Web Applications Project Work Documentation

Joona Ylijoki

Technology choises:

The code was written in Visual Studio Code using javascript, css, html (and pug). Some helpfull tools were used to ease the building of the application. These included node.js, express(-generator), materialize.css, MongoDB database, dotenv, bcryptjs, jsonwebtoken, express-validator and multer. The code was run using nodemon so that saving automatically started the new version of the code implemented in the application.

Application is only using port 3000 so that should be kept in mind when opening the application into the browser. The base of the application was generated using express and then the pages and functions were build on top of that generated base. Using of materialize helped make the pages nicer on top of the own css code that was written. Css was also used to make the application more user friendly for mobile device users. Information like users, posts and comments were saved to a mongoDB database. User login and registration used bcryptjs with password hashing. User authentication was created using jsonwebtoken and dotenv storing the secret to create the token. Token was stored in the localstorage which was implemented throught multer.

Installation guidelines:

To get the application running on your own computer follow the next steps:

1. Download the ZIP file from Github
2. Extract the downloaded file to a folder of your choice
3. Open Visual Studio Code
4. Select ”File” from the top left and then select ”Open folder” from the list
5. Find the right folder and open it to the Visual Studio Code
6. Open terminal from Visual studio code and write ”npm start” to the command window
7. The application is running when you get notified ” [nodemon] starting ’node ./bin/www’ ”
8. Then you can open the application in the web browser at URL: http://localhost:3000

User manual:

User has a list of links in the home page that he/she can use to move around in the web page. Some links are also available in the other pages.

Registration: User can register to the web page using the ”Register” link. In the register page the user have to come up with a username and password which is used to login the user in the future. If one would select already taken username, one will be notified to pick a different username or login. When the user has registered he/she will be taken to the login page.

Logging in: When the user has registered with a username he/she can go to ”Login” link and use the correct information to log in. After logging in the user is taken to home page. Logging in grants the user access to create posts and comment posts in the ”Content” page.

Posting and commenting: When a user has logged in he/she can create a new post on the ”Content” page. This can be done by pressing the button ”Add new post” which will take the user to a ”Add post” page where user can write his/her post. Then by pressing the button ”Add post” the post is saved and sent to the ”Posts” page behind the ”Content” link.

If a user is logged in he/she can also comment on the existing posts by writing his/her comment on the text field and pressing the button ”Add comment”. The comment is then saved and sent to the comments page behind the button ”Show comments”.

Reading posts and comments: Even if the user is not logged in to the page, one can use ”Content” link and read the posts on the posts page. Also the comments can be read in the comments page by pressing the ”Show comments” button below each post. Without logging in one can’t post or comment his/her own thoughts.

Logging out: Logged in user can logout by using the logout link in the home page. This will remove the token that validates the user and one cannot post or comment again before logging in.

List of features and number of points aimed at:

FEATURES IMPLEMENTED	**POINTS**
User register	
User login	
Non-authenticated users can see posts and comments	
There is a page listing all posts, user can open the comments page from the post	
Posting (while logged in)	
Commenting posts (while logged in)	
Authentication for users (JWT)	
Utilization of database (MongoDB)	
Documentation has been written	
Total points (aiming at)	25

Timetable: (what was done and when)
11.12.2021:
Creating the base with Visual Studio Code terminal commands
Installing some node modules
Started coding
Created register and login pages with pug
Included materialize to those pages
Created Users model with schema defining user information

Installed more modules
Created database connection with mongoDB
12.12.2021:
Created Comment model with schema defining comments
Created content page with list of all posts
Implemented login and updated it with authoratization (jwt) token
Created auth folder with middleware validateToken inside
Created posts page frame
Installed more multer
Created localStorage for token info
13.12.2021:
Updated login and register pages
Solving problems with authoratization
14.12.2021:
Fixed problems with authoratization while logging in
Solving problems with authoratization with posting
15.12.2021:
Solving problems with authoratization with posting
Updating posts page
Problem with authoratization was solved in Zoom exercise group
Changing posts page build
Updating posts page
Solving problems with showing posts on posts page
16.12.2021:
Solved problems with posts not showing up correctly
Created Subcomments model with schema defining comments to the posts
Added comments page
Edited the posts page to have buttons on posts that lead to comments
Added buttons to add comments
Created routes to lead to add comments page and show comments page
Constructed the comments page
Cleared some unnecessary files from the project
Commented the code
17.12.2021:
Created css files with some features modifying the pages so it looks nicer
Made the application more user friendly by applying own screen settings for mobile devices
Fixed some problems with the code
Wrote the documentation