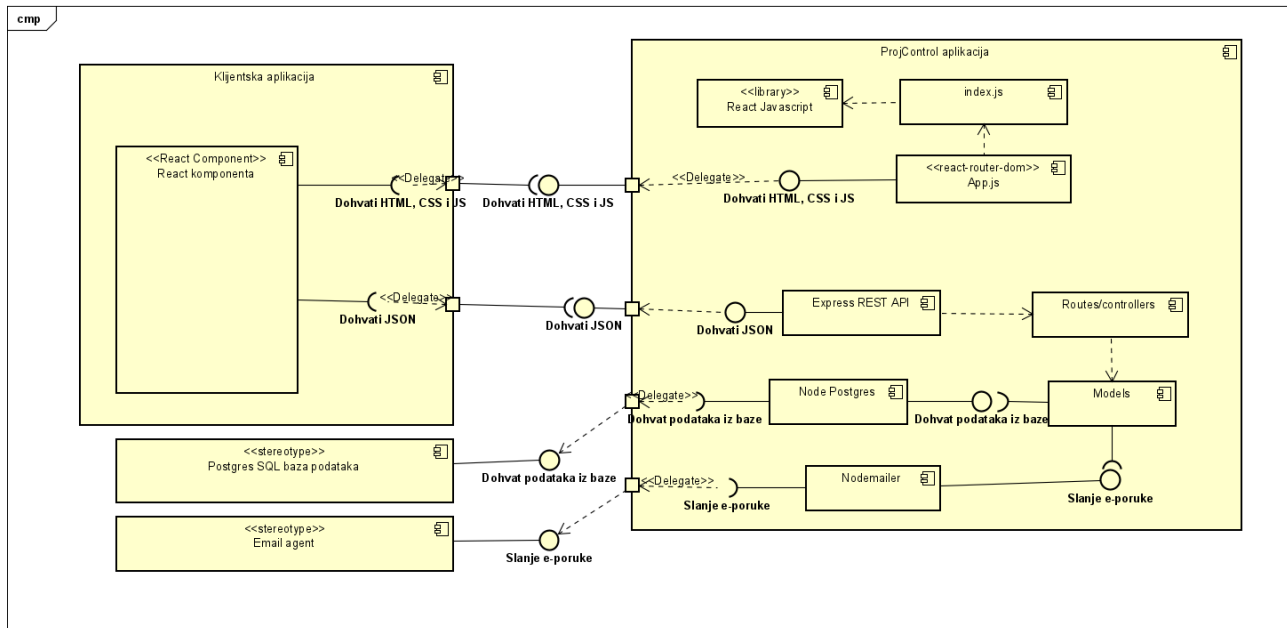
```

Slika 16. jwtGenerator.js

## 2.3 Dijagram komponenti

Na slici 17. nalazi se dijagram komponenti ove web aplikacije. Njime se vizualizira organizacija i međuovisnost između implementacijskih komponenata te odnos programske okoline prema okolini. Sučeljem za dohva CSS i JS/HTML datoteka dohvaćaju se datoteke sa frontend dijela aplikacije. Komponentom Router poslužuju se komponente stranice i React biblioteke na upit s URL-a. Dohvatom JSON podataka pristupa se Express REST API komponenti koja komunicira s backendom aplikacije. Node Postgress je kolekcija funkcija za komunikaciju Node.js-a i PostgreSQL-a. Pristigli podaci iz baze se šalju MVC arhitekturi u obliku JSON obliku. Nodemailer komponentna je koja služi za aktivaciju računa nakon uspješne registracije.

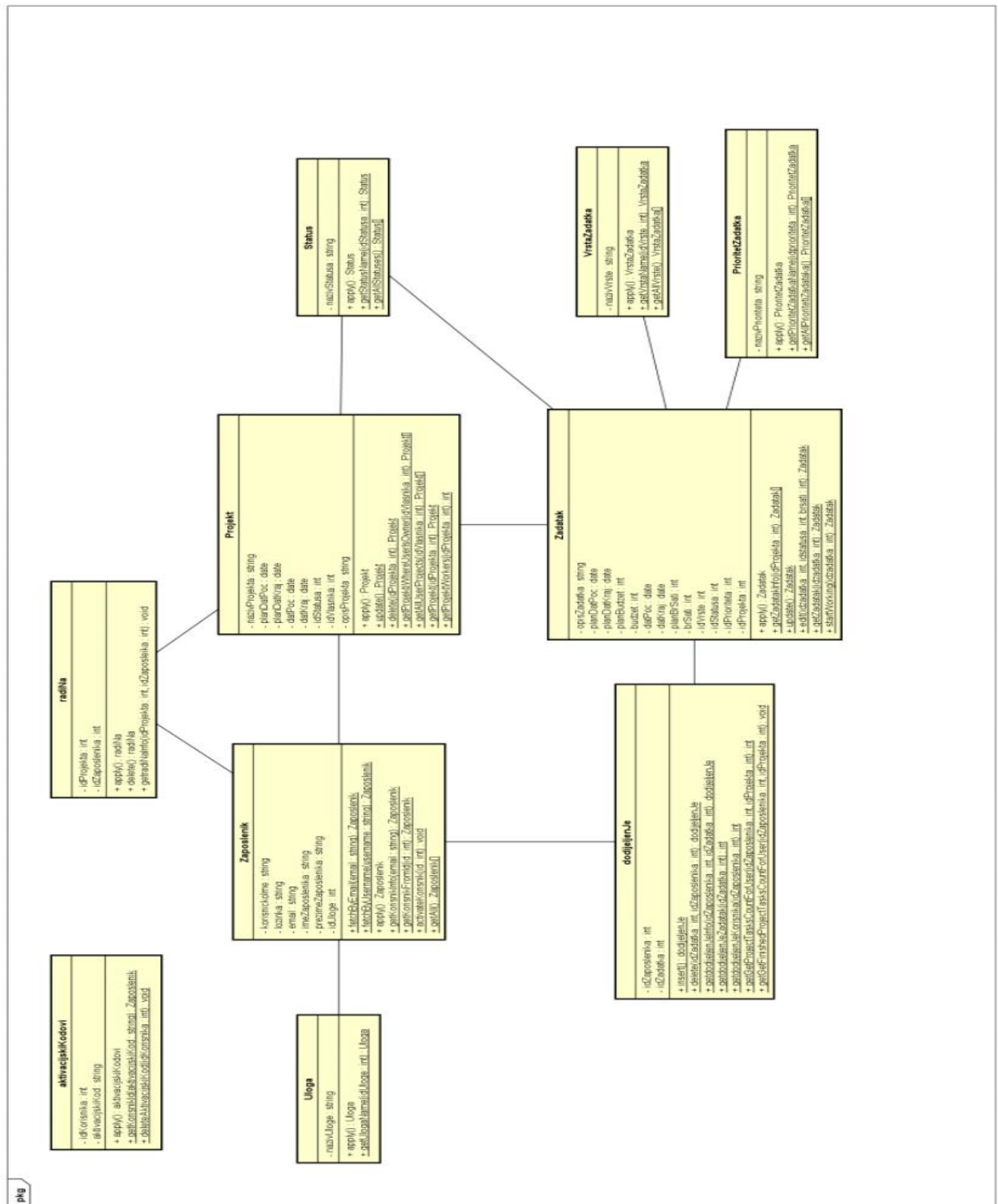
Izrada web aplikacije za upravljanje projektima	Verzija: <1.0>
Tehnička dokumentacija	Datum: 18.01.2022.



Slika 17. Dijagram komponenti

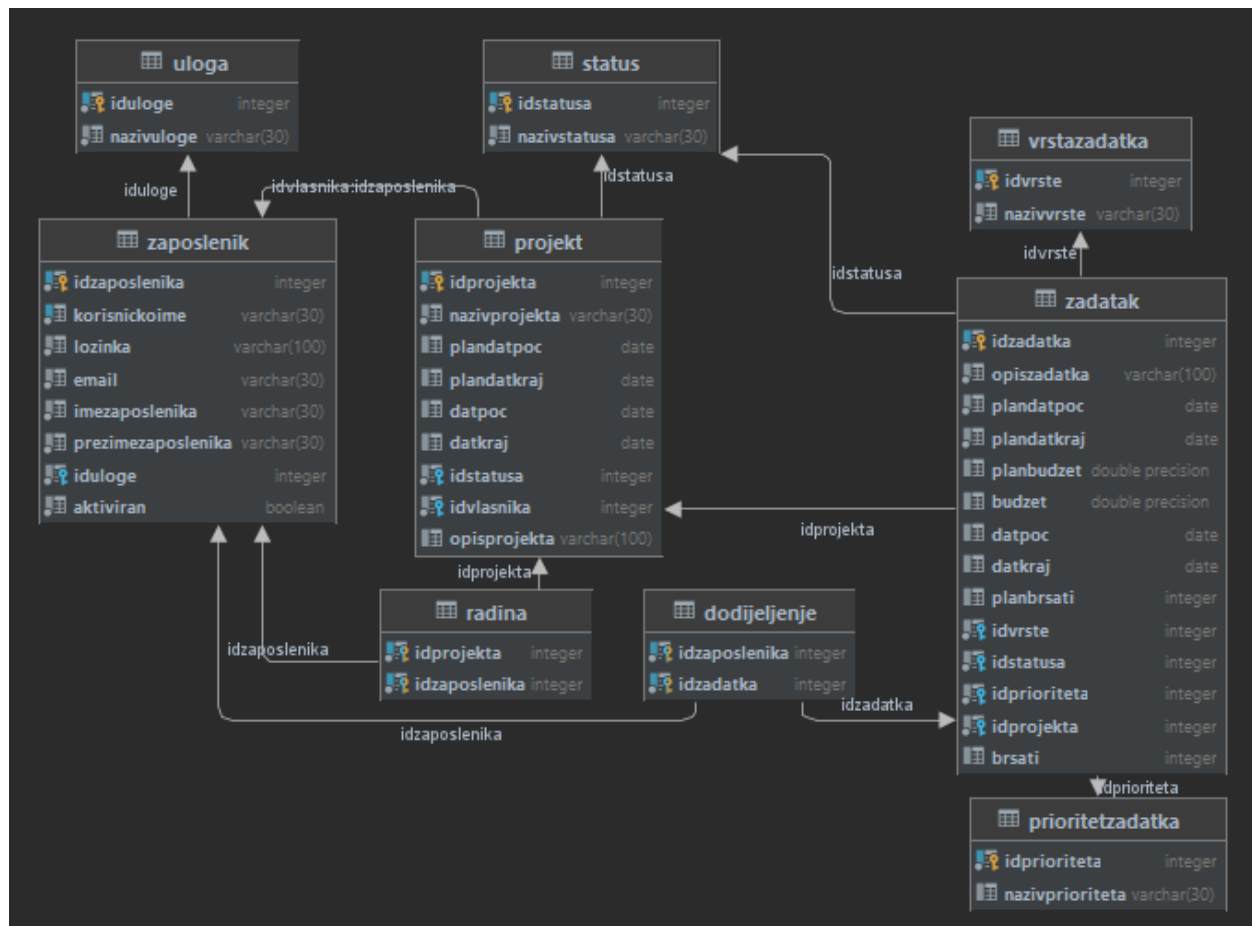
## 2.4 Dijagram klasa

Na slici 18. nalazi se dijagram klasa ove web aplikacije. Njime se vizualizira organizacija i međuovisnost između pojedinih razreda koji predstavljaju entitete baze podataka. Pri instanciranju ovih razreda, oni se inicijaliziraju pristiglim podacima iz baze nakon čega oni predstavljaju se na njima mogu pozivati opisane metode. Ako se ne inicijalizira postojećim podacima, onda instanca predstavlja novi objekt koji se tek treba dodati u bazu podataka.



Izrada web aplikacije za upravljanje projektima	Verzija: <1.0>
Tehnička dokumentacija	Datum: 18.01.2022.

## 2.5 Baza podataka



Slika 19. Dijagram baze podataka

Na slici 19. vidljiv je dijagram baze podataka.

Pri postavljanju aplikacije u produkciju, baza podataka se treba inicijalizirati. Serverska komponenta ProjControl aplikacije sadrži skripte za inicijaliziranje baze koje se mogu pokrenuti naredbama `npm run db-create` za stvaranje prazne baze podataka, ili `npm run db-seed` za stvaranje baze podataka s reprezentativnim dummy podacima.

U nastavku slijedi kratki pregled tablica i njihovih atributa. Polje obojano narančastom bojom primarni je ključ tablice.

Izrada web aplikacije za upravljanje projektima	Verzija: <1.0>
Tehnička dokumentacija	Datum: 18.01.2022.

Tablica 1. **Zaposlenik** - sadrži podatke o zaposleniku. Važno je napomenuti da se ne pohranjuje stvarna, već kriptirana lozinka.

<b>ZAPOSLENIK</b>		
<b>IDZAPOSLENIKA</b>	INTEGER	jedinstveni identifikator
KORISNICKO IME	VARCHAR(30)	korisničko ime
LOZINKA	VARCHAR(100)	kriptirana lozinka
EMAIL	VARCHAR(30)	email adresa korisnika
IMEZAPOSLENIKA	VARCHAR(30)	ime zaposlenika
PREZIMEZAPOSLENIKA	VARCHAR(30)	prezime zaposlenika
IDULOGE	INTEGER	identifikator uloge
AKTIVIRAN	BOOLEAN	status računa

Tablica 2. **Projekt** - pohranjuje podatke o projektima.

<b>PROJEKT</b>		
<b>IDPROJEKTA</b>	INTEGER	jedinstveni identifikator projekta
NAZIVPROJEKTA	VARCHAR(30)	naziv projekta
PLANDATPOC	DATE	planirani datum početka
PLANDATKRAJ	DATE	planirani datum završetka
DATPOC	DATE	stvarni datum početka
DATKRAJ	DATE	stvarni datum završetka
IDSTATUSA	INTEGER	jedinstveni identifikator statusa
IDVLASNIKA	INTEGER	jedinstveni identifikator vlasnika
OPISPROJEKTA	VARCHAR(100)	kratki opis projekta

Izrada web aplikacije za upravljanje projektima	Verzija: <1.0>
Tehnička dokumentacija	Datum: 18.01.2022.

Tablica 3. **Zadatak**- pohranjuje podatke o zadacima

<b>ZADATK</b>		
<b>IDZADATKA</b>	INTEGER	jedinstveni identifikator zadatka
OPISZADATKA	VARCHAR(100)	kratki opis zadatka
PLANDATPOC	DATE	planirani datum početka
PLANDATKRAJ	DATE	planirani datum završetka
DATPOC	DATE	stvarni datum početka
DATKRAJ	DATE	stvarni datum završetka
IDSTATUSA	INTEGER	jedinstveni identifikator statusa
PLANBUDZET	DOUBLE PRECISION	planirani budžet
BUDZET	DOUBLE PRECISION	stvarni budžet
PLANBRSATI	INTEGER	planirani broj sati
BRSATI	INTEGER	stvarni broj sati
IDPROJEKT	INTEGER	jedinstveni identifikator projekta
IDPRIORITETA	INTEGER	jedinstveni identifikator prioriteta

Tablica 4. **RADINA**- pohranjuje podatke o tome koji zaposlenik radi na kojem zadatku

<b>RADINA</b>		
<b>IDPROJEKTA</b>	INTEGER	jedinstveni identifikator projekta
<b>IDZAPOSLENIKA</b>	INTEGER	jedinstveni identifikator zaposlenika

Tablica 5. **VRSTAZADATKA** - pohranjuje podatke o vrstama zadataka

<b>VRSTAZADATKA</b>		
<b>IDVRSTE</b>	INTEGER	jedinstveni identifikator vrste
NAZIVVRSTE	VARCHAR(30)	naziv vrste zadatka

Izrada web aplikacije za upravljanje projektima	Verzija: <1.0>
Tehnička dokumentacija	Datum: 18.01.2022.

Tablica 6. **PRIORITETZADATKA** - pohranjuje podatke o prioritetima zadataka

<b>PRIORITETZADATKA</b>		
<b>IDPRIORITETA</b>	INTEGER	jedinstveni identifikator prioriteta
<b>NAZIVPRIORITETA</b>	VARCHAR(30)	naziv prioriteta

Tablica 7. **STATUS** - pohranjuje podatke o mogućim statusima zadataka ili projekata

<b>STATUS</b>		
<b>IDSTATUSA</b>	INTEGER	jedinstveni identifikator statusa
<b>NAZIVSTATUSA</b>	VARCHAR(30)	naziv statusa

Tablica 8. **ULOGE** - pohranjuje podatke o ulogama zaposlenika

<b>ULOGE</b>		
<b>IDULOGE</b>	INTEGER	jedinstveni identifikator uloge
<b>NAZIVULOGE</b>	VARCHAR(30)	naziv uloge

Tablica 9. **DODIJELJENJE** - pohranjuje podatke o tome koji zaposlenik radi na kojem projektu

<b>ULOGE</b>		
<b>IDZAPOSLENIKA</b>	INTEGER	jedinstveni identifikator zaposlenika
<b>IDPROJEKTA</b>	INTEGER	jedinstveni identifikator projekta

Izrada web aplikacije za upravljanje projektima	Verzija: <1.0>
Tehnička dokumentacija	Datum: 18.01.2022.

### 3. Upute za korištenje

U ovom poglavlju biti će opisane funkcionalnosti aplikacije, odnosno kako ih koristiti i pristupati im. Za svaku funkcionalnost bit će navedeni koraci te priložena slika iz web aplikacije.

#### 3.1 funkcionalnost – registracija

##### KORACI

1. Korisnik pritisne gumb REGISTRACIJA
2. Korisnik unosi sve potrebne podatke
3. Korisnik potvrđuje svoj račun putem unesene email adrese

The image shows a registration form with a light blue background and a dark grey rounded rectangle for the input fields. At the top, it says 'DOBRODOŠLI!' in bold. Below that, it says 'Unesite podatke potrebne za registraciju:'. The form contains four input fields: 'testUser', 'test@gmail.com', a password field with six dots, and a confirmation field with 'Testko'. At the bottom of the form is a button labeled 'REGISTRIRAJ SE'.

Slika 20. Forma za registraciju



Izrada web aplikacije za upravljanje projektima	Verzija: <1.0>
Tehnička dokumentacija	Datum: 18.01.2022.

### 3.2 funkcionalnost – prijava

#### KORACI

1. Korisnik pritisne gumb PRIJAVA
2. Korisnik unosi email i lozinku
3. Vraćanje na početni zaslon uz notifikaciju o uspješnoj prijavi

**DOBRODOŠLI NATRAG!**

**Unesite svoje podatke za prijavu:**

luka.cavalli@fer.hr

.....|

**PRIJAVA**

**Nemaš račun? REGISTRIRAJ SE**

**Zaboravljena lozinka? OBNOVI LOZINKU**

Slika 21. Forma za prijavu

Izrada web aplikacije za upravljanje projektima	Verzija: <1.0>
Tehnička dokumentacija	Datum: 18.01.2022.

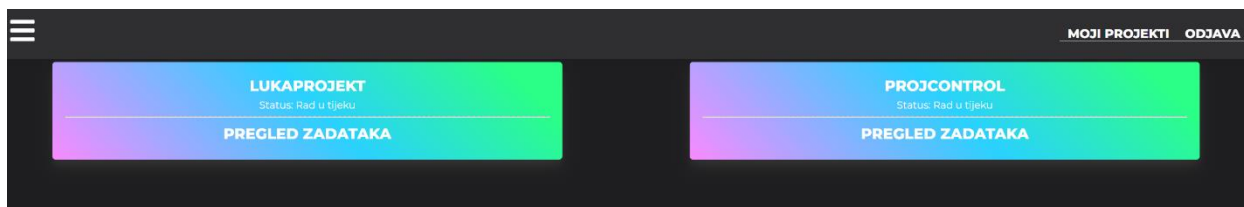
### 3.3 funkcionalnost – prikaz svih projekata

KORACI (1. verzija)

1. Ulogirani korisnik pritisne gumb MOJI PROJEKTI
2. Otvori se prikaz projekata na kojima korisnik trenutno radi

KORACI (2. verzija)

1. Ulogirani korisnik pritisne gumb za menu u gornjem lijevom kutu
2. Korisnik odabere opciju Moji projekti



Slika 22. Pregled svih projekata na kojima korisnik radi

### 3.4 funkcionalnost – dashboard projekta

KORACI

1. Ulogirani korisnik pritisne gumb MOJI PROJEKTI
2. Otvori se prikaz projekata na kojima korisnik trenutno radi
3. Korisnik odabere jedan od projekata
4. Korisnik ima sljedeće opcije:
  - a. Uređivanje podataka o projektu (pritisak na ikonu olovke)
  - b. Brisanje projekta
  - c. Promjena statusa projekta

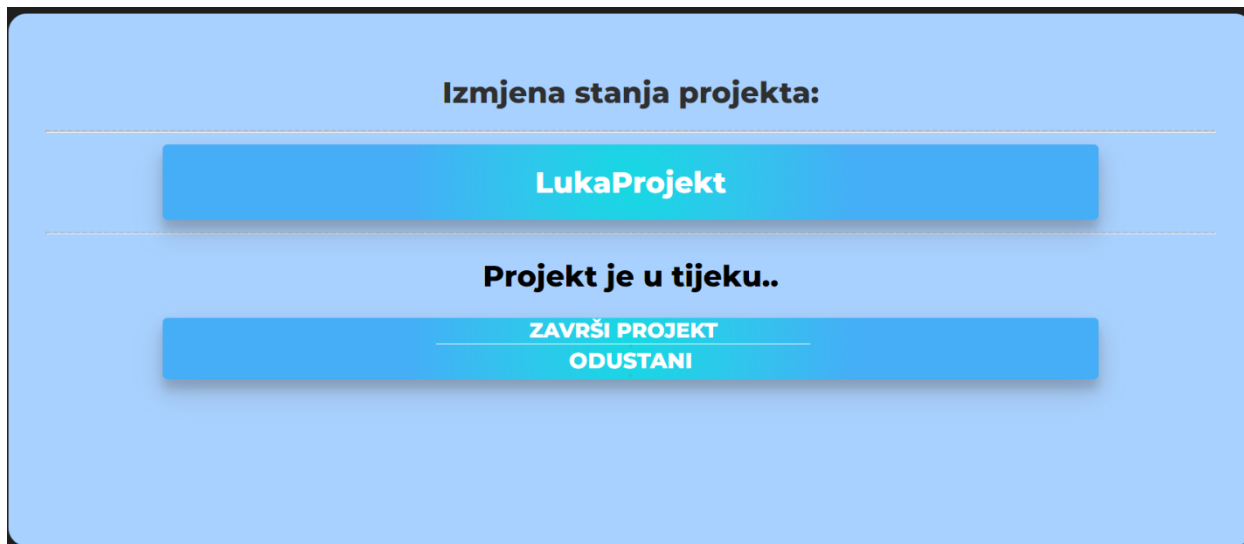
Izrada web aplikacije za upravljanje projektima	Verzija: <1.0>
Tehnička dokumentacija	Datum: 18.01.2022.



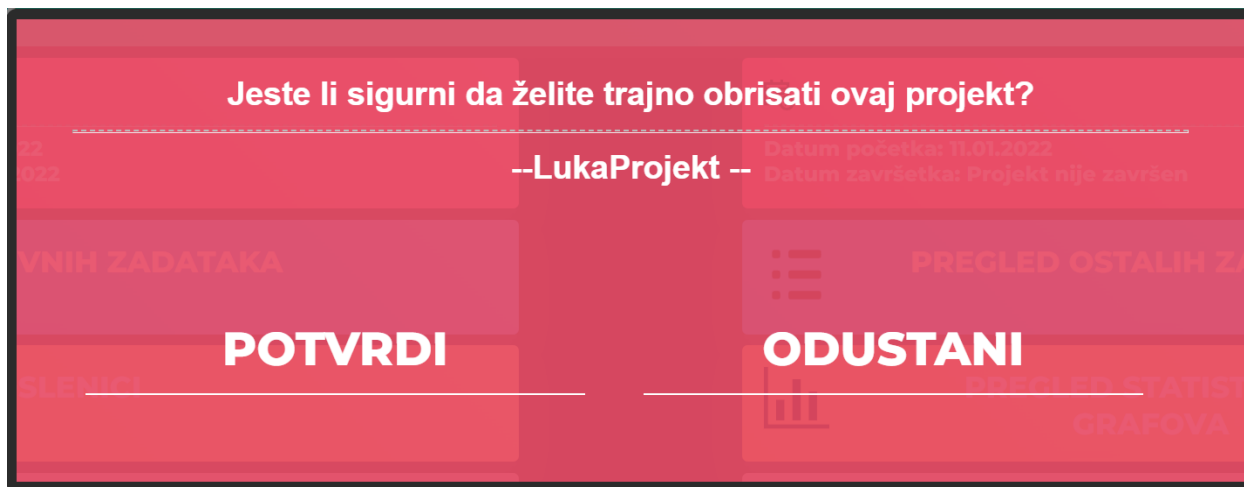
Slika 22. Dashboard projekta

Slika 23. Obrazac za uređivanje podataka o projektu

Izrada web aplikacije za upravljanje projektima	Verzija: <1.0>
Tehnička dokumentacija	Datum: 18.01.2022.



Slika 24. Promjena stanja projekta



Slika 25. Prozor za potvrdu brisanja

### 3.5 funkcionalnost – pregled aktivnih zadataka

#### KORACI

1. Na dashboardu projekta korisnik odabere PREGLED AKTIVNIH ZADATAKA

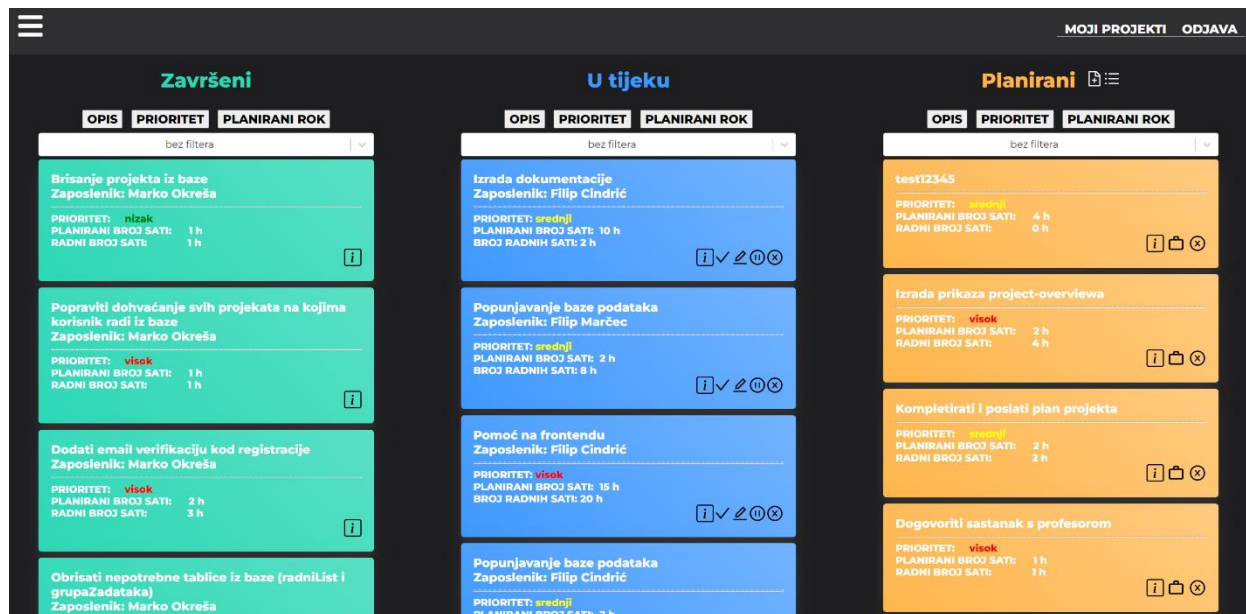
Postoje 3 vrste zadataka:

- Završeni
- U tijeku – mogu se završiti, urediti, pauzirati ili odbaciti
- Planirani – mogu se započeti ili odbaciti

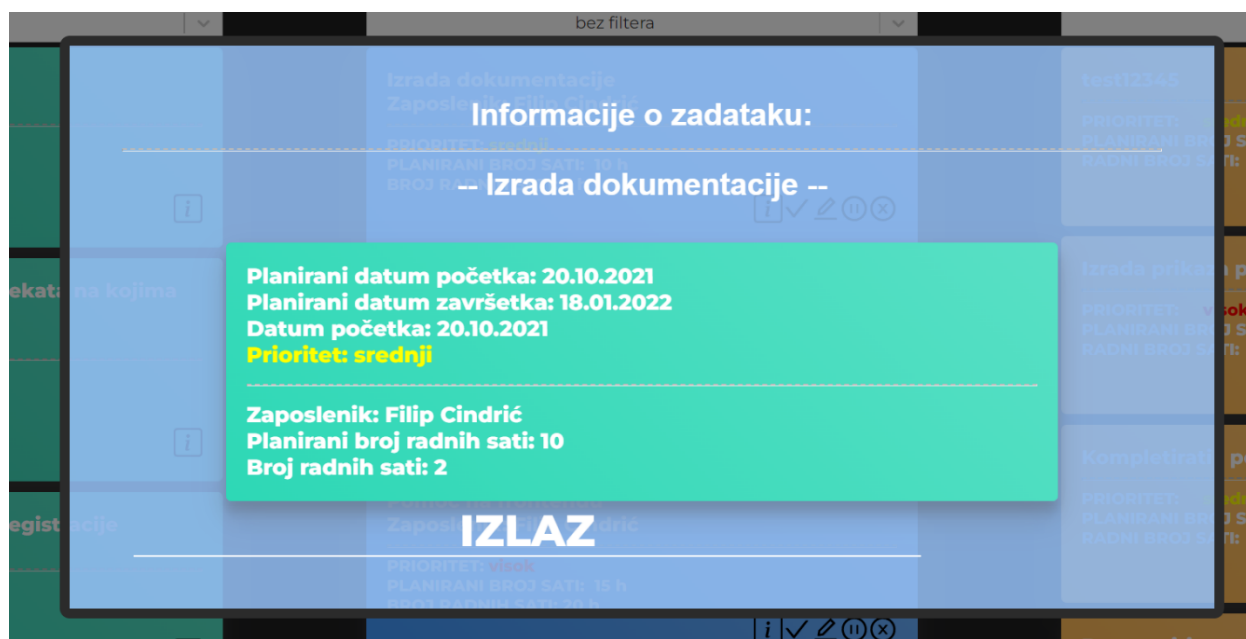
Izrada web aplikacije za upravljanje projektima	Verzija: <1.0>
Tehnička dokumentacija	Datum: 18.01.2022.

Za svaki zadatak mogu se vidjeti njegove detaljnije informacije i sortirati po opisu, prioritetu i planiranom roku.

Svi zadaci mogu se filtrirati po prioritetima.

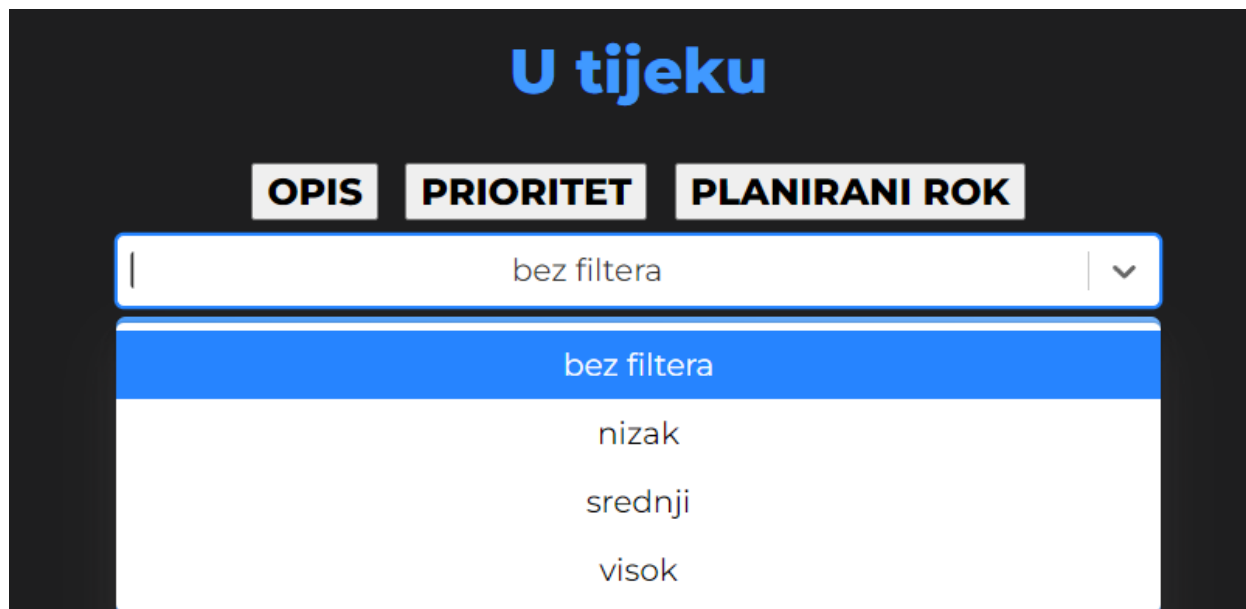


Slika 26. Pregled svih zadataka na projektu



Slika 27. Pregled informacija o zadatku

Izrada web aplikacije za upravljanje projektima	Verzija: <1.0>
Tehnička dokumentacija	Datum: 18.01.2022.

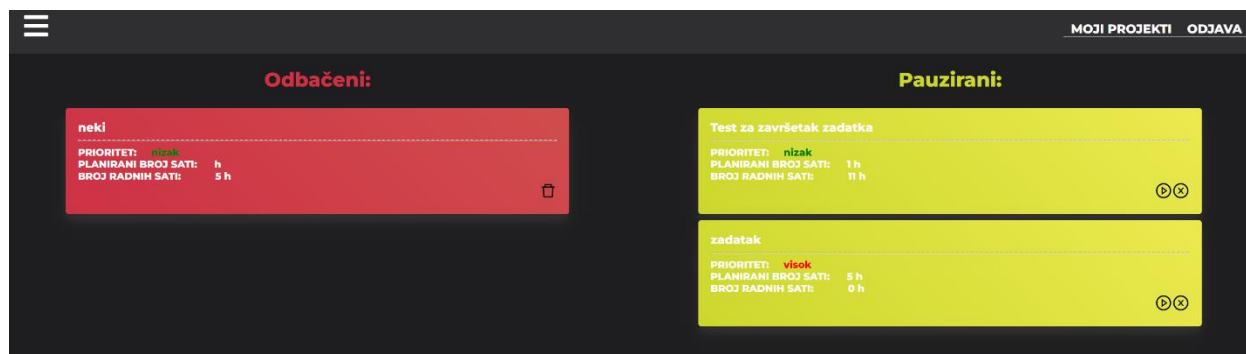


Slika 28. Prikaz opcija filtriranja

### 3.6 funkcionalnost – pregled ostalih zadataka

#### KORACI

1. Na dashboardu projekta korisnik odabere PREGLED OSTALIH ZADATAKA  
Pauzirani zadaci mogu se nastaviti ili odbaciti, a odbačeni se mogu obrisati.



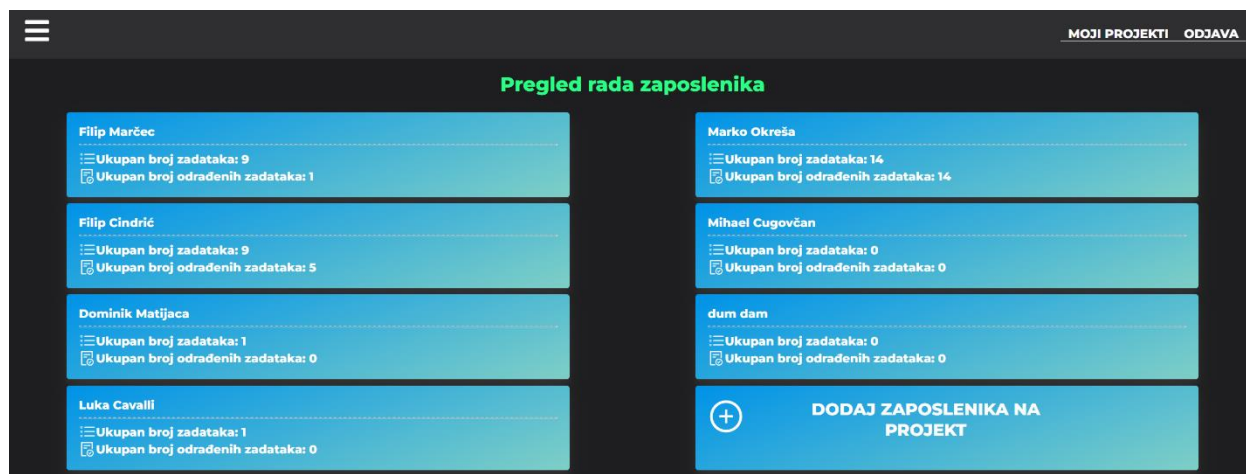
Slika 29. Prikaz “ostalih” zadataka

Izrada web aplikacije za upravljanje projektima	Verzija: <1.0>
Tehnička dokumentacija	Datum: 18.01.2022.

### 3.7 funkcionalnost – pregled zaposlenika

#### KORACI

1. Na dashboardu projekta korisnik odabere ZAPOSLENICI



Slika 30. Prikaz pregled zaposlenika

### 3.8 funkcionalnost – dodavanje novog ili uklanjanje postojećeg zaposlenika na projektu

#### KORACI

1. Na dashboardu projekta korisnik odabere ZAPOSLENICI
2. Korisnik odabere opciju DODAJ ZAPOSLENIKA NA PROJEKT

Izrada web aplikacije za upravljanje projektima	Verzija: <1.0>
Tehnička dokumentacija	Datum: 18.01.2022.

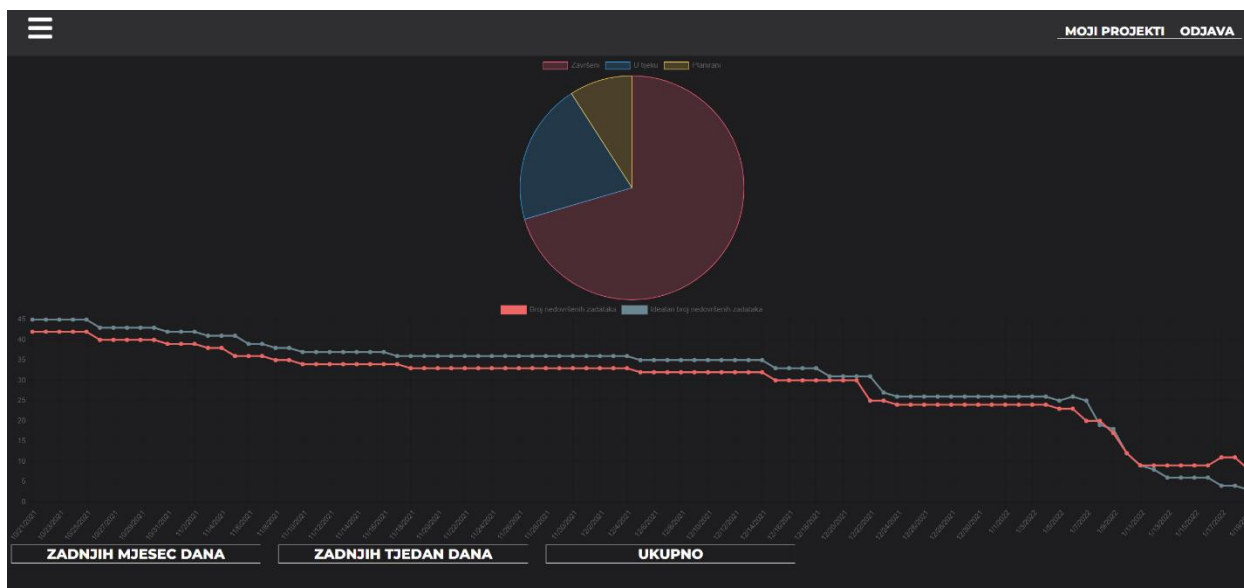
Zaposlenici na projektu:			
IME	PREZIME	KORISNIČKO IME	OBRISI
Filip	Marčec	fm52514	
Marko	Okreša	mo52775	⊖
Filip	Cindrić	fc52258	⊖
Mihael	Cugovčan	mc52657	⊖
Dominik	Matijaca	dm52456	⊖
dum	dam	dummy	⊖
Luka	Cavalli	LukaCava	⊖
Ostali korisnici:			
IME	PREZIME	KORISNIČKO IME	DODAJ
Filip	Marčec	mrkl	⊕
Bla	Test	bla123	⊕

Slika 31. Prikaz uklanjanje korisnika s projekta

### 3.9 funkcionalnost – pregled grafova

#### KORACI

1. Na dashboardu projekta korisnik odabere PREGLED STATISTIČKIH GRAFOVA  
(može se izabrati jedan od tri vremenskih intervala za burndown chart)



Slika 32. Prikaz grafova

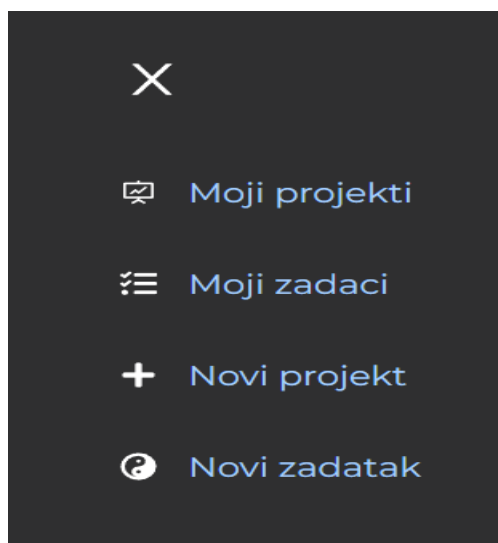


Izrada web aplikacije za upravljanje projektima	Verzija: <1.0>
Tehnička dokumentacija	Datum: 18.01.2022.

### 3.10 funkcionalnost – dodavanje novog projekta

#### KORACI

1. Ulogirani korisnik pritisne gumb za menu u gornjem lijevom kutu
2. Korisnik odabere opciju Novi projekt
3. Korisnik unese sve potrebne podatke



Slika 33. SideBar opcije

 A light blue header bar contains the title 'Dodavanje novog projekta...'. Below the title, the instruction 'Unesite sve potrebne podatke:' is displayed. The form area has a dark background and contains: a text input field for 'Upišite naziv projekta'; two date pickers for 'Planirani datum početka: mm/dd/yyyy' and 'Planirani datum završetka: mm/dd/yyyy'; a large text area for 'Unesite kratki opis projekta...'; and a prominent 'DODAJ PROJEKT' button at the bottom.

Slika 34. Obrazac za dodavanje novog projekta


Izrada web aplikacije za upravljanje projektima	Verzija: <1.0>
Tehnička dokumentacija	Datum: 18.01.2022.

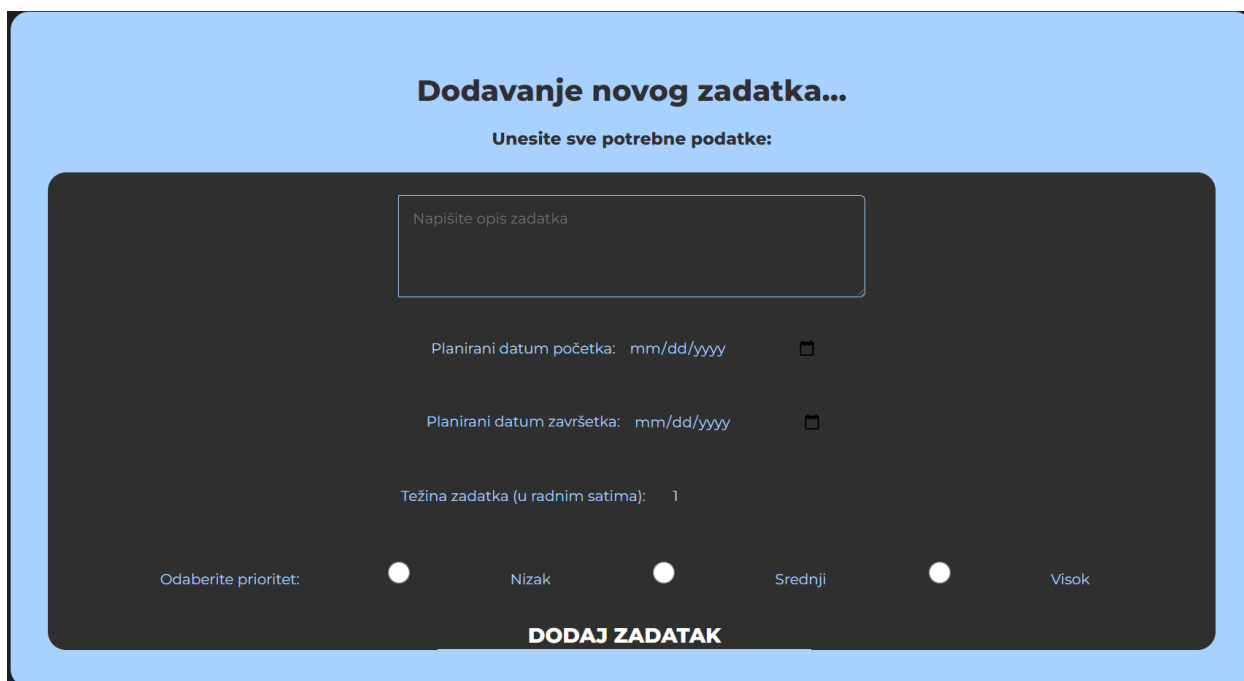
### 3.11 funkcionalnost – dodavanje novog zadatka

#### KORACI (1. verzija)

1. Ulogirani korisnik pritisne gumb za menu u gornjem lijevom kutu
2. Korisnik odabere opciju Novi zadatak
3. Korisnik unese sve potrebne podatke

#### KORACI (2. verzija)

1. Na dashboardu projekta korisnik odabere PREGLED AKTIVNIH ZADATAKA
2. Korisnik pritisne gumb s ikonicom 
3. Korisnik unese sve potrebne podatke



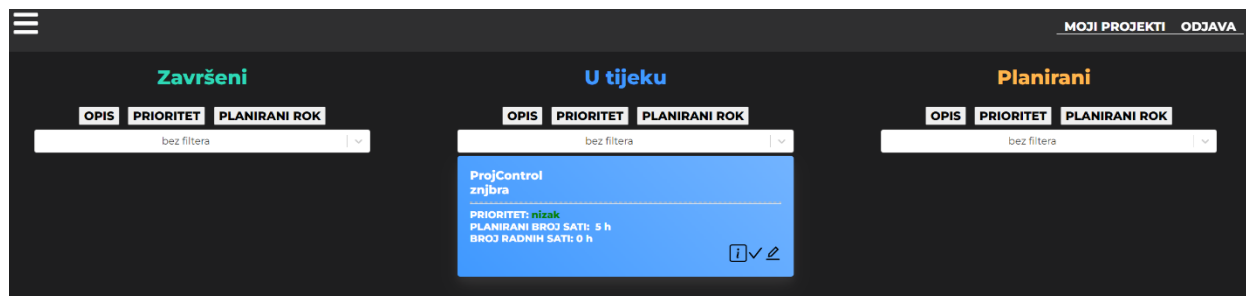
Slika 35. Obrazac za dodavanje zadataka

### 3.12 funkcionalnost – pregled zadataka na trenutnom projektu

#### KORACI

1. Ulogirani korisnik pritisne gumb za menu u gornjem lijevom kutu
2. Korisnik odabere opciju Moji zadaci

Izrada web aplikacije za upravljanje projektima	Verzija: <1.0>
Tehnička dokumentacija	Datum: 18.01.2022.



Slika 36. Prikaz "Moji zadaci"

## 4.Literatura

- [1] ReactJS dokumentacija, <https://reactjs.org/docs/getting-started.html> (18.01.2022.)
- [2] ChartJS dokumentacija, <https://github.com/chartjs/Chart.js/tree/master/docs> (18.01.2022.)
- [3] NodeJS dokumentacija, <https://nodejs.org/en/docs/> (18.01.2022.)
- [4] ExpressJS dokumentacija, <https://expressjs.com/en/5x/api.html> (18.01.2022.)
- [5] PostgreSQL dokumentacija, <https://www.postgresql.org/docs/> (18.01.2022.)
- [6] Node-postgress dokumentacija, <https://node-postgres.com/> (18.01.2022.)
- [7] Node-mailer, dokumentacija, <https://nodemailer.com/about/> (18.01.2022.)