

Sveučilište Jurja Dobrila
Fakultet Informatike u Puli

Buzzer
Web aplikacija za pčelare

David Kostić, 0303082313

David Maglica, 0066306616

Kolegij: Web aplikacije

Pula, rujan 2021.

Sadržaj

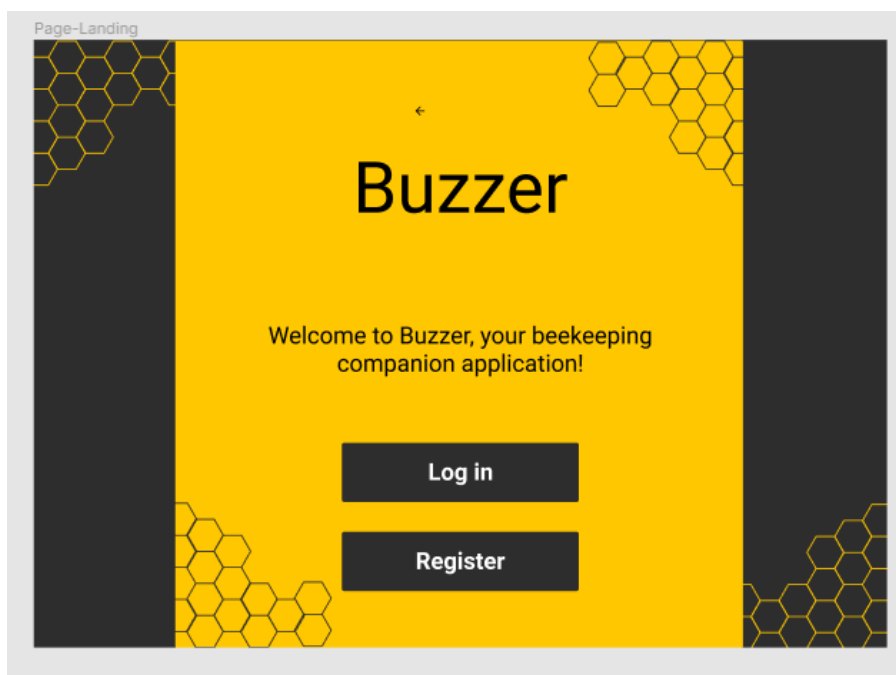
1. Sažetak	1
2. Prototip sučelja	1
3. Razrada funkcionalnosti	4
3.1. Use case i klasni model	4
3.2. Specifikacija API	6
3.2.1. Ruta GET /beehives	6
3.2.2. Ruta GET /beehives/:id	7
3.2.3. Ruta GET /tajna	7
3.2.4. Ruta POST /users	8
3.2.5. Ruta POST /auth	8
3.2.6. Ruta POST /beehives	9
3.2.7. Ruta PUT /beehives/:id	10
4. Korisničke upute	11
4.1. Home	11
4.2. My Grid	12
4.3. My Grid – adding beehives	12
4.4. My Grid – editing beehives	13
4.5. My Grid – Beehives list	14

1. Sažetak

Buzzer je web aplikacija namijenjena svim pčelarima koji žele digitalizirati i nadograditi svoje košnice. Pomoću naše aplikacije pčelar može pratiti vrijeme hranjenja pčela i daljinski pratiti temperaturu i aktivnost njihovih košnica. Naša će aplikacija imati vijesti i korisne informacije za starije pčelare, kao i savjete i trikove, korisne informacije pa tako i upute za ljude koji žele ući u predivan svijet pčelarstva.

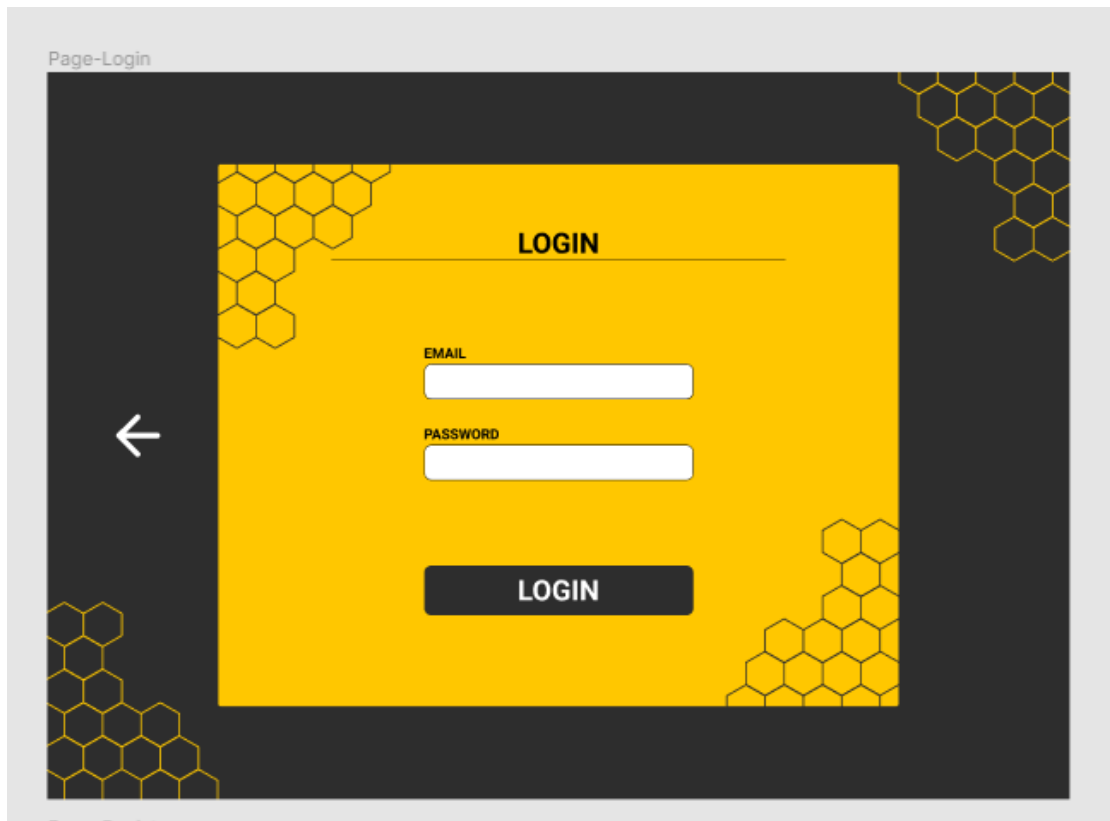
2. Prototip sučelja

Home (landing page) je u prototipu bio osmišljen ovako:

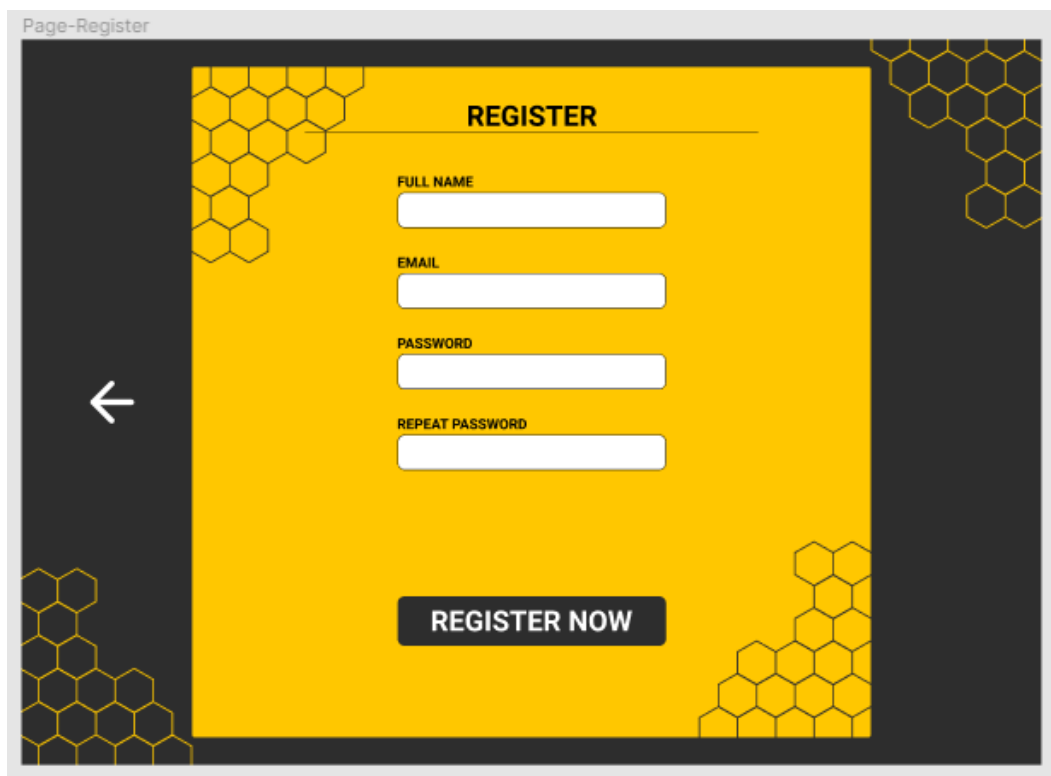


Slika 1 prototip landing page

Sa home stranice, posjetitelj može pristupiti; dijelu za prijavu (ukoliko ima već postojeći račun) ili registraciju

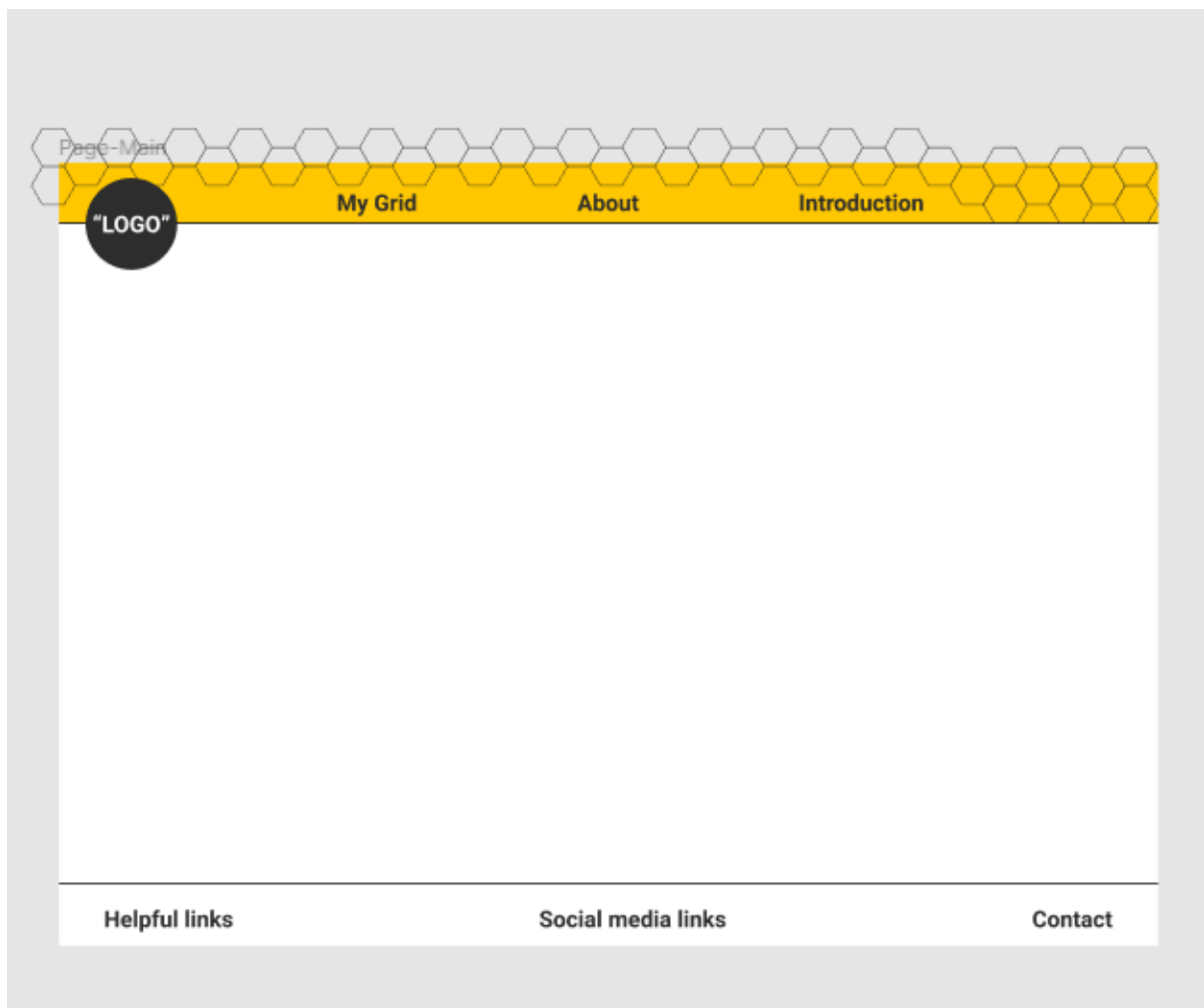


Slika 2 prototip login page



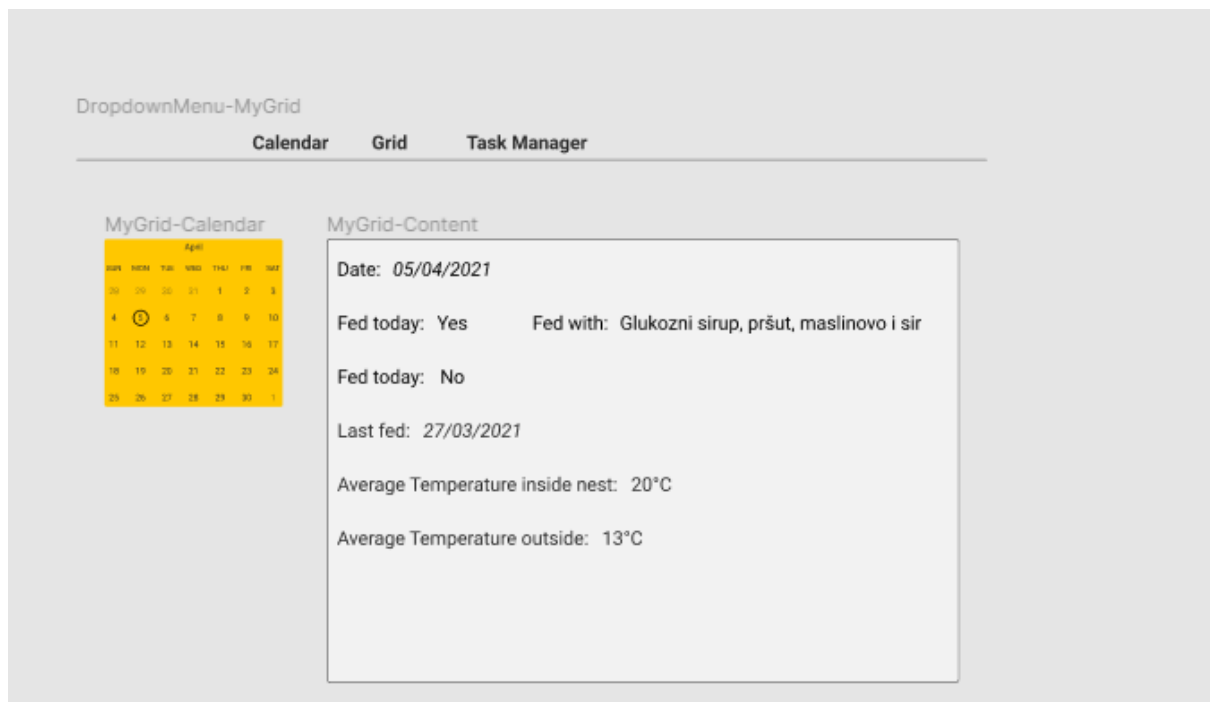
Slika 3 prototip registration page

Nakon prijave, osmislili smo da se korisniku pojavljuje mogućnosti biranja između 3 dropdowna (My Grid, About i Introduction)



Slika 4 protitip main page

Odabirom na "My Grid", zamislili smo da si korisnik može dodati svoje košnice u sustav, i tako voditi računa koja je temperatura u košnici pa tako i vanjska temperatura pa tako i kada ih je zadnji put nahranio



Slika 5 prototip myGrid

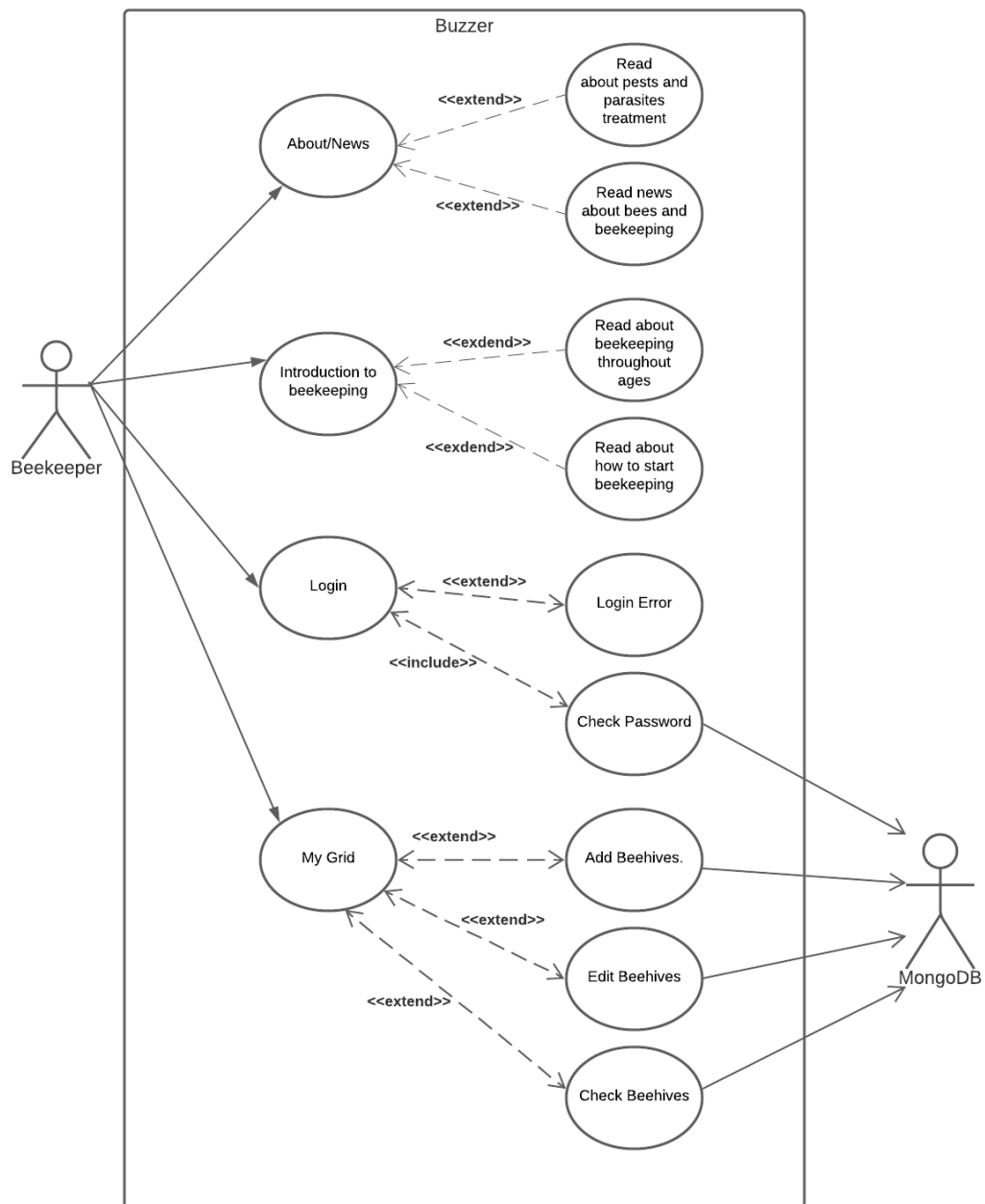
3. Razrada funkcionalnosti

3.1. Use case i klasni model

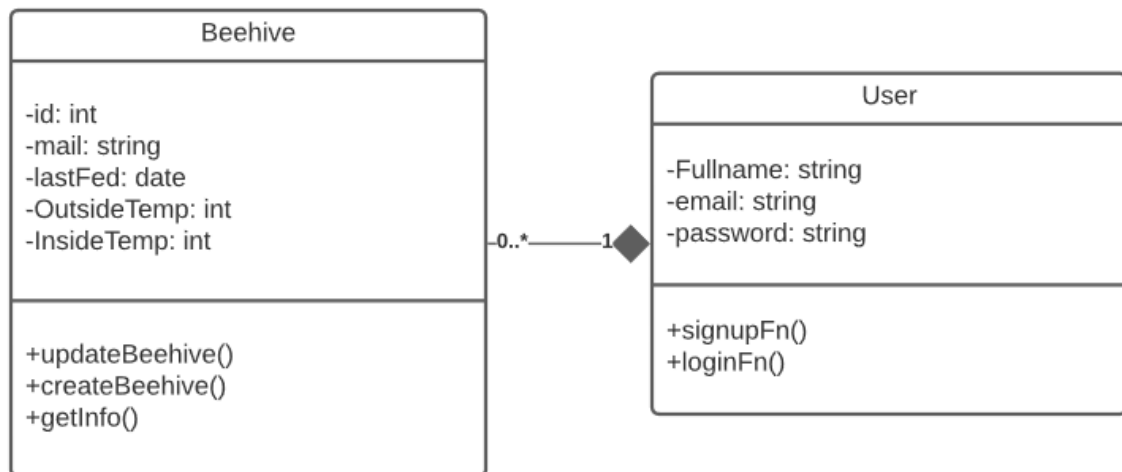
Nakon što se korisnik "ulogira" u svoji račun, može pregledavati razne elemente web aplikacije. Korisnik može na "My Grid" dodavati, uređivati i pregledavati (ime nastavka, kada ih je zadnji put nahranio i koja je vanjska/unutarnja temperatura) svoje košnice.

Naš backend sustav na koji je naša web aplikacija spojena zove MongoDB.

MongoDB se koristi za spremanje naših podataka pa tako i za autorizaciju korisnika pri registraciji i prijavi u sustavu.



Slika 6 buzzer use case



Slika 7 Buzzer class model

3.2. Specifikacija API

3.2.1. Ruta GET /beehives

```
app.get("/beehives", async (req, res) => {
  let db = await connect();
  let query = req.query;

  let selekcija = {};
  if (query.mail) {
    selekcija.mail = query.mail;
  }
  console.log(selekcija);

  let cursor = await db.collection("beehives").find(selekcija);

  let results = await cursor.toArray();

  res.json(results);
});
```

Slika 8 get/"beehives"

Ruta GET /beehives – povlači iz baze podataka sve podatke koji imaju u sebi mail ulogiranog korisnika

3.2.2. Ruta GET /beehives/:id

```
app.get("/beehives/:id", async (req, res) => {  
  let id = req.params.id;  
  let db = await connect();  
  
  let result = await db  
    .collection("beehives")  
    .findOne({ _id: mongo.ObjectId(id) });  
  
  res.json(result);  
});
```

Slika 9 get/"beehives:id"

Ruta GET /beehives/:id – povlači iz baze podataka jedan i samo jedan traženi objekt koji sadržava jedan naš postavljeni parametar.

3.2.3. Ruta GET /tajna

```
app.get("/tajna", (req, res) => {  
  res.json({ message: "Ovo je tajna " + req.jwt.mail });  
});
```

Slika 10 get/"tajna"

Ruta GET /tajna – koristimo ju za našu „tajnu“ tj za postavljanje tokena pomoću JWT-a.

3.2.4. Ruta POST /users

```
app.post("/users", async (req, res) => {
  // used to set user details
  // registracija
  let user = req.body;

  let id;
  try {
    id = await auth.registerFn(user);
  } catch (e) {
    res.status(500).json({ error: e.message });
  }
  res.json({ id: id });
});
```

Slika 11 post/"users"

Ruta POST /users - koristimo kako bi registrirali korisnika, dodijelimo varijabli podatke iz body-a te pozovemo auth.registerFn, koja se nalazi u drugoj datoteci, sa prosljeđenom varijablom. Nakon toga, registerFn registrira korisnika sa prosljeđenim podacima, ukoliko nije već registriran.

3.2.5. Ruta POST /auth

```
app.post("/auth", async (req, res) => {
  // login
  let user = req.body;

  try {
    let result = await auth.loginFn(user.mail, user.password);
    res.json(result);
  } catch (e) {
    res.status(403).json({ error: e.message });
  }

  res.send(user);
});
```

Slika 12 post/"auth"

Ruta POST /auth- koristimo kako bi ulogirali korisnika, dodijelimo varijabli podatke iz body-a te pozovemo auth.loginFn, koja se nalazi u drugoj datoteci, te prosljedimo mail i lozinku. Nakon toga, loginFn ulogira korisnika te mu dodjeli token.

3.2.6. Ruta POST /beehives

```
app.post("/beehives", async (req, res) => {
  // used to set beehive details
  let data = req.body;

  delete data._id;

  if (
    !data.nick ||
    !data.mail ||
    !data.lastFeedingDate ||
    !data.InsideTemperature ||
    !data.OutsideTemperature
  ) {
    res.json({
      status: "fail",
      reason: "incomplete",
    });
    return;
  }

  let db = await connect();

  let result = await db.collection("beehives").insertOne(data);

  if (result && result.insertedCount == 1) {
    res.json("all ok");
  } else {
    res.json({
      status: "fail",
    });
  }
});
```

Slika 13 post/"beehives"

Ruta POST /beehives – koristimo ju kako bi u našu bazu podataka spremili podatke vezane uz same pčelinjake.

3.2.7. Ruta PUT /beehives/:id

```
app.put("/beehives/:id", async (req, res) => {
  // used to set beehive details
  let id = req.params.id;
  let data = req.body;

  console.log(data);

  let check = checkAttributes(data);

  if (!check) {
    res.json({
      status: "fail",
      reason: "incomplete",
    });
    return;
  }

  let db = await connect();

  let result = await db
    .collection("beehives")
    .replaceOne({ _id: mongo.ObjectId(id) }, data);

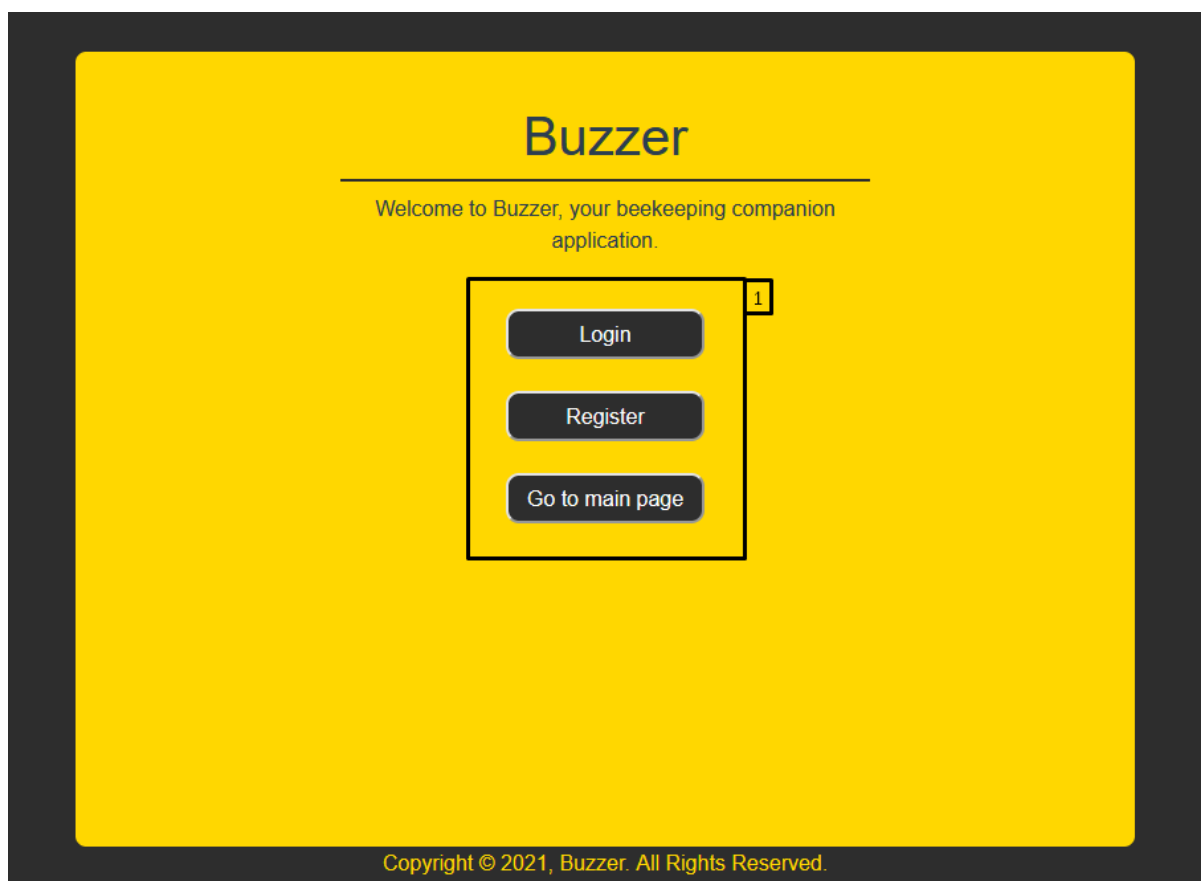
  if (result && result.modifiedCount == 1) {
    res.json({ status: "success" });
  } else {
    res.json({
      status: "fail",
    });
  }
});
```

Slika 14 put/"beehives:id"

Ruta PUT /beehives/:id – ova ruta služi kako bi izmjenili podatke vezane za jednu(1) datoteku u našoj bazi koju dohvaćamo preko id-a.

4. Korisničke upute

4.1. Home

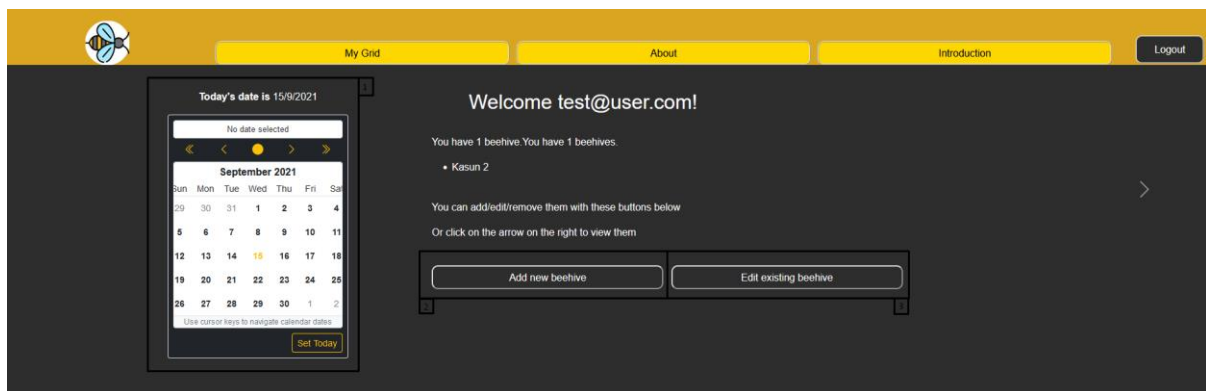


Slika 15 Buzzer landing guide

Pri ulasku u aplikaciju korisnik je suočen sa:

1. Register button – ukoliko korisnik nema račun, ovdje ga kreira
2. Login button – ukoliko korisnik već ima stvoreni račun, tu se prijavljuje

4.2. My Grid

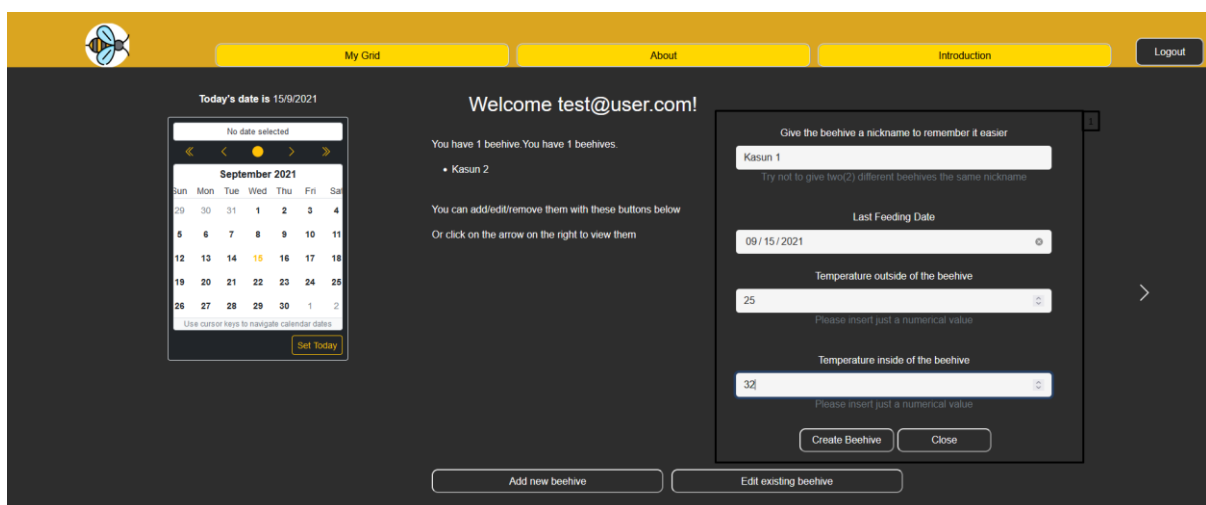


Slika 16 buzzer grid guide

Dolaskom na MyGrid, korisnik ima sljedeće opcije:

1. Kalendar – Pomoć oko datuma
2. Add new beehive – gumb za dodaju pčelinjaka
3. Edit existing beehive – gumb za uređenje postojećih pčelinjaka

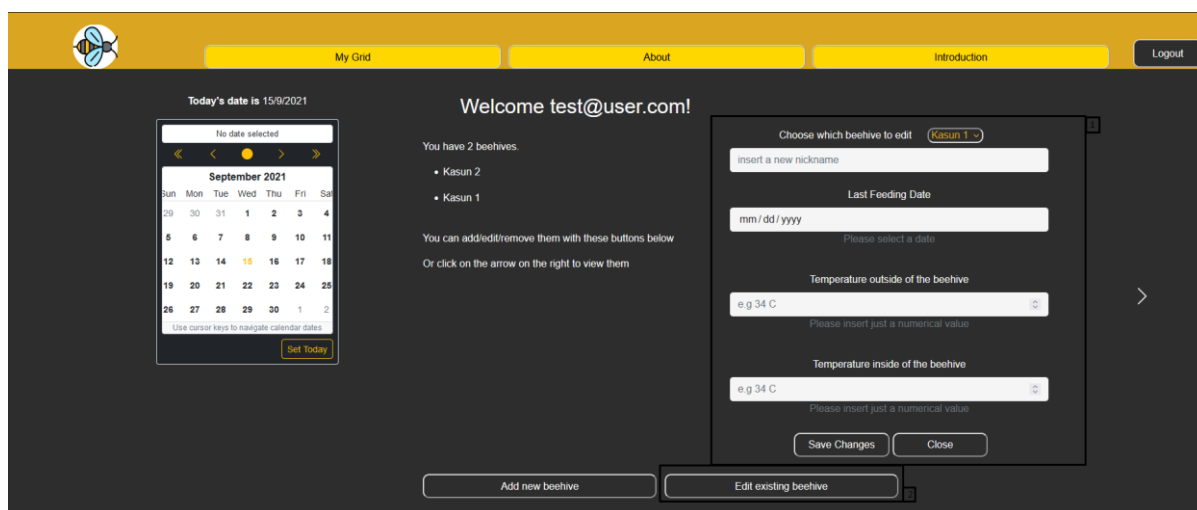
4.3. My Grid – adding beehives



Slika 17 add new beehive guide

Klikom na gumb "Add new beehive" korisniku se otvara sljedeća forma koja se može popuniti kako bi se dodao pčelinjak u naš sustav.

4.4. My Grid – editing beehives



The screenshot displays the 'My Grid' interface for editing beehives. The top navigation bar includes 'My Grid', 'About', 'Introduction', and 'Logout'. The main content area shows a calendar for September 2021, a list of beehives (Kasun 2, Kasun 1), and a form to edit an existing beehive. The form includes fields for nickname, last feeding date, and temperatures outside and inside the beehive.

Today's date is 15/9/2021

Welcome test@user.com!

You have 2 beehives.

- Kasun 2
- Kasun 1

You can add/edit/remove them with these buttons below
Or click on the arrow on the right to view them

Choose which beehive to edit: **Kasun 1**

insert a new nickname

Last Feeding Date

mm / dd / yyyy

Please select a date

Temperature outside of the beehive

e.g 34 C

Please insert just a numerical value

Temperature inside of the beehive

e.g 34 C

Please insert just a numerical value

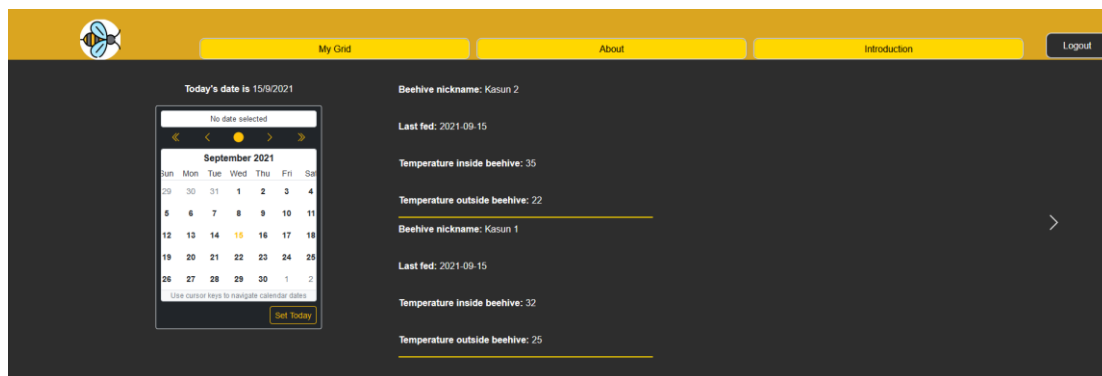
Save Changes Close

Add new beehive Edit existing beehive

Slika 18 edit existing beehive guide

Dodane košnice koje su u sustavu, korisnik može izabrati bilo koju košnicu pomoću imena koje im je dao, te ih može slobodno urediti ukoliko se pojavila neka greška u pisanju. Klikom na gumb "Save Changes", novi podaci se spremaju u sustav.

4.5. My Grid – Beehives list



Slika 19 beehive list guide

Klikom na desnu "strijelicu" dođemo do liste svih košnica koje je korisnik unio.