



# MissBug

## CRUDL end 2 end

### Node.js

## Phase 3 – User Support

### Model

The bug should have the following properties:

```
{
  "_id": "abc123",
  "title": "Cannot save a Car",
  "description": "errors when clicking Save",
  "severity": 3,
  "createdAt": 1542107359454,
  "creator": {
    "_id": "u101",
    "fullname": "Puki Ja"
  }
}
```

### Backend

- Add `user.json` – that holds all the users and a `userService`
- Add `auth` folder to your API
  - `/api/auth/signup` – add a new user to the file
  - `/api/auth/login` – check if username and password are correct - generate a `loginToken` and return a *mini-user* to the frontend
    - When bug is added – get the creator from the `loginToken`
    - Only the bug's creator can DELETE/UPDATE a bug
  - `/api/auth/logout` – clear the cookie
- Test your API from POSTMAN

### Frontend

- Use or Create the component: `<login-signup>`
- Add a `userService`
  - Implement the functions: `login`, `signup`, `logout`, `getLoggedInUser`
  - Use the `sessionStorage` to hold the `loggedInUser` and survive browser refresh

- Add a [user-details](#) page
  - This is a user profile page
  - Show the user's **bugs** (bugs that he has created)
    - Can you use your bug-list component?
  - At the header, add a [Profile](#) link that route to user-details page of the logged-in user.

## Implement ownership

- When adding a new Bug – add the creator (the `loggedInUser`)
- Only the bug's creator can DELETE/UPDATE a bug
- Use **postman** to test the APIs

## Add Admin Support

- Add `isAdmin` to the user entity
- Hard-coded mark a user (username: admin, pass: admin) as admin in your [user.json](#) file
- Admin is the only one who can use the CRUDL endpoints for the user entity (update, remove and read)
- Admin can delete / edit all bugs
  - Admin has a link to user-list page where he can view and delete users
  - Prevent deletion of users that own bugs

## DRY your backend – using middlewares

- Add a [logger](#) middleware and implement it where you see fit.
- Add a [requireAuth](#) middleware and implement it for all CRUD operations on the bug entity but LIST, and `getById`.

## Deploy to Render

Follow the needed steps to upload your project to [Render.com](#)

Set up a `SECRET1` environment variable holding the encryption key