

Description

I used the express-generator for creating the application and pug-view engine for creating the webpage files. I also used the materialize library to make the website's design easy to customize elements and make use of their library of icons. The functionality of the website is done using JavaScript including different JavaScript libraries. The user authentication is done using jsonwebtokens, where the jwt-token is generated when logging in and it is stored in the browser's cookies using the localstorage-library, and it can be removed from the cookies when logging out. The token eventually becomes invalid after a certain amount of time. The browser cookies are checked when the jwt-token is needed for authentication.

The website's database is done through mongoose. The database has 3 different models for users, posts, and comments. I kept the comments as a separate model from the posts as the comments had quite a bit of data of its own and including the data in just the post would make it quite big. Each of the comments can be linked to its own post by storing the post's unique ID. I used the MongoDBCompass application to keep track of what is being stored in the database and it works well if you want to check what is being stored. Mongoose keeps the database in the address 'mongodb://127.0.0.1:27017/project'.

How to start the website

The source code for the project can be found here: [GitHub](#)

1. Download the source code as a zip file from the above link and then unzip it to your preferred location. The created folder should be named Adv_web_Project-main but it can be renamed to something else.
2. Open the created folder inside a terminal or Visual Studio Code, and run the command 'npm install', which will download all the necessary libraries in the package.json file.
3. The server can then be started with the command 'npm run dev', which creates a nodemon of the webpage. The website should now be fully functional.

How to use the website

At the top of the page is the navigation bar, which includes links for the home-, register- and login-pages.

To be able to create posts and comments, you will need to create an account in the register page. In the register page, input the desired data into their respective input fields and make sure that the account info follows the guidelines stated above each input field.

After creating an account, you should be redirected to the login page, which you can also access from the navigation bar. Input the email address and password you input into the register page. After logging in, you should then be redirected to the home page, where you can create a post.

At the top of the home page is the input fields to create a new post. Type anything in these fields and press the submit button below to create the post. After submitting, you should be redirected to the post's own webpage, where you are able to edit and delete the post, and as see comments other people have made for your post and create your own comments. After creating a comment it will show up in the comment section and you can edit and delete it using the buttons on the right side. If you aren't the user that created the post, an upvote button should appear which can be toggled on and off if it was helpful. Non-logged in users can't see any buttons at all and can't interact with the comments.

If you want to create another account, you can press the logout button at the top and navigate to the register page again. Follow the same steps as before to create a new account and start creating new posts and comments using the newly created account.

Table of included features

Feature	Max Points
Basic features (as stated in the previous chapter) with well written documentation	25
Users can edit their own comments/posts	4
In total:	29