

**Blog - Blogzilla**

IT354 - Veb sistemi 1

Projektna dokumentacija

19.1.2025.

Student: Marko Simonović 5349

Profesor: Nikola Dimitrijević

**Sadržaj**

[1. Uvod 3](#_Toc193295670)

[ Opis teme projekta 3](#_Toc193295671)

[ Cilj projekta i funkcionalnosti koje će biti implementirane. 3](#_Toc193295672)

[ Kratak pregled tehnologija i alata koji će biti korišćeni. 3](#_Toc193295673)

[2. Planiranje i dizajn sistema 3](#_Toc193295674)

[ Definicija zahteva 3](#_Toc193295675)

[ Use-case dijagram 4](#_Toc193295676)

[3. Razvoj korisničkog interfejsa (UI) 5](#_Toc193295677)

[ Izbor tehnologije i razlog za izbor 5](#_Toc193295678)

[ Pregled ključnih stranica 5](#_Toc193295679)

[4. Implementacija funkcionalnosti 9](#_Toc193295680)

[ Korisničke funkcionalnosti 9](#_Toc193295681)

[ Administratorske funkcionalnosti 11](#_Toc193295682)

[5. Simulacija backend-a 12](#_Toc193295683)

[ Korišćenje json-servera 12](#_Toc193295684)

[ Primer JSON datoteke 13](#_Toc193295685)

[ Povezivanje frontenda i backend-a 13](#_Toc193295686)

[6. Zaključak 16](#_Toc193295687)

[ Pregled postignutih rezultata 16](#_Toc193295688)

[ Izazovi i prepreke tokom rada na projektu 16](#_Toc193295689)

[ Mogućnosti za buduća unapređenja aplikacije 17](#_Toc193295690)

[7. Prilozi 17](#_Toc193295691)

# Uvod

## Opis teme projekta

Ovaj projekat predstavlja jednostavan blog sistem razvijen korišćenjem React-a za frontend i JSON Server-a kao backend rešenja. Blog omogućava korisnicima da pregledaju objave, dok registrovani korisnici imaju dodatne mogućnosti kao što su komentarisanje i kreiranje sopstvenih blogova. Administratori imaju punu kontrolu nad sadržajem platforme.

## Cilj projekta i funkcionalnosti koje će biti implementirane.

Cilj projekta je da omogući jednostavno upravljanje blogovima i interakciju između korisnika kroz komentare. Implementirane funkcionalnosti uključuju:

* Pregled svih blogova (dostupno svim korisnicima).
* Registracija i prijava korisnika.
* Kreiranje, uređivanje i brisanje sopstvenih blogova (samo za registrovane korisnike).
* Dodavanje komentara na blogove (samo za registrovane korisnike).
* Administratorske funkcije:
  + Uređivanje i brisanje svih blogova.
  + Brisanje komentara.

## Kratak pregled tehnologija i alata koji će biti korišćeni.

Za razvoj aplikacije korišćene su sledeće tehnologije i alati:

* Frontend: React + Bootstrap (za UI/UX dizajn).
* Backend: JSON Server (za simulaciju REST API-ja).
* Autentifikacija: Ručno implementirana provera korisnika (bez pravog backend-a).
* Menadžment stanja: useState i useEffect React hook-ovi.

# Planiranje i dizajn sistema

## Definicija zahteva

Funkcionalni zahtevi

Sistem se sastoji od tri tipa korisnika: neregistrovani korisnici, registrovani korisnici i administratori. Funkcionalnosti su raspoređene na sledeći način:

Nelogovani korisnik može:

* Pregledati listu blogova.
* Pregledati pojedinačne blogove i njihove komentare.

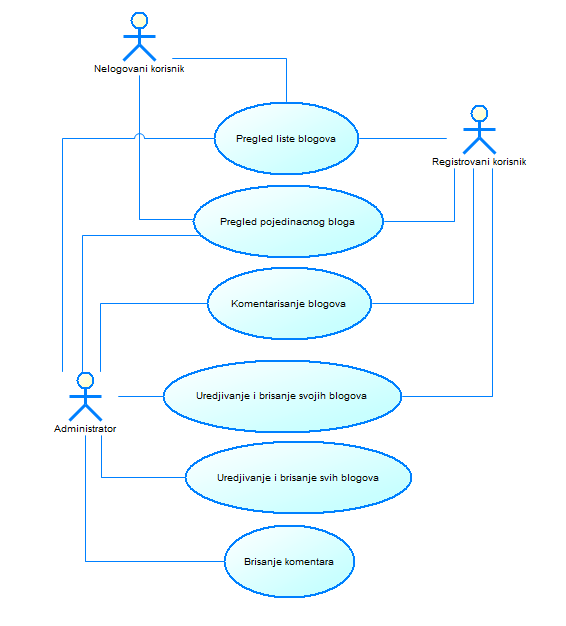
Registrovani korisnik može:

* Sve što i nelogovani korisnik.
* Kreirati nove blogove.
* Uređivati i brisati svoje blogove.
* Dodavati komentare na blogove.

Administrator može:

* Sve što i registrovani korisnik.
* Uređivati i brisati sve blogove.
* Brisati komentare svih korisnika.

## Use-case dijagram



Prikaz aktera i slučajeva korišćenja

Use-case dijagram prikazuje interakciju između korisnika i sistema.

Akteri su:

* Neregistrovani korisnik
* Registrovani korisnik
* Administrator

Dijagram vizuelno prikazuje kako različiti korisnici koriste sistem. Neregistrovani korisnici imaju ograničene mogućnosti (samo pregled blogova), dok registrovani korisnici mogu dodavati i uređivati svoje sadržaje. Administratori imaju proširene povlastice i mogu kontrolisati celokupan sadržaj bloga.

# Razvoj korisničkog interfejsa (UI)

Za stilizaciju interfejsa u ovom projektu korišćen je Bootstrap, popularni CSS framework koji omogućava brzo i efikasno kreiranje responzivnog dizajna.

## Izbor tehnologije i razlog za izbor

Korišćenje Bootstrap-a omogućilo je:

* Brzu implementaciju dizajna – gotove klase za layout, tipografiju i komponente.
* Responzivnost – automatsko prilagođavanje sadržaja različitim uređajima bez potrebe za dodatnim medija upitima.
* Jednostavno prilagođavanje – mogućnost modifikacije Bootstrap klasa ili dodavanja prilagođenih CSS stilova.
* Laku integraciju sa React-om – bez potrebe za dodatnim alatima poput Styled Components ili Tailwind-a.

Osim Bootstrap-a, korišćeni su i prilagođeni CSS stilovi u pojedinim delovima aplikacije kako bi se dodatno personalizovao izgled interfejsa.

## Pregled ključnih stranica

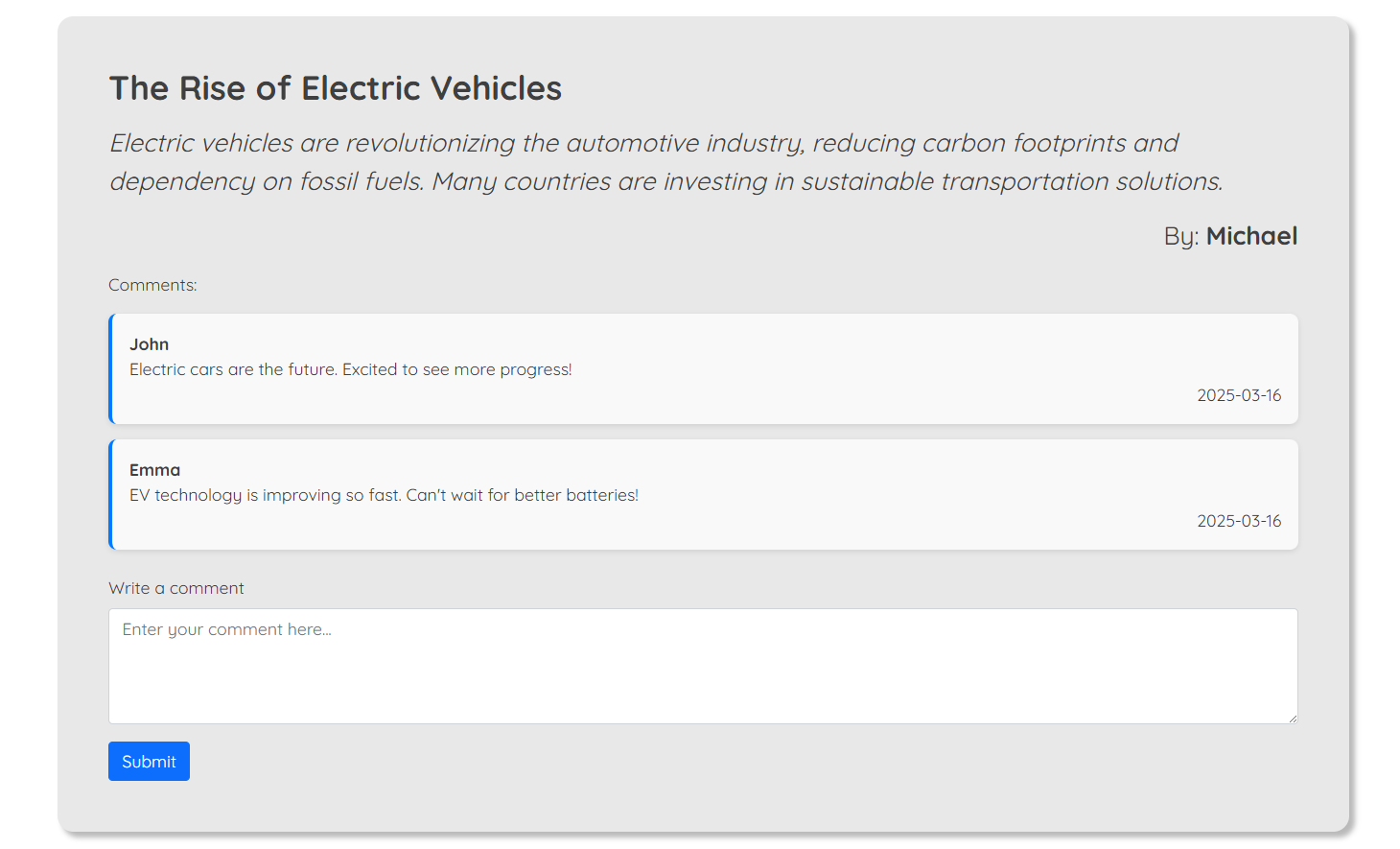
**Početna stranica** prikazuje sve blogove u sažetom obliku, gde svaki blog sadrži naslov, kratak opis i dugme "View More" za detaljan prikaz. Korisnici mogu pretraživati blogove pomoću search polja, koje filtrira rezultate prema naslovu (title) i sadržaju (content). Dizajn stranice je jednostavan i pregledan, omogućavajući korisnicima brz i lak pristup dostupnim blogovima.

A screenshot of a computer

AI-generated content may be incorrect.

Na detaljnoj stranici prikazuje se kompletan sadržaj odabranog bloga, uključujući:

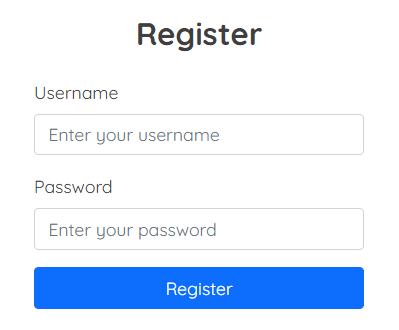
* Naslov i celokupan tekst bloga,
* Ime autora bloga,
* Lista komentara ostavljenih od strane korisnika,
* Polje za unos novog komentara, dostupno samo registrovanim korisnicima.



Login stranica – korisnik unosi korisničko ime (username) i lozinku (password) kako bi se prijavio u sistem.

Register stranica – novi korisnici mogu kreirati nalog unosom korisničkog imena i lozinke.

A screenshot of a login box

AI-generated content may be incorrect. 

Stranica za kreiranje novog bloga. Stranica omogućava registrovanim korisnicima i administratorima da kreiraju novi blog unosom sledećih podataka:

* Naslov bloga
* Sadržaj bloga

Nakon popunjavanja forme, korisnik može sačuvati blog, koji će potom biti prikazan na početnoj stranici. Stranica sadrži jednostavan i pregledan dizajn, a validacija unosa sprečava kreiranje bloga bez naslova ili sadržaja.

A screenshot of a blog

AI-generated content may be incorrect.

Administrator ima proširene mogućnosti upravljanja sadržajem bloga, uključujući:

* Brisanje blogova – može obrisati bilo koji blog na platformi.
* Uređivanje blogova – može izmeniti naslov i sadržaj bilo kog bloga.
* Brisanje komentara – može ukloniti komentare svih korisnika.

Ove funkcije omogućavaju administratoru potpunu kontrolu nad sadržajem i interakcijama na blogu, osiguravajući moderaciju i održavanje kvaliteta objava.

A screenshot of a computer

AI-generated content may be incorrect.

# Implementacija funkcionalnosti

## Korisničke funkcionalnosti

Funkcija omogućava prijavu postojećeg korisnika tako što proverava korisničko ime i lozinku. Ako su podaci tačni, korisnik se čuva u localStorage i preusmerava na početnu stranicu. U suprotnom, prikazuje se upozorenje o grešci.

A screen shot of a computer

AI-generated content may be incorrect.

Funkcija omogućava kreiranje novog bloga slanjem podataka na server. Ako je blog uspešno kreiran, korisnik se preusmerava na stranicu sa detaljima bloga, a u suprotnom se vraća na početnu stranicu.

A computer screen with text on it

AI-generated content may be incorrect.

Funkcija za učitavanje i pretragu blogova

Ovaj kod omogućava učitavanje blogova sa API-ja i filtriranje na osnovu pretrage:

* Učitavanje blogova – koristi useEffect kako bi učitao blogove prilikom inicijalnog renderovanja stranice i čuva ih u blogs i allBlogs stanjima.
* Filtriranje blogova – funkcija research pretražuje blogove po naslovu i sadržaju, filtrirajući ih na osnovu unetog pretraživačkog stringa.

A screen shot of a computer program

AI-generated content may be incorrect.

## Administratorske funkcionalnosti

Funkcija za brisanje bloga i komentara

* Brisanje bloga – funkcija handleDeleteBlog briše blog sa servera i preusmerava korisnika na početnu stranicu ako je brisanje uspešno.
* Brisanje komentara – funkcija handleDeleteComment briše konkretan komentar i aktivira ponovni render stranice ako je brisanje uspešno.

A computer screen with text on it

AI-generated content may be incorrect.

Funkcija omogućava ažuriranje bloga slanjem izmenjenih podataka na server. Ako je ažuriranje uspešno, korisnik se preusmerava na stranicu sa detaljima bloga, a u suprotnom na početnu stranicu.

A computer screen shot of a program code

AI-generated content may be incorrect.

# Simulacija backend-a

## Korišćenje json-servera

json-server je alat koji omogućava brzo postavljanje lažnog REST API servera koristeći JSON datoteku kao izvor podataka. Idealno je za razvoj i testiranje aplikacija koje zahtevaju backend bez potrebe za stvaranjem pravog servera.

Postavljanje json-servera

Instalacija:

*npm install -g json-server*

Pokretanje servera:

*json-server --watch db.json --port 3000*

Ovaj komandni red pokreće server na portu 3000 i nadgleda izmene u db.json datoteci.

## Primer JSON datoteke

A screen shot of a computer program

AI-generated content may be incorrect.

## Povezivanje frontenda i backend-a

Funkcija useFetchGet je asinhrona funkcija koja šalje GET zahtev na zadatu URL adresu, vraća JSON odgovor i u slučaju greške loguje grešku i vraća null. Koristi se za preuzimanje podataka sa servera.

A screen shot of a computer program

AI-generated content may be incorrect.

Funkcija useFetchPost je asinhrona funkcija koja šalje POST zahtev na zadatu URL adresu sa datim telom (body). Ako je zahtev uspešan, vraća JSON odgovor, a u suprotnom loguje grešku i vraća null.

A screen shot of a computer

AI-generated content may be incorrect.

Funkcija useFetchPut je asinhrona funkcija koja šalje PUT zahtev na zadatu URL adresu sa datim telom (body). Ako je zahtev uspešan, vraća JSON odgovor, a u suprotnom loguje grešku i vraća null.

A screen shot of a computer program

AI-generated content may be incorrect.

Funkcija useFetchDelete je asinhrona funkcija koja šalje DELETE zahtev na zadatu URL adresu. Ako je zahtev uspešan, vraća true, a u slučaju greške ili neuspeha vraća false i loguje grešku.

A screen shot of a computer program

AI-generated content may be incorrect.

# Zaključak

## Pregled postignutih rezultata

Projekat blog aplikacije je uspešno implementiran koristeći React za frontend i json-server za backend. Aplikacija omogućava korisnicima da pregledaju blogove, dok registrovani korisnici mogu komentarisati, kreirati i uređivati svoje blogove. Administratori imaju dodatnu funkcionalnost za upravljanje blogovima i komentarima. Sve funkcionalnosti su postavljene uz korišćenje REST API zahteva za interakciju sa serverom.

## Izazovi i prepreke tokom rada na projektu

Tokom rada na projektu, najveći izazov bio je pravilno implementirati upravljanje korisničkim sesijama, kao i osigurati tačnu interakciju između frontenda i json-servera. Takođe, testiranje svih funkcionalnosti u simuliranom okruženju bila je prepreka zbog ograničenih mogućnosti json-servera u odnosu na pravi server.

## Mogućnosti za buduća unapređenja aplikacije

Aplikacija može biti unapređena dodavanjem funkcionalnosti za autentifikaciju sa JWT tokenima, što bi omogućilo sigurnije upravljanje korisničkim sesijama. Takođe, proširenje mogućnosti za filtriranje i sortiranje blogova, kao i dodavanje podrške za upload slika, moglo bi poboljšati korisničko iskustvo. U budućnosti bi aplikacija mogla biti migrirana na pravi backend sistem sa bazom podataka za veću skalabilnost i sigurnost.

# Prilozi

https://github.com/MarkoSimonovic03/Blogzilla