

Webapplications Project

Web applications

CT30A3203

Oskari Kivekäs

0570588

22.4.2021



Contents

1. Introduction.....	2
2. System Overview	2
3. Usage	2
4. Requirements implemented	3

1. Introduction

This project was the final assignment of web applications course. Program itself is a codepen like application where user can authenticate and post snippets of code (or whatever as text..), see other users profiles, and make comments on posts. Project is full stack application, containing front and back-end application folders. Front end was developed using React with React-Bootstrap and back-end with Node.js, Express and mongoDB, MERN stack in short.

Today (17.12.2021), project is still not finished, but a rough version of what it will be with more time. Though it should have been expected, I didn't reserve enough time for this, leaving half completed features and unfinished look in the project. On a bright side, this was my first actual web application, and I have learned a lot.

ps. Project will be finished, just not in this course timeline.

2. System Overview

Backend was done with Node.js and express framework, Node being a mandatory and both being familiar to me. MongoDB works as a database which I absolutely prefer over anything else in small projects like this. NoSQL syntax, and simple queries made it an easy choice (not a fan of sql). For authentication, api uses passport-jwt library, though this feature is "lightweight" implementation of an actual production ready authentication method.

On the front-end, I decided to give React a go. Not being very experienced with responsive design (or any ui for that matter), I also used bootstrap to help with the layout and save time building the interface. React handles routing and context management also, but when restricted actions are attempted, last check will be done by the passport in back end.

3. Usage

Readme.md has basic step by step operations how to get the program running in dev mode (actual build is not yet done). React dev server runs in port 3000 and back-end server in 5000. React has the proxy address in its package file and cors is configured in server. You need to open two different shells and run these programs separately.

4. Requirements implemented

Feature	Max points
Basic features with well written documentation	25
Users can edit their own comments /posts	4 → 2
Utilization of a frontside framework, such as React , but you can also use Angular, Vue or some other	5
User profiles can have images which are show next to posts/ comments	3 → 2
User can click username and see user profile page where name, register-date, (user picture) and user bio is listed	2
Last edited timestamp is stored and shown with posts/comments (stored on both, shown only on posts...)	2 → 1