

# Aplikacija za prediktivni nadzor naprednih zgrada

---

**Brkić, Vinko**

**Undergraduate thesis / Završni rad**

**2024**

*Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj:* **University of Zagreb, Faculty of Electrical Engineering and Computing / Sveučilište u Zagrebu, Fakultet elektrotehnike i računarstva**

*Permanent link / Trajna poveznica:* <https://um.nsk.hr/um:nbn:hr:168:300834>

*Rights / Prava:* [In copyright](#)/[Zaštićeno autorskim pravom.](#)

*Download date / Datum preuzimanja:* **2025-03-20**



*Repository / Repozitorij:*

[FER Repository - University of Zagreb Faculty of Electrical Engineering and Computing repository](#)



SVEUČILIŠTE U ZAGREBU  
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

ZAVRŠNI RAD br. 1575

**APLIKACIJA ZA PREDIKTIVNI NADZOR NAPREDNIH  
ZGRADA**

Vinko Brkić

Zagreb, lipanj 2024.

SVEUČILIŠTE U ZAGREBU  
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

ZAVRŠNI RAD br. 1575

**APLIKACIJA ZA PREDIKTIVNI NADZOR NAPREDNIH  
ZGRADA**

Vinko Brkić

Zagreb, lipanj 2024.

## ZAVRŠNI ZADATAK br. 1575

Pristupnik: **Vinko Brkić (0036539333)**  
Studij: Elektrotehnika i informacijska tehnologija i Računarstvo  
Modul: Računarstvo  
Mentor: izv. prof. dr. sc. Vinko Lešić

Zadatak: **Aplikacija za prediktivni nadzor naprednih zgrada**

### Opis zadatka:

Zgrade su jedan od najvećih svjetskih proizvođača gdje se energetska bilanca unaprjeđuje korištenjem obnovljivih izvora energije. Instaliranjem fotonaponskih panela na krovove rezidencijalnih zgrada, one postaju distribuirana solarna elektrana. U radu je potrebno dohvatiti podatke o distribuiranoj proizvodnji i potrošnji energije u zgradama pohranjenoj u PostgreSQL središnjoj bazi podataka te načiniti aplikaciju za vizualizaciju podataka u internetskom pregledniku korištenjem programskog okvira Native. Pri tome se prikazuju povijesni trendovi u zgradi kao što su potrošnja energije različitih uređaja, ukupna proizvodnja, pohrana i potrošnja te vanjski vremenski uvjeti. Aplikaciju je potrebno proširiti i za prikaz prediktivne analitike, odnosno buduće procjene varijabli dobivene linearnom regresijom.

Rok za predaju rada: 14. lipnja 2024.

*Želim se zahvaliti svom mentoru, izv. prof. dr. sc. Vinku Lešiću, na nesebičnom vođenju, korisnim savjetima i kontinuiranoj podršci tijekom cijelog procesa izrade rada. Vaša stručnost i pristupačnost bili su mi od velike pomoći. Zahvaljujem se i svim profesorima i asistentima s Fakulteta elektrotehnike i računarstva na prenesenom znanju i vještinama koje su bile ključne za realizaciju ovog projekta. Velika hvala i mojim kolegama i prijateljima koji su mi pružali podršku i ohrabrenje, te dijelili svoje znanje i iskustva. Posebnu zahvalnost upućujem svojoj obitelji na njihovoj podršci, razumijevanju i strpljenju. Bez njihove podrške i vjere u mene, ovaj rad ne bi bio moguć. Na kraju, hvala svima koji su na bilo koji način doprinijeli uspjehu ovog završnog rada.*

# Sadržaj

<b>1. Uvod</b>	<b>3</b>
<b>2. Opis razvijenog proizvoda</b>	<b>4</b>
2.1. Statistika	4
2.2. Grafički prikaz	6
2.3. Profil	11
<b>3. Linearna regresija</b>	<b>12</b>
3.1. Programski jezik R	12
3.2. Strojno učenje	13
3.2.1. Nadzirano učenje	14
3.2.2. Nenadzirano učenje	14
3.2.3. Pojačano učenje	15
3.3. Linearna regresija unutar aplikacije	15
<b>4. Tehničke značajke</b>	<b>20</b>
<b>5. Metodološki put</b>	<b>22</b>
5.1. Backend server	22
5.1.1. Rute definirane u Express.js	22
5.2. Frontend	23
5.3. Blokovska shema arhitekture	23
5.3.1. Klijentska strana (React)	23
5.3.2. Backend server (Node.js, Express, PostgreSQL)	24
5.3.3. Baza podataka	25
5.3.4. Komunikacijski put	27

<b>6. Rezultati i rasprava</b> . . . . .	<b>30</b>
6.1. Predikcija spremnika ( <i>spremnik</i> ) . . . . .	30
6.2. Predikcija potrošnje ( <i>potrosnja</i> ) . . . . .	31
6.3. Predikcija proizvodnje ( <i>proizvodnja</i> ) . . . . .	31
6.4. Ukupne metričke performanse . . . . .	31
<b>7. Zaključak</b> . . . . .	<b>34</b>
<b>Literatura</b> . . . . .	<b>35</b>
<b>Sažetak</b> . . . . .	<b>36</b>
<b>Abstract</b> . . . . .	<b>37</b>

# 1. Uvod

U 21. stoljeću, dok se svijet suočava s rastućim izazovima vezanim uz klimatske promjene i održivost, solarna energija se izdvaja kao ključna snaga budućnosti. Ovaj obnovljivi izvor energije ne samo da pruža alternativu fosilnim gorivima, već predstavlja temeljnu komponentu održivog razvoja. U tom kontekstu, integracija solarnog načina napajanja postaje ne samo trend, već nužnost za stvaranje ekološki osviještenih te energetske učinkovitih domova i zgrada. Također, solarna energija ne nudi samo alternativu tradicionalnim izvorima, već otvara vrata za novu eru energetike - era u kojoj svaki kvadratni metar krova može postati generator obnovljive energije. Baš iz tog razloga pametne zgrade postaju laboratoriji za inovacije što je ekološki pozitivno i ekonomski isplativo. Pametno korištenje solarnih resursa može značajno smanjiti troškove energije, osigurati energetske neovisnost, te smanjiti emisije štetnih plinova. Pametne zgrade i kućanstva postaju oslonac održive infrastrukture, ali njihova efikasnost i ekonomičnost zahtijevaju mnogo više od pasivnog prikupljanja sunčeve energije. Potrebno je predviđati potrebe za energijom, optimizirati potrošnju, te integrirati sustave skladištenja energije. U sklopu rada istražuje se važnost solarnog napajanja u kontekstu pametnih zgrada, s naglaskom na ulogu aplikacija za prediktivni nadzor omogućavajući sustavima solarnog napajanja da postignu svoj puni potencijal. Kroz prediktivnu analizu podataka, aplikacija pruža korisnicima uvid u optimalno korištenje solarnih resursa, prilagođavajući se promjenama u vremenskim uvjetima i potrebama korisnika.



## 2. Opis razvijenog proizvoda

Web aplikacija je specifično dizajnirana za kućanstva koja koriste solarni sustav kao primarni izvor energije, pružajući korisnicima sveobuhvatan pregled energetske stanja na jednom centralnom mjestu. Glavni cilj je olakšati korisnicima praćenje, analizu i upravljanje energetske resursima, posebno naglašavajući odluke o prodaji viška energije kako bi se održala pozitivna energetska ravnoteža. Prikaz energetske stanja sastoji se od dva ključna dijela: statističkog i grafičkog.

### 2.1. Statistika

Statistika pruža korisniku dohvat podataka iz baze o potrošnji, proizvodnji i količini energije u spremniku na temelju odabranog vremenskog perioda. Korisnik može na predviđenom kalendaru označiti period koji ga zanima, te može usporediti vrijednosti u odabranom vremenu naprema onima na godišnjoj razini. Ukoliko na kalendaru nije označen nikakav period, statistika za svaku od navedene tri komponente prati i ispisuje njihove prosječne godišnje vrijednosti, te za tekući mjesec njihove vrijednosti: ukupnu mjesečnu proizvodnju, potrošnju i zalihu energije, iznos prosječne dnevne vrijednosti za tekući mjesec svake komponente te maksimalno postignutu vrijednost u jednom danu. Ovi statistički podaci pomažu korisnicima u praćenju i analizi njihove energetske efikasnosti tijekom vremena te dublji uvid u njihovu energetske potrošnju i proizvodnju. Podaci se čuvaju u bazi podataka koji se kasnije koriste za daljnje analize uporabom algoritama strojnog učenja.

## STATISTIKA

Odaberi datum

Today

Yesterday

This Week

Last Week

This Month

Last Month

1 days up to today

1 days starting today

Jun 3, 2024 Jun 3, 2024

June 2024

Sun	Mon	Tue	Wed	Thu	Fri	Sat
26	27	28	29	30	31	1
2	3	4	5	6	7	8
9	10	11	12	13	14	15
16	17	18	19	20	21	22
23	24	25	26	27	28	29
30	1	2	3	4	5	6

Start Date: Mon Jun 03 2024  
End Date: Mon Jun 03 2024

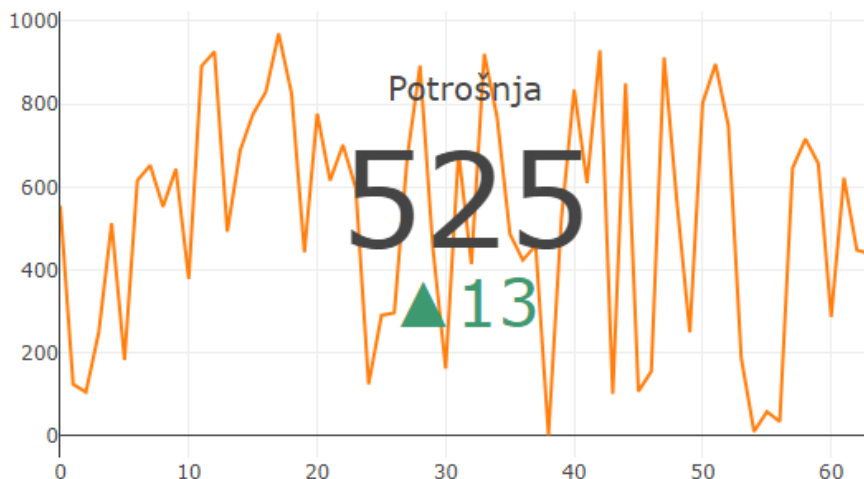
	POTROŠNJA	PROIZVODNJA	SPREMNIK
Prosječna <b>godišnja</b> količina	276	505	300
Potrošnja za <b>odabrani mjesec: prosječna</b> količina energije <b>po danu</b>	82	494	420
Potrošnja za <b>odabrani mjesec: ukupna mjesečna</b> količina energije	2527	15324	13018
Potrošnja za <b>odabrani mjesec: maksimalna</b> količina energije <b>u jednom danu</b>	140	981	869

Slika 2.1. Prozor za prikaz statistike

## 2.2. Grafički prikaz

Grafički prikaz podataka omogućuje korisnicima intuitivno razumijevanje njihovog energetskog stanja. Grafički dio vizualizira dohvaćene statističke podatke u četiri grafa: graf kretanja potrošnje energije, graf omjera potrošene i proizvedene energije kroz godinu, udio ureda u potrošnji energije te prikaz trenutne vrijednosti energije u spremniku. Graf 2.2. prikazuje kako se varijabla "Potrošnja" mijenja kroz period od protekla 64 dana prikazujući tako svakodnevne oscilacije. Os x prikazuje vremenski period od 64 dana, dok y označava vrijednost potrošene količine energije za taj dan. Raspon osi y može se mijenjati približavajući se (zoom in) ili odmakivajući se od grafa (zoom out). Uz oscilacije, prikazuje se i prosjek potrošnje tokom zadnja 64 dana, te razlika trenutnog prosjeka u odnosu na prijašnji; zeleni odmak znači da je razlika negativna, točnije da je trenutni prosjek potrošnje manji od prosjeka prije, suprotno ako je odmak crven, znači da je prosječna potrošnja veća od prijašnje i predstavlja rast u potrošnji.

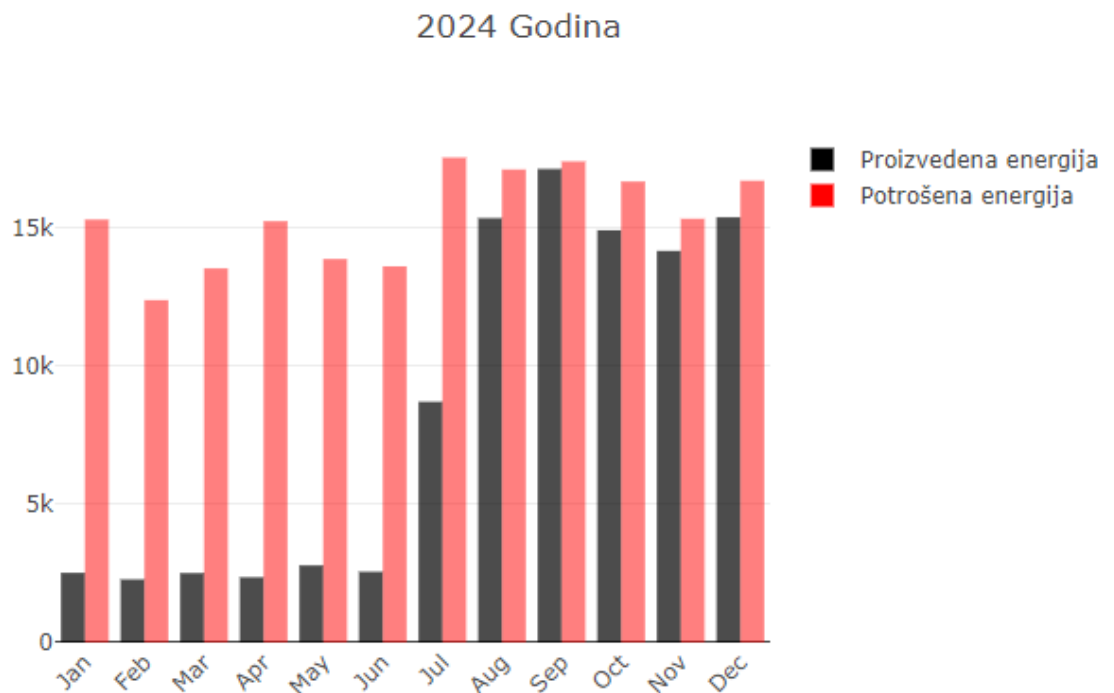
### Potrošnja energije



Slika 2.2. Graf kretanja potrošnje energije

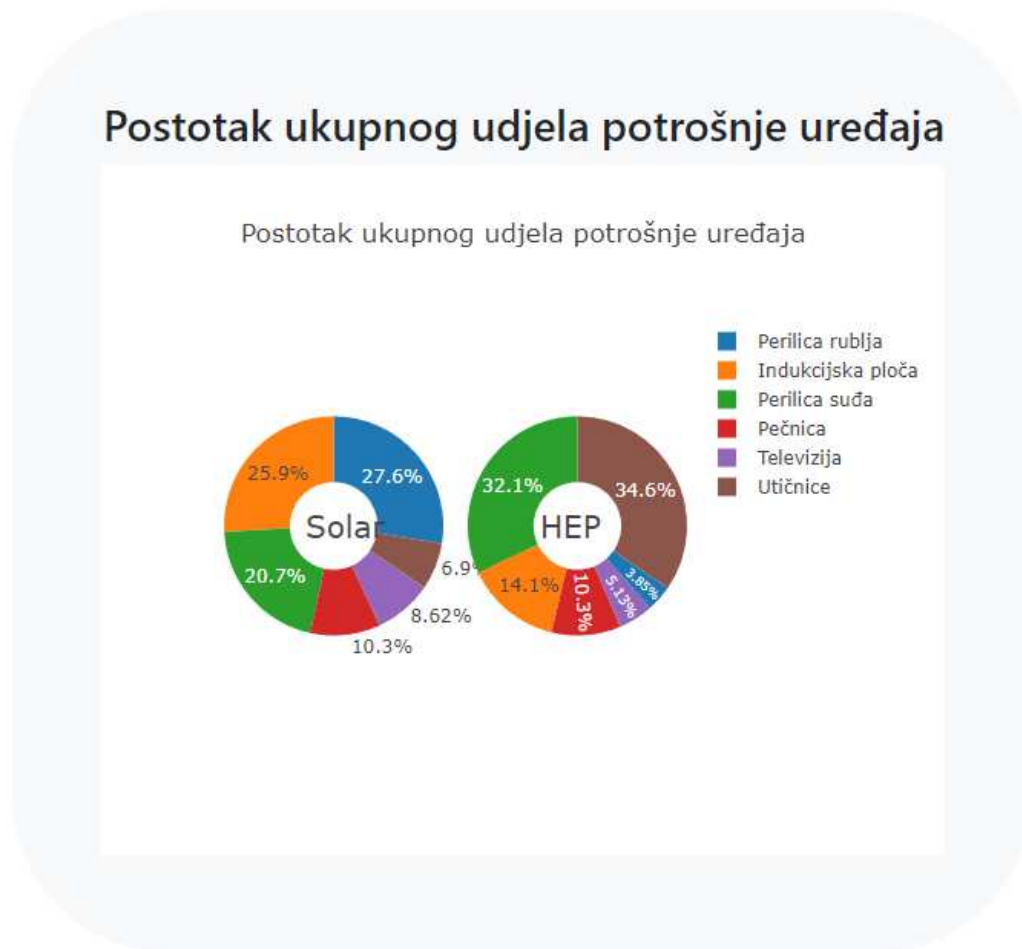
Slijedeći Graf 2.3. vizualizira razliku varijable potrošnje i proizvodnje u mjesecu kroz godinu, pružajući tako jasnu sliku o korisnikovom omjeru mjesečne potrošnje i proizvodnje te ovisnost istog o mjesecu i njegovim vremenskim uvjetima.

## Omjer proizvedene i potrošene energije kroz godinu



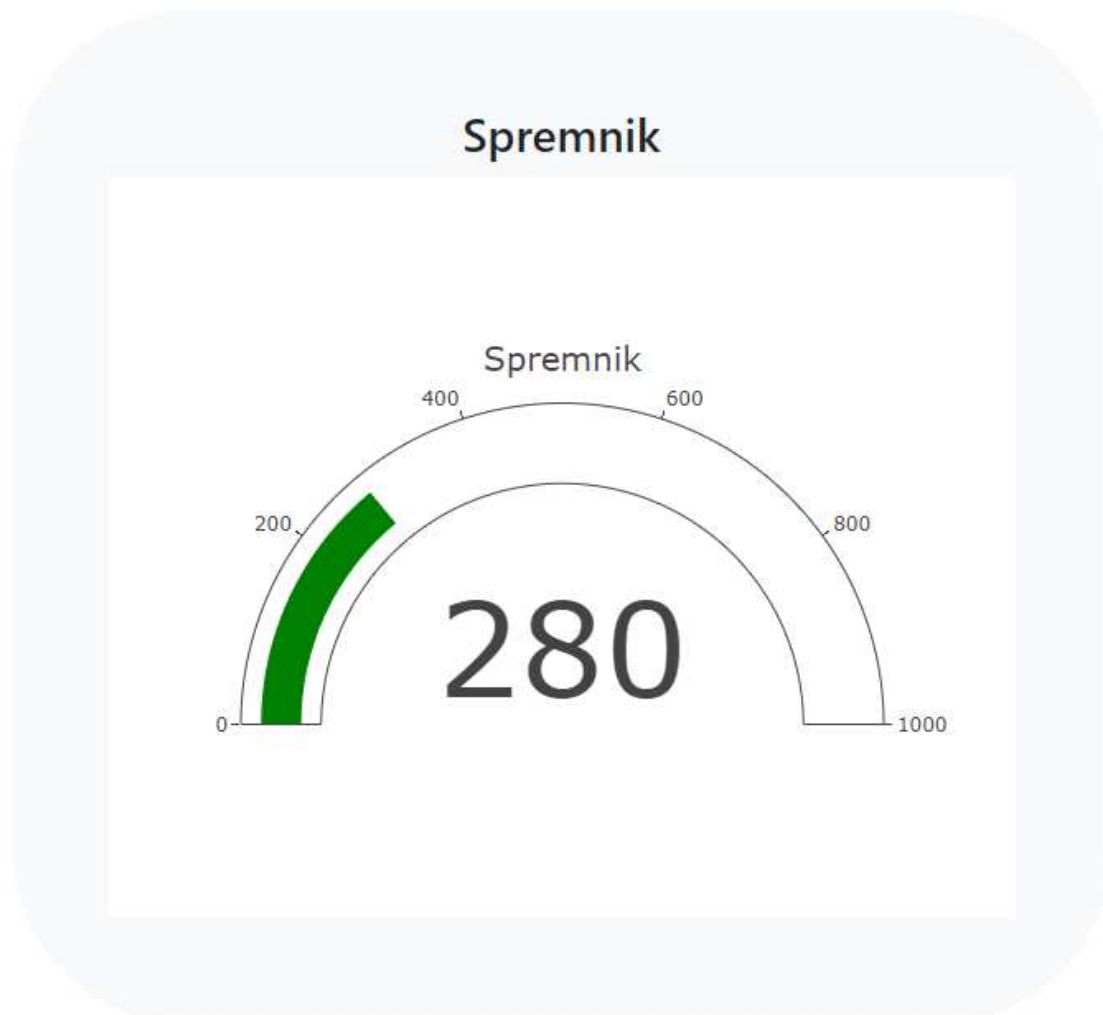
**Slika 2.3.** Graf omjera potrošene i proizvedene energije kroz godinu

Graf 2.4. sadrži dva kružna dijagrama koji predstavljaju ukupnu potrošnju proizvedene energije solarnih panela, odnosno HEP-ove energije. S lijeve strane nalazi se graf potrošnje svakog ureda u zgradi FER-a koji troši energiju proizvedenu na solarnim panelima. Drugi dio grafa naslici prikazuje iste udjele potrošnje po uredima samo na energiji čiji izvor nije solarni panel. Na taj način korisnik vidi koji uredi zahtjevaju najveću količinu energije.



**Slika 2.4.** Graf udjela po uredima u potrošnji energije

Četvrti, ujedno i posljednji Graf 2.5., prikazuje trenutno dnevno stanje zalihe energije u spremniku. Sve brojčane vrijednosti u grafičkom dijelu predstavljaju kilovat-sat (kWh) kao mjernu jedinicu.



**Slika 2.5.** Graf spremljene energije

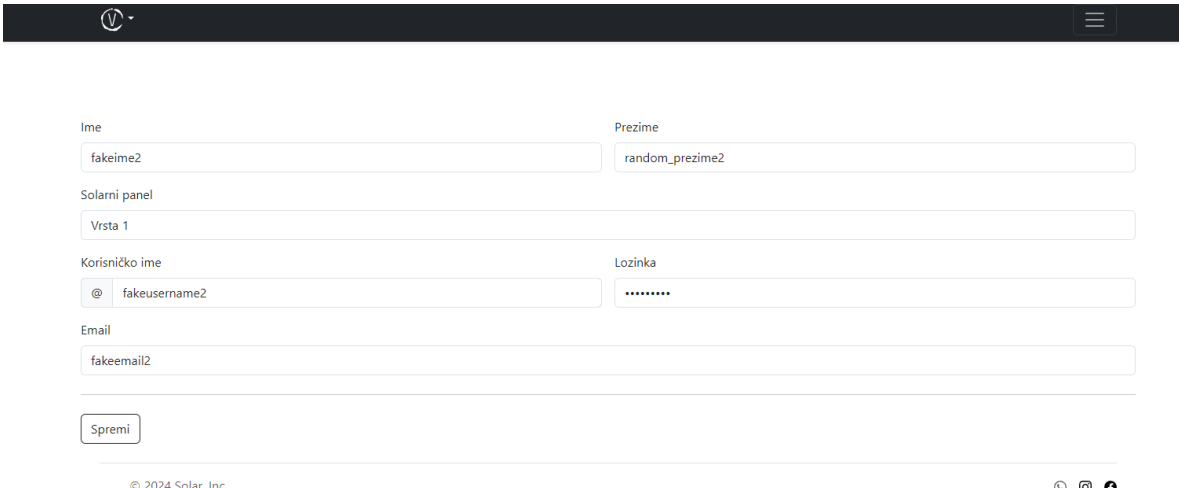
Aplikacija dodatno dohvaća podatke vremenske prognoze za narednih šest dana kako bi bolje informirala korisnike o optimalnom čuvanju ili prodaji viška energije. Podaci se kasnije koriste u proračunima prediktivne analize algoritmima strojnog učenja za predikciju proizvedene te spremljene količine energije. Dnevna temperatura, vidljivost, snaga sunčanih zraka, prodirnost oblaka te opis vremenskih prilika pružaju korisne informacije prilikom donošenja odluka vezanih uz energetske sustav. Slika 2.6. prikazuje prikaz vremenske prognoze za narednih šest dana te numerički rezultat algoritama prediktivne analize.



**Slika 2.6.** Prikaz vremenske prognoze te optimalnog viška energije

## 2.3. Profil

Stranica profila omogućuje korisnicima pregled i upravljanje njihovim trenutno aktivnim osobnim podacima. Čine ih ime, prezime, korisničko ime, lozinka, e-mail te informacije o vrsti solarnog panela koje koriste. Podaci se mogu mijenjati u bilo kojem trenutku pružajući fleksibilnost i personalizaciju korisničkog iskustva. Prikaz stranice profila prikazan je na slici 2.7.



The screenshot shows a user profile data entry form. At the top, there is a dark navigation bar with a logo on the left and a menu icon on the right. The form contains several input fields: 'Ime' (First Name) with the value 'fakeime2', 'Prezime' (Last Name) with the value 'random\_prezime2', 'Solarni panel' (Solar Panel) with the value 'Vrsta 1', 'Korisničko ime' (Username) with the value '@ fakeusername2', 'Lozinka' (Password) with masked characters '.....', and 'Email' with the value 'fakeemail2'. Below the form is a 'Spremi' (Save) button. At the bottom of the page, there is a copyright notice '© 2024 Solar, Inc' and social media icons for WhatsApp, Instagram, and Facebook.

**Slika 2.7.** Prozor podataka profila



## 3. Linearna regresija

U ovom se radu koristi linearna regresija kao glavni algoritam strojnog učenja. Za potrebe ovog rada, ona je implementirana u programskom jeziku R. Prediktivna analiza je širi pojam koji obuhvaća različite metode i tehnike za analizu podataka i predviđanje budućih događaja. Ona koristi različite statističke tehnike i modele za predviđanje budućih ishoda na osnovu povijesnih podataka, a linearna regresija je samo jedna od tih metoda.

### 3.1. Programski jezik R

R je programski jezik i okruženje za statističku analizu, grafičku prezentaciju podataka i strojno učenje. Razvijen je 1990-ih godina i postao je vrlo popularan u znanstvenoj zajednici i industriji zbog svojih snažnih statističkih i vizualizacijskih mogućnosti. Programski jezik R proizašao je iz programskog jezika S, razvijenog za potrebe „Bell Telephone“ Laboratorija. Standardna distribucija R programskog jezika sastoji se od:

- “jezgre” R-a, sa temeljnim funkcijama i tzv. base paketom koji omogućuje osnovnu funkcionalnost,
- kolekcije dodatnih paketa (“osnovni” - base i “preporučeni” - recommended) za upravljanje podacima, vizualizacije i statističke analize.

R je integriran s bogatim repozitorijem paketa zvanom Comprehensive R Archive Network (CRAN) koja omogućuje brzu i jednostavnu instalaciju bilo kojeg paketa iz navedenog repozitorija nakon čega on postaje dio lokalne R instalacije. R se koristi za:

- statističke analize - provođenje različitih statističkih testova i modela,
- grafičku vizualizaciju - stvaranje naprednih grafikona i vizualizacija podataka,

- strojno učenje - razvoj i evaluaciju modela za predikciju i analizu podataka.

Ključne značajke R-a su:

- opsežna biblioteka - mnogo paketa dostupnih za specifične analize i metode,
- prilagodljivost - mogućnost prilagodbe i proširenja funkcionalnosti,
- open-source - besplatan za korištenje i prilagodbu.

Kreativnost i fleksibilnost u korištenju R-a se smatra njegovom velikom prednošću, iako rezultira određenom neformalnošću i liberalnim pristupom programiranju koji nije omiljen korisnicima naviknutim na stroge i formalne programske okvire sa jasnim skupom smjernica i pravila koja se moraju slijediti. Usprkos iznimno velikoj prihvaćenosti jezika R za podatkovne analize te mnoštvu opcija koje nudi korisniku, postoje određena ograničenja: R intenzivno koristi radnu memoriju što je dugo vremena predstavljalo ozbiljno ograničenje, prilično je nekonvencionalan te nije optimiran za brzinu izvođenja. R je prvenstveno namijenjen za interaktivni rad, tj. izvođenje niza strojnih instrukcija koje se dinamički upisuju i izvode uz pomoć programske konzole. Ovo je prilagođeno standardnom procesu analize podataka gdje analitičar može učitavati podatke, čistiti ih, transformirati, razvijati modele, testirati i sl. uz konstantnu povratnu informaciju od računala, mogućnost pregleda međurezultata, izmjene procesa prema trenutnim saznanjima i sl. Neke od alternativa jeziku R su: Python, SAS, SPSS, Weka, Orange.

## 3.2. Strojno učenje

Strojno učenje je grana umjetne inteligencije koja se bavi razvojem algoritama i modela koji omogućuju računalima da uče iz podataka i donose odluke bez eksplicitnog programiranja za svaki zadatak. Bavi se specifičnim načinom programiranja u kojem se računalu ne daju eksplicitne instrukcije, već se očekuje da računalo samostalno dođe do određenih spoznaja na osnovu odabranih podatkovnih skupova i određene metode učenja. Strojno učenje najčešće se dijeli na nadzirano učenje, nenadzirano učenje, te poticano učenje, postoje i dodatne discipline strojnog učenja poput detekcija anomalija. Pojam strojnog učenja često se povezuje sa pojmom dubinske analize podataka (engl. data mining) i otkrivanja znanja iz podatkovnih skupova (engl. knowledge discovery

from data, KDD).

### 3.2.1. Nadzirano učenje

Modeli se treniraju na označenim podacima gdje su ulazni podaci povezani s odgovarajućim izlazima. Podaci uključuju ulazne i izlazne parove  $(x, y)$ , gdje je  $x$  ulazna značajka, a  $y$  željeni izlaz. Cilj je naučiti funkciju koja mapira ulaz na izlaz, omogućujući modelu da predvidi  $y$  za nove  $x$ . Nadzirano učenje dijeli se na dvije kategorije:

- regresija - predviđanje kontinuiranih vrijednosti (npr. cijena kuće),
- klasifikacija - predviđanje diskretnih kategorija (npr. klasifikacija e-maila kao spam ili ne-spam).

### 3.2.2. Nenadzirano učenje

Nenadzirano učenje radi s neoznačenim podacima, gdje sustav pokušava pronaći skrivene strukture ili obrasce u podacima. Neki od osnovnih algoritama grupiranja su:

- K-means - radi tako da dijeli podatke u  $k$  skupina na temelju udaljenosti do središta klastera,
- hijerarhijsko grupiranje - stvara hijerarhiju grupa koja može biti predstavljena stablom.

Neki od algoritama smanjenja dimenzionalnosti su:

- Principal Component Analysis (PCA) - tehnika koja smanjuje dimenzionalnost podataka identificirajući glavne osi varijance,
- t-SNE - vizualizacijska tehnika koja smanjuje dimenzionalnost za prikaz visokodimenzionalnih podataka u 2D ili 3D prostoru.

Detekcija anomalija (engl. Anomaly Detection) je proces identifikacije podataka koji se značajno razlikuju od većine ostalih podataka. Ovi "anomalni" podaci mogu ukazivati na važne događaje, greške ili neočekivane promjene u sustavu. Osnovni algoritam detekcije anomalija je Z-Score oblika:

$$Z = \frac{X - \mu}{\sigma} \quad (3.1)$$

### 3.2.3. Pojačano učenje

Pojačano učenje uključuje agenta koji uči putem interakcije s okolinom. Agent donosi odluke, dobiva povratne informacije u obliku nagrada ili kazni, i prilagođava svoje akcije kako bi maksimizirao ukupnu nagradu tijekom vremena. Neki od osnovnih pristupa pojačanom učenju su:

- Markovljevi lanci odluke (engl. Markov Decision Process, MDP) - modeli koji definiraju skup stanja, akcija i nagrada za optimizaciju dugoročnog ishoda,
- Q-učenje - algoritam koji uči funkciju nagrade za svaku akciju u svakom stanju, bez potrebe za modelom okoline.

### 3.3. Linearna regresija unutar aplikacije

Linearna regresija je statistička metoda koja se koristi za modeliranje odnosa između jedne ovisne varijable i jedne ili više neovisnih varijabli. Spada pod kategoriju nadziranog strojnog učenja zato što ima za cilj pronalazak mapiranja između ulaznih/neovisnih varijabli i jedne izlazne/ovisne varijable. Dodatak tome je svrha predviđanja kontinuiranih vrijednosti. Cilj je pronaći koeficijente koji najbolje odgovaraju podacima. Ovisno o broju ulaznih varijabli (koje se također nazivaju i značajke), dijeli se u dvije skupine:

- jednostavna linearna regresija - predviđa kontinuiranu vrijednost na temelju jedne značajke, ona ima oblik:

$$\hat{y} = \beta_0 + \beta_1 x$$

gdje su:

- $\hat{y}$ : ovisna varijabla,
  - $\beta_0$ : presjek (engl. intercept),
  - $\beta_1$ : koeficijent nagiba (engl. slope),
  - $x$ : neovisna varijabla.
- višestruka linearna regresija - koristi više značajki za predviđanje, analogno jed-

nostavnoj linearnoj regresiji ima oblik:

$$\hat{y} = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \dots + \beta_n x_n$$

Kod implementiran u programskom jeziku R koristi višestruku linearnu regresiju za modeliranje sljedećih varijabli:

**model spremnik:** spremnik kao ovisna varijabla, potrošnja i proizvodnja kao neovisne varijable,

**model potrošnja:** potrošnja kao ovisna varijabla, proizvodnja i datum kao neovisne varijable,

**model proizvodnja:** proizvodnja kao ovisna varijabla, datum kao neovisna varijabla. Ovi modeli se zatim koriste za predikciju na testnom setu i za buduće datume, kako bi se procijenio višak energije. Nakon uključivanja potrebnih knjižnica programskog jezika R, slijedi glavna funkcija koja izvodi linearnu regresiju. U njoj se podaci dijele na skup za treniranje (kojeg čini 70% podataka) i skup za testiranje (kojeg čini preostalih 30% podataka). Pseudokod funkcije za izvođenje linearne regresije prikazan je algoritmom 3.1.

```

1 // Podijelimo podatke na trening (70%) i test (30%) setove
   koriste i razdjelitelj podataka
2 set_seed(123) // Za reproduktivnost
3 train_index = createDataPartition(df['proizvodnja'], p =
   0.7)
4 train_data = df[train_index]
5 test_data = df[~train_index]
6 model_spremnik = linear_regression(spremnik ~ potrosnja +
   proizvodnja, train_data)
7 model_potrosnja = linear_regression(potrosnja ~ proizvodnja
   + datum, train_data)
8 model_proizvodnja = linear_regression(proizvodnja ~ datum,
   train_data)
9 return model_spremnik, model_potrosnja, model_proizvodnja

```

Algoritam 3..1: pseudokod linearne regresije

Potom se izvršavaju tri linearne regresije nad svakim od navedenih modela te se provodi predikcija vrijednosti za narednih 5 dana prikazana pseudokodom 3.2.

```

1 # Predikcija za testni skup podataka
2 Za svaki redak u testnom skupu podataka:
3     Predvidi vrijednosti za 'spremnik' koriste i model '
4         model_spremnik'
5     Pohrani predvi enu vrijednost u testni skup podataka
6 Za svaki redak u testnom skupu podataka:
7     Predvidi vrijednosti za 'potrosnja' koriste i model '
8         model_potrosnja'
9     Pohrani predvi enu vrijednost u testni skup podataka
10 Za svaki redak u testnom skupu podataka:
11     Predvidi vrijednosti za 'proizvodnja' koriste i model '
12         model_proizvodnja'
13     Pohrani predvi enu vrijednost u testni skup podataka
14 # Predikcija buducih vrijednosti na temelju modela obucenog na
15     cijelom skupu podataka
16 Kreiraj niz datuma koji pocinju od zadnjeg datuma u skupu
17     podataka i idu 5 dana unaprijed
18 Kreiraj novi podatkovni okvir (future_df) koji sadrzi:
19     Kolonu 'proizvodnja' sa srednjom vrijednoscu proizvodnje iz
20     originalnog skupa podataka, za svaki od buducih datuma
21     Kolonu 'potrosnja' sa srednjom vrijednoscu potrosnje iz
22     originalnog skupa podataka, za svaki od buducih datuma
23     Kolonu 'datum' s buducim datumima
24 Za svaki redak u future_df:
25     Predvidi vrijednosti za 'spremnik' koriste i model '
26         model_spremnik'
27     Pohrani predvi enu vrijednost u future_df
28 Za svaki redak u future_df:
29     Predvidi vrijednosti za 'potrosnja' koriste i model '
30         model_potrosnja'
31     Pohrani predvi enu vrijednost u future_df
32 Za svaki redak u future_df:
33     Predvidi vrijednosti za 'proizvodnja' koriste i model '
34         model_proizvodnja'

```

### Algoritam 3.2: testiranje gotovod modela linearne regresije

Predikcije se rade na testnom setu za svaku od modeliranih varijabli koristeći srednje vrijednosti proizvodnje i potrošnje iz cijelog skupa podataka. Slijedi izračun performansi modela pomoćnim funkcijama; izračun korijena srednjeg kvadrata pogreške (engl. Root Mean Square Error, RMSE) i srednje apsolutne pogreške (engl. Mean Absolute Error, MAE) između stvarnih i predikcijskih vrijednosti. Formula za RMSE prikazana je u 3.2

$$\text{RMSE} = \sqrt{\frac{1}{n} \sum_{i=1}^n (\hat{y}_i - y_i)^2} \quad (3.2)$$

Formula za MAE dana je formulom 3.3 kao:

$$\text{MAE} = \frac{1}{n} \sum_{i=1}^n |\hat{y}_i - y_i| \quad (3.3)$$

Rezultat su već spomenute metrike RMSE i MAE spremljene u csv datoteku. Funkcija za osvježavanje podataka i regresije koja dohvaća podatke s API-ja prikaza je pseudokodom 3:

```

1 updateDataAndRegression <- function() {
2   df <- getChartData()
3   performLinearRegression(df)
4 }
5 updateDataAndRegression()
```

### Algoritam 3.3: pojednostavljeni prikaz dohvaćanja i osvježavanja podataka



## 4. Tehničke značajke

Projekt integrira nekoliko ključnih tehnologija i funkcionalnosti kako bi pružio korisnicima informacije o potrošnji i proizvodnji energije te vremenskoj prognozi.

**Bootstrap proširenje 'bootstrap/dist/css/bootstrap.min.css'**: služi za ubrzanje razvoja izrade CSS koda te olakšanje implementacije responzivnog i estetski ugodnog korisničkog sučelja koji su lako prilagodljivi

**'Openweathermap.org'**: Besplatan API za dohvaćanje vremenske prognoze za narednih šest dana. API pruža važne informacije poput vrste vremena (je li sunčano s puno ili malo oblaka, je li kišovito...), vidljivosti i jačine sunčanih zraka za vrijeme prisutnosti oblaka. Također omogućuje korisnicima praćenje uvjeta koji mogu utjecati na proizvodnju energije solarnih panela.

**Lokalna baza podataka**: Solarni panel bilježi svoje dnevne vrijednosti proizvodnje energije u lokalnu bazu podataka. Aplikacija koristi ove podatke za izračun potrošnje i zaliha energije, pružajući korisnicima jasne informacije o njihovoj energetskej potrošnji. Baza podataka također sadrži informacije o korisnicima, što je ključno za funkcionalnost stranice profila, te starije statističke podatke koji će osigurati bolje poznavanje navika panela, ali i prikaz starijih statističkih podataka na korisnikov zahtjev.

**Profilna stranica**: Na profilnoj stranici korisnici mogu pregledavati i mijenjati svoje podatke.

**Predikcija i održavanje zaliha**: algoritam višestruke linearne regresije izračunava prediktivne vrijednosti za svaki model podataka (proizvodnja, potrošnja i spremnik) na temelju "train" skupa podataka koje čine sedamdeset posto dohvaćenih povijesnih podataka za svaki model iz lokalne baze podataka.

**Programski jezik R:** izvrsna podloga za statističku analizu podataka koja pruža mogućnost vizualizacije grafova te primjenu strojnog učenja. U njemu se dohvaćeni podaci iz baze podataka provode kroz algoritme predikcije te algoritam višestruke linearne regresije i pruža se način vizualizacije te izvoz podataka

## 5. Metodološki put

Ovo poglavlje objašnjava metodološki put razvoja aplikacije za praćenje potrošnje i proizvodnje energije, kao i upravljanje profilima korisnika. Poglavlje je podijeljeno u nekoliko sekcija koje pokrivaju ključne komponente sustava, uključujući backend server, frontend aplikaciju, bazu podataka, te komunikacijske puteve između različitih dijelova sustava.

### 5.1. Backend server

U aplikaciji se koristi Express.js kao glavni web radni okvir (engl. framework) za definiranje ruta i obradu HTTP zahtjeva. PostgreSQL koristi se kao baza za pohranu podataka o potrošnji, proizvodnji i spremniku energije te drugim relevantnim statistikama. Backend server se povezuje na PostgreSQL bazu podataka pomoću pg modula.

#### 5.1.1. Rute definirane u Express.js

Backend server definira nekoliko ključnih ruta za dohvat i ažuriranje podataka, one u Express.js obrađuju zahtjeve (HTTP GET i PUT) za dohvat i ažuriranje podataka prateći arhitektonski stil REST princip. Svaka ruta odgovara određenoj API krajnjoj točki i povezana je s odgovarajućom funkcijom upita iz datoteke 'queries.js'. Ove rute obrađuju zahtjeve koji se odnose na potrošnju energije, proizvodnju, korisnički profil i ažuriranja. Razvijene su sljedeće rute:

- */api/potrosnja*: dohvaća zadnjih 64 reda iz tablice "panel" (potrošnja energije)
- */api/zbroj*: dohvaća zbrojeve potrošnje, proizvodnje i zaliha po mjesecima
- */api/godisnji-zbroj*: dohvaća godišnje zbrojeve potrošnje, proizvodnje i zaliha za

određenu godinu

- */api/profile*: dohvaća podatke za profilnu stranicu korisnika s određenim ID-om
- */api/update-profile/:id*: ažurira podatke profila korisnika s određenim ID-om.

## 5.2. Frontend

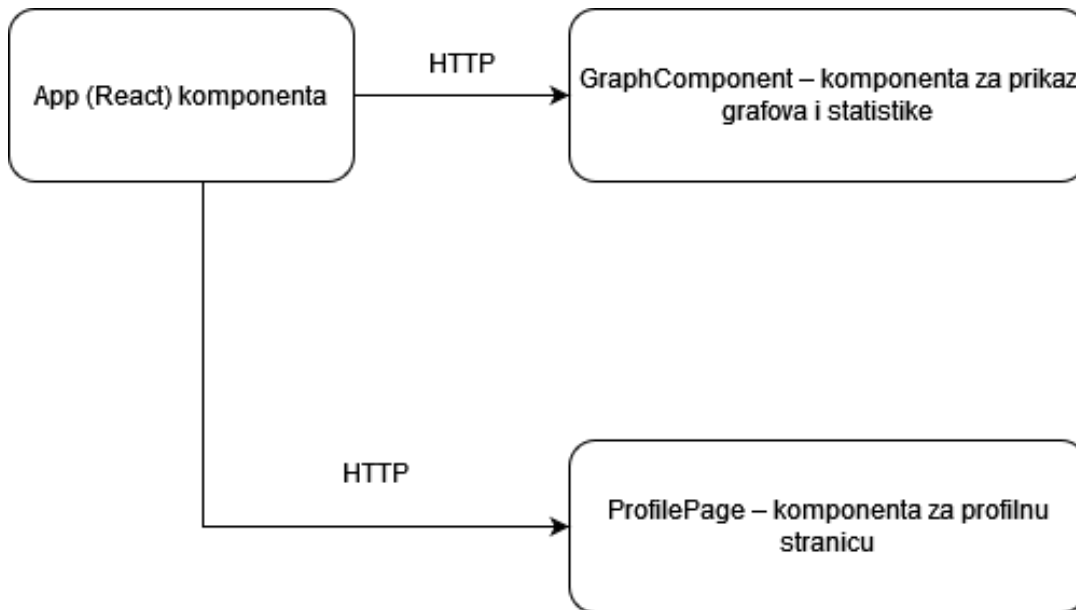
Za izradu korisničkog sučelja, aplikacija koristi biblioteku React. Glavna komponenta **GraphComponent** prikazuje grafove potrošnje, proizvodnje i druge statistike. Grafikon koristi podatke dohvaćene s backend servera putem HTTP zahtjeva. Za izradu i prikaz grafičkih komponenti koristi se React-ov API „**react-plotly.js**“ kako bih bio veći izbor vrste grafova koji se žele koristiti. Datoteka **App.jsx** definira glavnu React komponentu koja postavlja rutiranje između različitih stranica koristeći React Router za navigaciju između glavne stranice i stranice profila:

- */* - glavna stranica s grafovima potrošnje i proizvodnje
- */profil* - profilna stranica korisnika

## 5.3. Blokowska shema arhitekture

### 5.3.1. Klijentska strana (React)

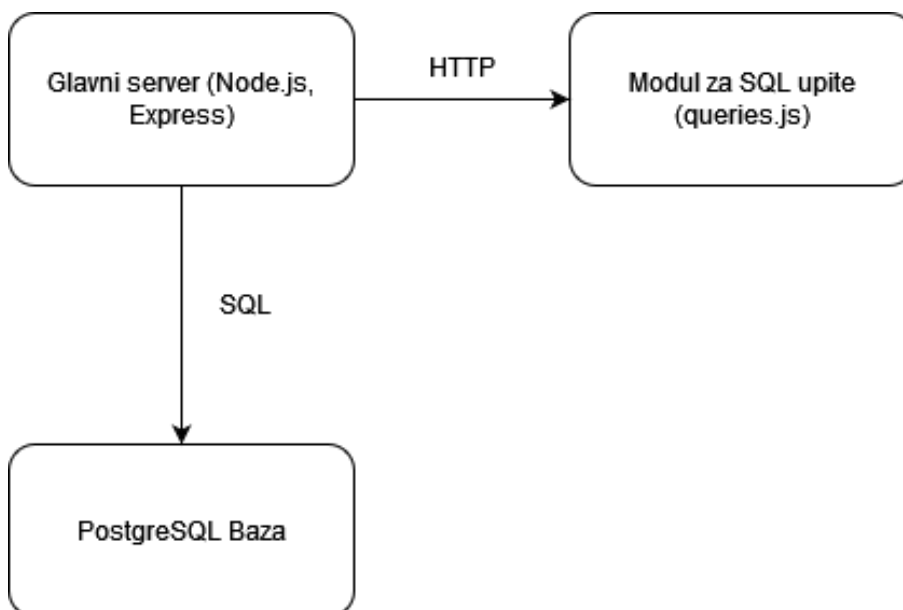
Ovo su glavne React komponente aplikacije, gdje svaka komponenta ima specifičnu funkciju: **App.jsx** upravlja rutiranjem, **GraphComponent.jsx** prikazuje grafove, **HeaderComponent.jsx** prikazuje navigaciju i korisnički profil, **FooterComponent.jsx** prikazuje podnožje stranice, **ProfilePage.jsx** prikazuje stranicu profila, a **WeatherComponent.jsx** prikazuje vremensku prognozu. Ove komponente čine korisničko sučelje aplikacije. Glavna stranica na kojoj je prikazan sav sadržaj aplikacije sastoji se od **GraphComponenta** koji uključuje **HeaderComponent**, **FooterComponent** i **WeatherComponent**.



Slika 5.1. Blokowska shema frontend arhitekture

### 5.3.2. Backend server (Node.js, Express, PostgreSQL)

Datoteka **index.js** je glavni server file koji postavlja Express aplikaciju, **queries.js** je modul koji sadrži SQL upite za dohvat i ažuriranje podataka u PostgreSQL bazi, **package.json** je konfiguracija projekta s potrebnim paketima, a **services/api.js** sadrži funkcije za izvođenje HTTP zahtjeva prema backend serveru pomoću fetch (axios) biblioteke. Pseudokod osnovne konfiguracije servera prikazan je algoritmom 5.1.



Slika 5.2. Blokowska shema backend arhitekture

```
1 1. Uvezi potrebne module:
2   - express
3   - pg (PostgreSQL)
4   - cors
5   - child_process
6 2. Inicijaliziraj Express aplikaciju.
7 3. Definiraj port na kojem ce server slusati (5000).
8 4. Omoguci CORS (Cross-Origin Resource Sharing).
9 5. Konfiguriraj PostgreSQL bazu podataka
10 6. Pokreni server i slusaj na definiranom portu.
11 7. Prikazi poruku u konzoli kada je server pokrenut.
```

Algoritam 5.1: Pseudokod osnovne konfiguracije servera bez endpoint-ova

### 5.3.3. Baza podataka

Temeljni pregled baze podataka čine dvije tablice, panel i profil. Tablica "panel" sadrži sljedeće varijable:

- id: SERIAL PRIMARY KEY
- potrošnja: INTEGER
- proizvodnja: INTEGER
- zaliha: INTEGER
- datum: DATE

Tablica profil sadrži varijable:

- ime: VARCHAR
- prezime: VARCHAR
- solarpanel: VARCHAR
- username: VARCHAR PRIMARY KEY

- password: VARCHAR
- email: VARCHAR
- id: INTEGER

U tablici panel, svako kućanstvo ima svoj jedinstveni ID koji je ujedno i primarni ključ preko kojeg su tablice panel i profil povezani. Naknadno slijedi iznos dnevne izračunate i unesene potrošnje, dnevne solarno proizvedene te dnevne zalihe energije na današnji datum. U tablici profil za zadani ID pamte se profilni podaci:

- ime: VARCHAR
- prezime: VARCHAR
- solarpanel (vrsta solarnog panela): VARCHAR
- username (korisničko ime): VARCHAR PRIMARY KEY
- password (lozinka): VARCHAR
- email (email korisnika): VARCHAR
- id: INTEGER

Povezivanje s bazom podataka prikazano je slijedećim kodom:

```

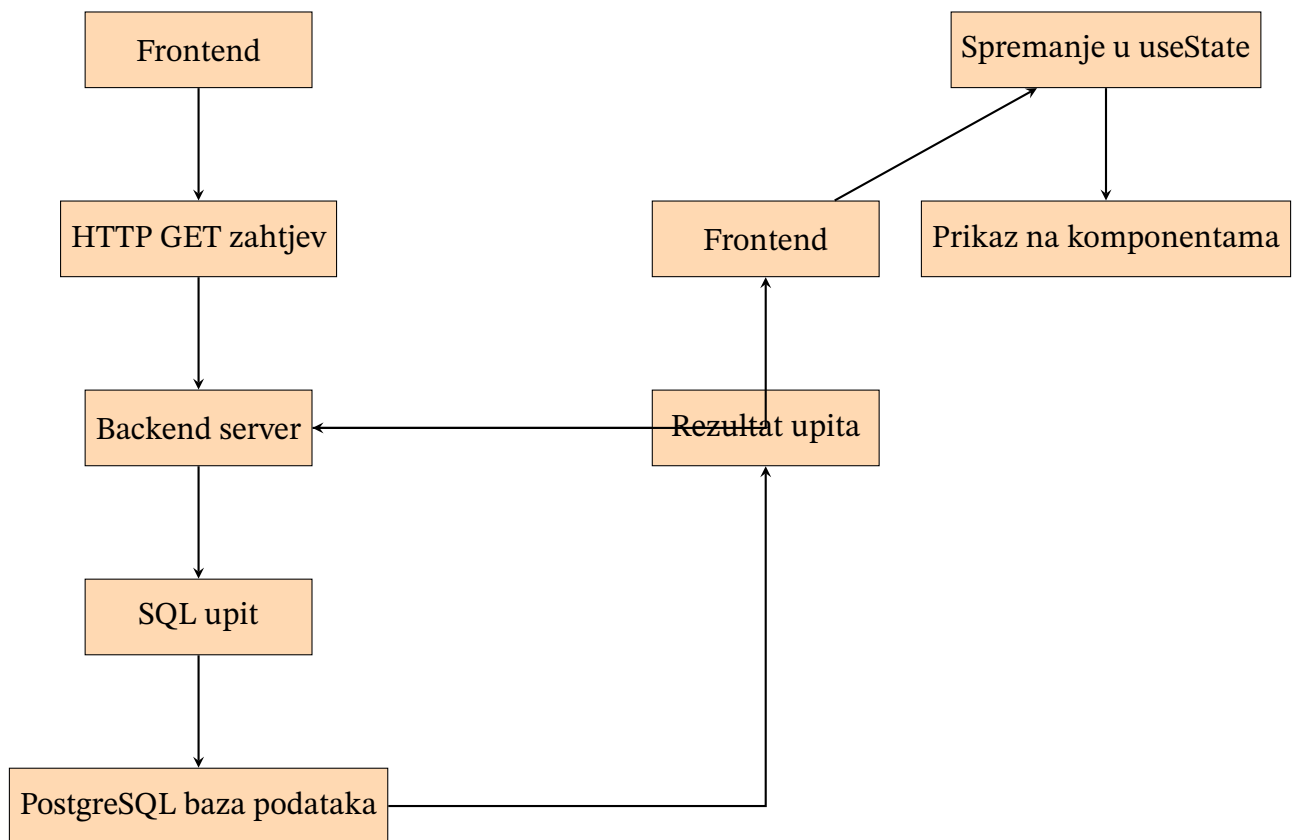
1 // Konfiguracija za PostgreSQL
2 const pool = new Pool({
3   user: 'postgres',
4   host: 'localhost',
5   database: 'solar',
6   password: 'bazepodataka',
7   port: 5434,
8 });

```

Algoritam 5..2: Konfiguracije baze podataka

### 5.3.4. Komunikacijski put

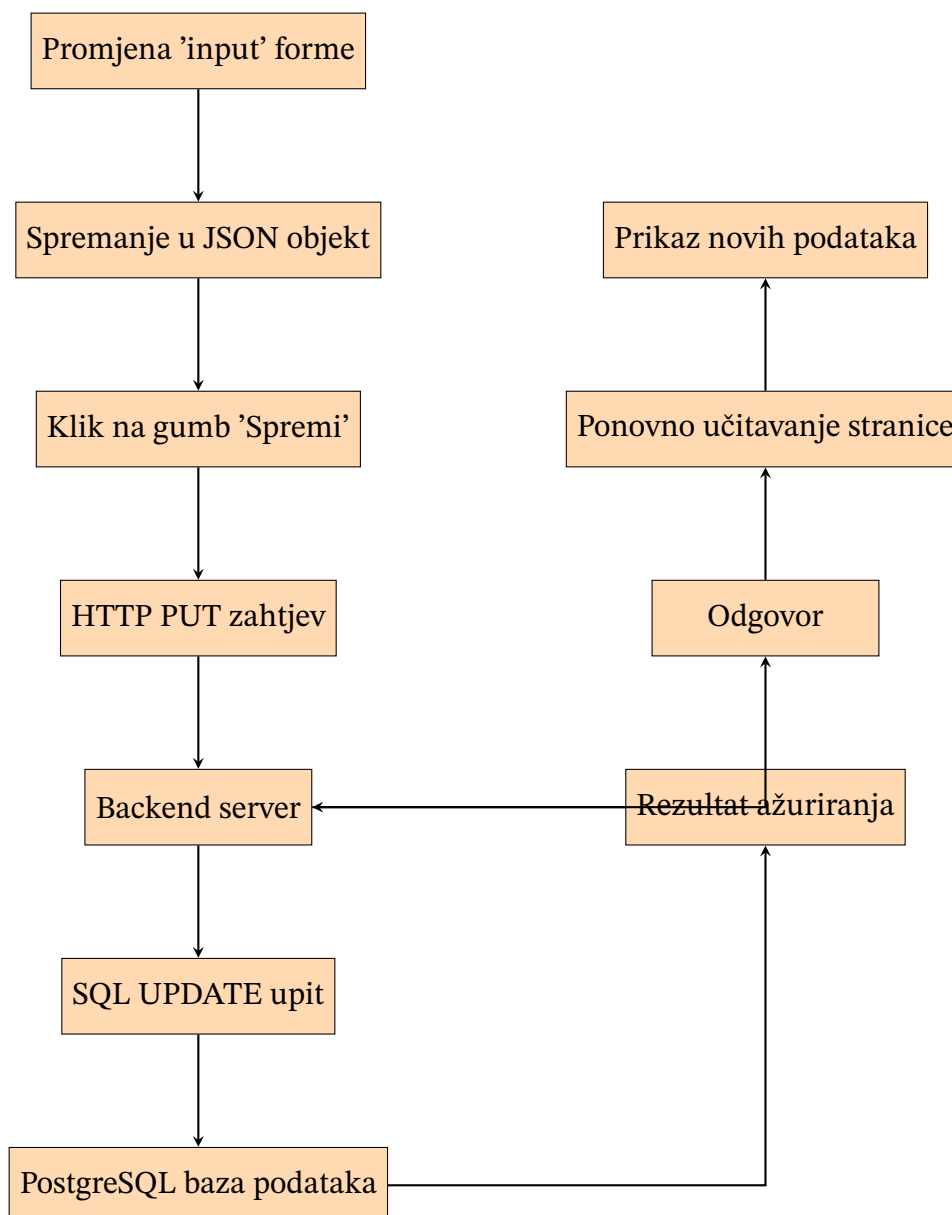
#### Dijagram toka za dohvat podataka



**Slika 5.3.** Dijagram toka za dohvat podataka prikazuje korake koje frontend komponenta poduzima prilikom slanja HTTP GET zahtjeva prema backend serveru, izvršavanja SQL upita, te vraćanja rezultata nazad na frontend gdje se podaci spremaju u useState i prikazuju na komponentama.



## Dijagram toka za ažuriranje profila



**Slika 5.4.** Dijagram toka za ažuriranje profila prikazuje korake koje frontend komponenta poduzima prilikom promjene vrijednosti 'input' forme, spremanja novih vrijednosti u JSON objekt, slanja HTTP PUT zahtjeva prema backend serveru, izvršavanja SQL UPDATE upita, te vraćanja rezultata nazad na frontend gdje se podaci prikazuju nakon ponovnog učitavanja stranice.

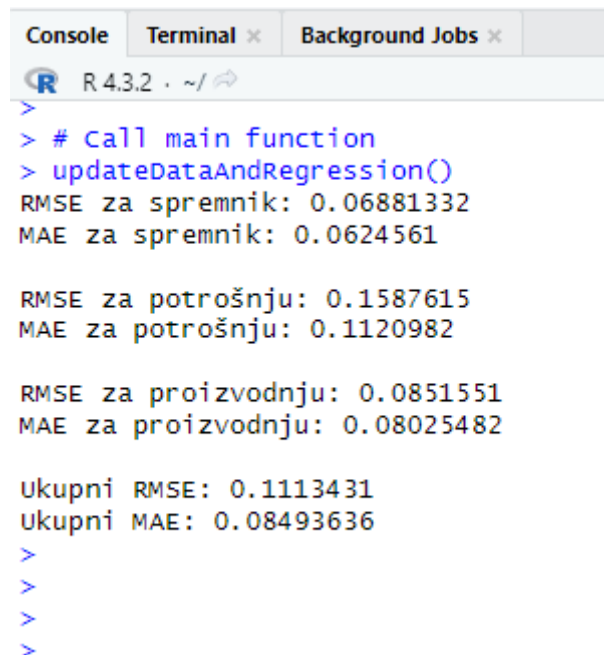
Za periodično dohvaćanje novih podataka (svakih 100 sekundi) koristi se `useEffect` u Reactu kako bi se osvježavali grafovi i prikazivali najnoviji podaci. Ovom arhitekturom omogućava se komunikacija između klijentske i serverske strane u stvarnom vremenu.

```
1 const intervalId = setInterval(fetchData, 100000);  
2 const stopIntervalTime = 1800000; // 30 minuta  
3 setTimeout(( ) => clearInterval(intervalId), stopIntervalTime);
```

Algoritam 5..3: Periodično dohvaćanje podataka

## 6. Rezultati i rasprava

Da bismo procijenili učinkovitost modela, analiziramo vrijednosti srednje kvadratne pogreške (RMSE) i srednje apsolutne pogreške (MAE). Ove metričke performanse daju nam uvid u točnost modela u predikciji stvarnih vrijednosti. Dobivene greške prikazane su slikom 6.1. te označavaju slijedeće:



```
>
> # call main function
> updateDataAndRegression()
RMSE za spremnik: 0.06881332
MAE za spremnik: 0.0624561

RMSE za potrošnju: 0.1587615
MAE za potrošnju: 0.1120982

RMSE za proizvodnju: 0.0851551
MAE za proizvodnju: 0.08025482

Ukupni RMSE: 0.1113431
Ukupni MAE: 0.08493636
>
>
>
>
```

Slika 6.1. Rezultatne greške algoritma

### 6.1. Predikcija spremnika (*spremnik*)

- **RMSE (srednja kvadratna pogreška):** 0.06881332
- **MAE (srednja apsolutna pogreška):** 0.0624561

Ove vrijednosti RMSE i MAE za spremnik su relativno male, što sugerira da model ima dobru točnost u predikciji vrijednosti spremnika i u prosjeku griješi 6%. Manje vri-

jednosti RMSE i MAE ukazuju na manju razliku između predikcija modela i stvarnih vrijednosti.

## 6.2. Predikcija potrošnje (*potrošnja*)

- **RMSE (srednja kvadratna pogreška):** 0.1587615
- **MAE (srednja apsolutna pogreška):** 0.1120982

Vrijednosti RMSE i MAE za potrošnju su veće u usporedbi s spremnikom, što znači da model ima veću pogrešku u predikciji potrošnje. Ipak, ove vrijednosti su još uvijek relativno niske, što ukazuje na prihvatljivu razinu točnosti.

## 6.3. Predikcija proizvodnje (*proizvodnja*)

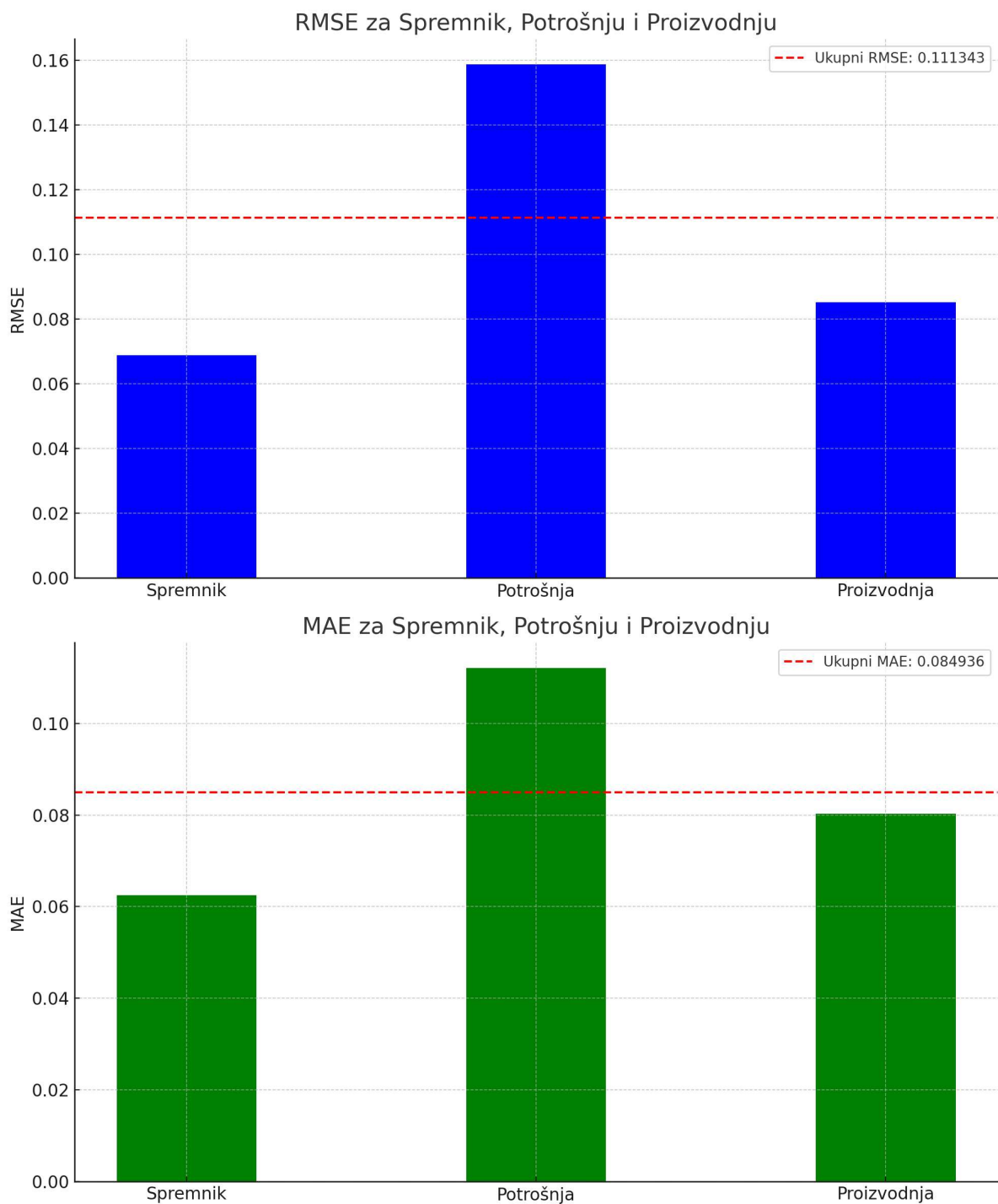
- **RMSE (srednja kvadratna pogreška):** 0.0851551
- **MAE (srednja apsolutna pogreška):** 0.08025482

Vrijednosti RMSE i MAE za proizvodnju su također relativno niske, što sugerira da model ima dobru točnost u predikciji proizvodnje. Ove metričke performanse su slične onima za spremnik, što ukazuje na konzistentnu točnost modela.

## 6.4. Ukupne metričke performanse

- **Ukupni RMSE:** 0.1113431
- **Ukupni MAE:** 0.08493636

Ukupne vrijednosti RMSE i MAE, koje uzimaju u obzir sve predikcije (spremnik, potrošnja i proizvodnja), ukazuju na prihvatljivu ukupnu točnost modela. Niže vrijednosti RMSE i MAE općenito znače bolji model jer ukazuju na manju pogrešku između predikcija i stvarnih vrijednosti. Vrijednosti svih pogrešaka za svaki model varijable prikazane su na grafu 6.2.



**Slika 6.2.** Grafički prikaz rezultatnih pogrešaka MAE i RMSE

## **RMSE (srednja kvadratna pogreška)**

- Plave trake predstavljaju RMSE vrijednosti za spremnik, potrošnju i proizvodnju.
- Crvena isprekidana linija označava ukupni RMSE (0.1113431).

## **MAE (srednja apsolutna pogreška)**

- Zelene trake predstavljaju MAE vrijednosti za spremnik, potrošnju i proizvodnju.
- Crvena isprekidana linija označava ukupni MAE (0.08493636).

## 7. Zaključak

Razvoj aplikacije za prediktivni nadzor naprednih zgrada rezultirao je korisnim alatom za optimizaciju energetske bilance korištenjem obnovljivih izvora energije. Aplikacija omogućava učinkovito praćenje i analizu proizvodnje i potrošnje solarne energije kroz intuitivno web sučelje, što korisnicima olakšava razumijevanje energetske trendova i planiranje potrošnje. Implementacija fotonaponskih panela i korištenje PostgreSQL baze podataka osigurali su pouzdanu pohranu i pristup podacima. Uz to, primjena višestruke linearne regresije za prediktivnu analitiku omogućuje bolje planiranje i upravljanje energetske resursima, uključujući optimalnu prodaju viška proizvedene energije.

Postignuti rezultati zadovoljavaju zadane kriterije i ciljeve projekta. Projekt nudi dobru osnovu za buduće nadogradnje i dodatne analize, čineći ga vrijednim doprinosom u području održivog upravljanja energijom.

## Literatura

- [1] Damir, P. (2024). *Programirajmo u R-u*. Zagreb: 1. izdanje.
- [2] Sarker, I. H. (2021). Machine Learning: Algorithms, Real-World Applications and Research Directions. *SN Computer Science*, 2(160). doi:10.1007/s42979-021-00592-x. <https://doi.org/10.1007/s42979-021-00592-x>.
- [3] Openweathermap. (2024). API Documentation. <https://openweathermap.org/api>.
- [4] Logo. (2024). API Documentation. <https://logo.com/>.
- [5] Plotly. (2024). API Documentation. <https://plotly.com/javascript/react/>.
- [6] Bootstrap. (2024). API Documentation. <https://getbootstrap.com/>.



# Sažetak

## Aplikacija za prediktivni nadzor pametnih zgrada

Vinko Brkić

U ovom radu razvijena je Web aplikacija za prediktivni nadzor naprednih zgrada, s ciljem optimizacije energetske bilance putem korištenja obnovljivih izvora energije. Instaliranjem fotonaponskih panela na krovove rezidencijalnih zgrada, omogućeno je praćenje i analiza distribucije solarne energije. Aplikacija prikuplja podatke o proizvodnji i potrošnji energije panela spremajući ih u PostgreSQL bazu podataka i vizualizira ih putem web sučelja izrađenog u programskom okviru Native. Prikazani grafički i statistički podaci uključuju povijesne trendove potrošnje energije različitih uređaja u zgradi, ukupnu proizvodnju, pohranu i potrošnju energije kroz odabrani period te buduće vanjske vremenske uvjete. Dodatno, aplikacija pruža prediktivnu analitiku koristeći višestruku linearnu regresiju za analizu te procjenu budućih varijabli s ciljem optimalne prodaje proizvedenog viška energije.

**Ključne riječi:** prediktivni nadzor; napredne zgrade; obnovljivi izvori energije; fotonaponski paneli; višestruka linearna regresija; web sučelje; prediktivna analitika

# Abstract

## Application for predictive monitoring of smart buildings

Vinko Brkić

In this project, a web application for predictive monitoring of advanced buildings was developed, aiming to optimize the energy balance through the use of renewable energy sources. By installing photovoltaic panels on the roofs of residential buildings, it enables the monitoring and analysis of solar energy distribution. The application collects data on the production and consumption of energy from the panels, storing it in a PostgreSQL database, and visualizes it through a web interface built with the Native framework. The displayed graphical and statistical data include historical trends of energy consumption by different devices in the building, total production, storage, and consumption of energy over a selected period, as well as future external weather conditions. Additionally, the application provides predictive analytics using multiple linear regression to analyze and estimate future variables with the aim of optimizing the sale of surplus energy produced.

**Keywords:** predictive monitoring; advanced buildings; renewable energy sources; photovoltaic panels; multiple linear regression; web interface; predictive analytics