ProjectDocumentation

Technology choices

Project is full-stack, backend has been implemented using node.js and express. The frontend has been implemented using react and materialize UI.

Installation guidelines

Node version 19.6.0 is used for this project.

Download the .zip file for the project and extract the files to a folder of your choosing.

Open the folder in Visual Studio Code and navigate to the "Course Project-master" folder in the terminal.

All commands are given in the ".../Course Project-master" folder and can be found there in package.json.

Before running the app make sure to connect to "mongodb://127.0.0.1:27017/project" using for example MongoDBCompass.

Execute command "npm run install" then open another terminal in ".../Course Project-master"

User manual

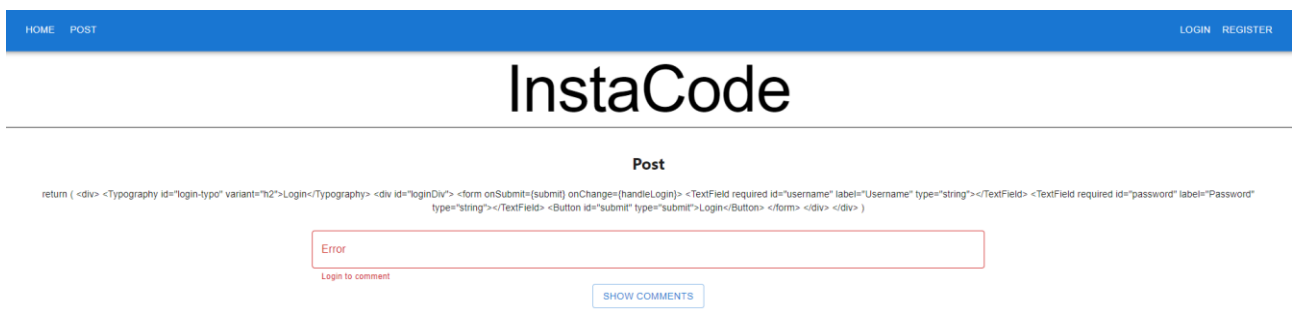**NOTE** If the page is refreshed the login no longer applies.

There is a header present on every page that has links to the home page, post page, login page and the register page.
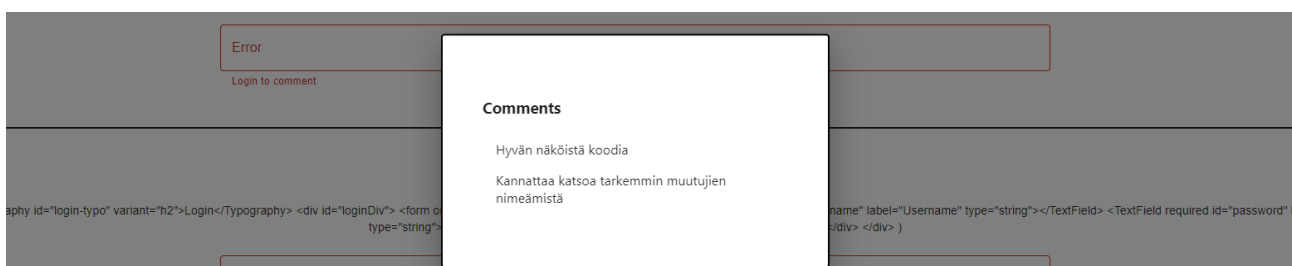

The home page has all the posts that are found in the database. If the database is empty the home page should look like this.

If there are posts in the database, it should look like this. There are also comment fields for each post. If the user is not logged in these will show in red and not work.

Comments can be seen by pressing the show comments tab. The page has to be refreshed for these to show.

Create post page has a text input with multiple lines and if you are logged in there is also a submit button.

Create post

Post

SUBMIT

Login page has two inputs that you can use to input the username and password and a login button. You should register before login in. **NOTE** the user is not notified if the login is a success or not. The login can be confirmed via the conditional functions.

Login

Username * | Password * | LOGIN

Register page works the same as the login page. You can create a new account here. The same **NOTE** applies here.

Register

Username * | Password * | REGISTER

Features

Feature	Points
Backend implemented with Node.js and Express	Mandatory
MongoDB has been used to keep track of posts, users and comments	Mandatory
Users can register and login	Mandatory
Users need to login to post or comment	Mandatory
Login implemented with JWT	Mandatory
Authenticated users can make new posts and comment on existing posts	Mandatory
Non-authenticated users can see posts and comments	Mandatory
The posts are listed on the index page and comments to a post can be opened with the "Show comments" button	Mandatory
App is responsive	Mandatory
App is documented	Mandatory
React used as frontside framework	5
App works with a screen reader, cannot be used with only a keyboard or only with voice commands	3
Points total	**33**