Documentation

Installation quidelines

Recommended Node version for running the project is v19.1.0. It's the version that I was using myself.

To successfully install and run my project, you will need to download the project folder from my github, then unpack the folder.

Then, next step would be installing necessary node modules as it was recommended not to post node_modules folder on the desktop.

In my case it is enough to run "npm install" in the root folder, because it is a fullstack application.

(If error occurs and app doesn't run, then try running "npm install" in the client and server folder, but most probably it won't be needed)

To run my application, run command "npm run dev" in the root folder. The client and server sides will start running after that, you will see nodemon and react starting in terminal and browser with the website opening.

The server is running on port 1234, and the client is running on port 3000. Proxying is used.

User manual (instructions on the web page use)

So, the website is running, and now I will explain some of the functions:

No-login: if the user is not logged in, he can browse through the website.

You can go and log in/register by clicking user button on the top right side of the page.

You can go to the about page or to the home page by clicking the buttons on header on top of the page.

You can change the language of the page to Finnish or English by pressing FI or EN on the header new the user button.

Posts. You can view the post subjects on the main page. You can click on the author of the post and see his profile. You can also click the subject and the post will open.

Post. You can view the post content. Also you can click the post's and comments' authors and see their profiles.

You may comment the post after logging in

Registration. You can register by clicking the user button, then clicking “register”.

You have to enter desired username, email, password and you can check the box that says “Try admin account” if you want to try the admin features.

Note! The email needs to be legit for successful registration(at least 5 characters long, with @). Also, password has to be minimum 7 characters long.

After successful registration you will get to the login page, where you will need to login in order to access your account.

Login. For logging in, enter your email and password. You will be logged in after that.

Now you can access your profile and logout. For doing that, click on the user button on the header and click the desired function.

Note! For the successful logout, you need to click “go to the home page” after you clicked “logout”

Create a new post. Now after you are logged in, you can create your new post! Click on “Create post” on the header near and “about” button. You will be routed into a page where you enter information for your new post.

Note! “code language” is used only to determine if the code highlighter supports your code. If it doesn’t, it will alert you about it and post it anyway. The “code language” will not be seen in the post after you submit the form.

Commenting on existing posts. Now when you are logged in, you can open any post and comment it on the bottom of the page. Just write your comment in the box and submit it.

Editing or deleting your own post. You cannot edit or delete other users’ posts, but you can edit or delete your own posts. For doing so, open your own post, when click on “edit post” or “delete post”.

When editing post, just change information in the fields and submit.

When deleting post, just press “yes” on the “are you sure?” page.

Editing or deleting own comments. You can edit or delete your own comments. The procedure is the same as with posts, just open the post where you commented, find your comment, and click “edit comment” or delete comment.

Admin. If you checked “try admin account” when registering, it means that you can edit and delete all of the posts, and do same with the comments.

Technology choices and features.

In this project, I used many technologies. For the server side I used Express- generator for generating a server, and I created some get and post api requests possibilities by writing a corresponding code in the Index.js

As for the client side, I used React.js for creating a frontend application. Also I used MaterializeUI for better optimization of the interface.

For authentication I used JWT based authentication, because It seemed easier to use in this case.

For translations I used i18n library, because it's what the course recommends.

For code highlighting, I used react-syntax-highlighter, as the highlight.js recommended library wasn't able to detect the code languages in my case. The highlighter that I used is simple to use.

Features.

Here I have some features that I think worth giving points:

Basic features. Seems like I did all of the basic features that were asked to be done. 25 points

Users can edit their own posts/comments. This feature is present. 4 points

Utilization of React framework. 5 points

Code highlighting. Used react-syntax-highlighter. 2 points

Admin account. You can edit and delete all the posts. 3 points

Testing software for accessibility. I tested the software with Windows voice command, screen reader, keyboard and it appears to be working. It doesn't work perfectly good but it can be used. 3 points

User can click username and see user profile. I did that, but the problem is that I didn't have enough time to add user's register date and his bio, so I would expect maybe 1 point for that feature.

Last edited timestamp. I added last edited timestamp. When you edit the post/comment, the time changes. 2 points

Translation in two languages. 2 points.

My own feature would be an attempt to create some kind of web design in my application. From 1 to 2 points.

I should say that my app is poorly optimized for different screens afterall, because I had problems dealing with CSS (even though I tried materialize and used @media to optimize it for phone)

But the total number of points I’m aiming at is 47 points. I was aiming at this number of points because I’m trying to get grade 4 for this course, as I liked web programming a lot and I would like to have a good grade for it.

Even though my optimization is not good, I should say that the app works pretty well on the backend side, and (at least from my own testing) all of the backend functionality works.

Overall, I liked the project a lot. Thank you for this experience!