

Janne Pennanen

537026

Web Applications - Course Project Documentation

1. Introduction

The aim for the course project was to create a web application for displaying code snippets. Users should be able to register and log in, furthermore, only logged in users should be able to post new code snippets and comment on existing ones. There should be a page listing all the code snippets and clicking on a snippet should display its comments.

A database should be utilized and backend should be implemented with Node.js. The use of content management systems is not allowed.

My implementation follows the guidelines.

2. Technology choices and description of the application

The back-end is implemented with **Node.js** and **Express**, while the front-end uses **React**. I decided to use these technologies since they were the ones used during the course, and I got comfortable with using them.

The landing page of the application displays a feed of all snippets, (or information about no snippets being posted yet) with options to register and login in the top bar. Clicking on the text of any of the snippets opens a snippet page which displays the snippet and its comments, if it has any. If the snippet has no comments, information about it is shown.

The register and login pages are simple, with just a form to enter credentials and an error container which displays any errors that the back-end sends, such as those related to an insufficient password when registering. The pages also contain buttons to navigate back to the feed page or to the login page.

When a user has logged in, the top bar only displays a logout button, which deletes the authtoken from local storage when clicked, thus logging the user out. For a logged in user, the feed page displays a form to post a snippet, and if the feed contains snippets posted by the user that's logged in, they display an edit button. Clicking on the edit button opens a text area in which the user can input a replacement for the posted snippet. A confirmation button is also shown and clicking on it confirms the editing and the database and feed are updated with the new snippet.

The single snippet page is also different for a logged in user. If the snippet in question was posted by the user that's logged in, it can be edited there in the same way as on the feed page. The page also displays a form to post comments about the snippet. If the comment list contains any comments posted by the user that's logged in, they have the same editing functionality as mentioned with the snippets.

3. Installation guidelines and user manual

After downloading the code from Github, run **npm install** in the root folder, server folder and client folder. Then run **npm run dev** in the server folder and **npm start** in the client folder. The application will then be running at localhost:3000. I was unsure if the project should've been setup with Docker, but there was no mention of it in the description and it works just as well like this.

Note: there are warnings about deprecations when running npm install in the client folder. For some reason running audit on those caused the code to not compile so I had to let those be.

User Manual

1. Registration:

Click the register button in the top bar to get to the registration page. On the registration page, enter a valid email and password and press register. Password requirements are shown on the registration page. The page shows information if the registration is successful / unsuccessful. Once the registration is successful, you can navigate to the main page or login page with the buttons.

2. Logging in:

Click the login button in the top bar or the “proceed to login” button on the registration page to get to the login page. Enter your credentials. The page shows information if the registration is successful / unsuccessful. Once login is successful, you can navigate to the main page by clicking on the button displayed.

3. Posting a snippet:

If you’re logged in, the main page shows a form to post a snippet. Enter a title and a snippet into the form and click Post. The snippet will be displayed at the bottom of the list. *Duplicate snippets are not allowed!*

4. Accessing the single snippet page:

You can access the single snippet page by clicking on the snippet text (not the title) of any snippet on the feed page.

5. Posting a comment on a snippet:

On the single snippet page, if you’re logged in, a form to post a comment is displayed. Input a comment and click post. The comment will be displayed at the bottom of the comment list. *Duplicate comments are not allowed!*

6. Editing a snippet

On the feed page or a single snippet page, any snippet that’s been posted by you (the user) displays an edit button. Clicking on it opens a field where you can type an edited snippet, and a confirmation button. Clicking on the confirmation button confirms the edit and the snippet will be updated.

7. Editing a comment

On the single snippet page, any comment that’s been posted by you (the user) displays an edit button. Clicking on it opens a field where you can type an edited comment, and a confirmation button. Clicking on the confirmation button confirms the edit and the comment will be updated.

4. List of features and points

The points I'm aiming for:

Feature	Points
Basic features with well written documentation	25
Users can edit their own comments/posts	4
Utilization of a frontside framework - React	5
Last edited timestamp is stored and shown with posts/comments	2
Total	36