CT30A3204 Advanced Web Applications

Project: Web application for posting code snippets and commenting

Implementation

The project back-end is implemented with Node.js runtime environment with JavaScript. Express framework is used for routing and server-side logic. NoSQL database is running on 'mongoDB' with access help from 'mongoose' Object Data Modeling (ODM) library. User authentication is done through session authentication with the help of 'express-session'. User id is saved as a variable on login and checked when user tries to do actions that require authorization. User passwords are being hashed using 'bcrypt'. Front-end is built using Reach for user interface and component management, Materialize CSS for responsive style and element design. React router for routing and navigating in the application. Writing of the software was done with Visual Studio Code and testing was done using Postman, Google Chrome and Microsoft Edge browsers and MongoBDCompass. Stackoverflow.com and ChatGPT were used to help with troubleshooting and interpreting error messages.

Key features

The project is a web application that allows users to create, view, and comment on code snippets. Key features are 1) registration where users can register and log in to the application to gain access to full functionality. Logged in users can view a list of code snippets that have been shared by other users. 2) Code snipped management where code snippet includes a title in form of the author, authors id, snippet id and content information. 3) Adding comments to snippets. Users can leave comments through snippet display page or through detailed page where all comments under specific snippet are shown. Users can click on the author of listed snippet to take themselves to this detailed comment page. The comments have their own id, snippet id, content and author. The application is made to handle errors and display them to user when necessary. Some successful actions also display messages to make functionality clearer such as sending a comment or registering a user.

Front-end functionality

The application consists of 5 main pages and currently an empty index page. Top of the page displays a navbar for easy navigation of the application.

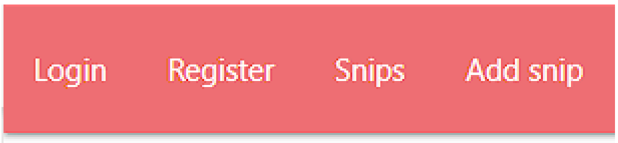


Figure 1. NavBar

Login takes user to login screen that uses "/login" route to access user database.

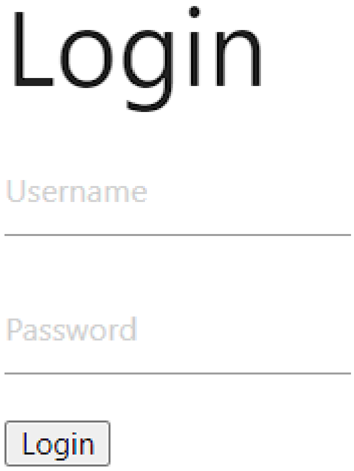


Figure 2. Login

"/register" route sends input information to server for user registration.

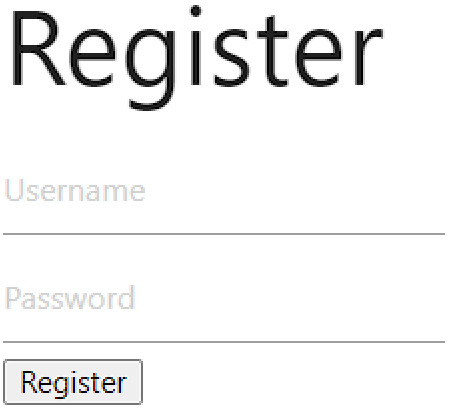


Figure 3. Register

Add snip takes user to “/addsnippets” page, lets user input a text and send it to the server for database storage.

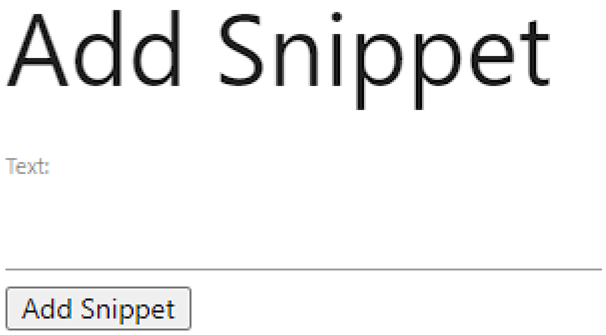


Figure 4. Add snippet

Snips takes user to “/snipptes”, fetches all the snippets saved to a database and displays them in a list. User can submit comments to a specific snippet in a text box under it on this page.

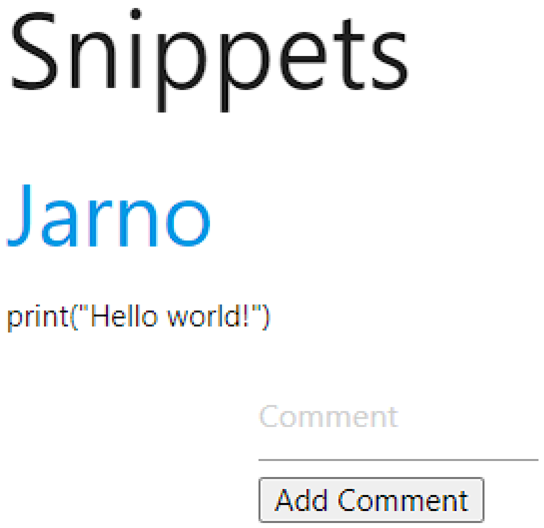
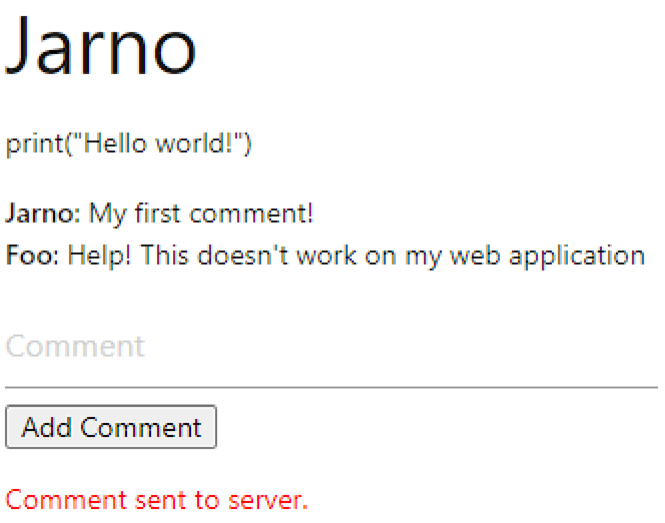


Figure 5. Snipptes

Lastly users can access snippet comments by clicking on the snippet title. This page displays comments and is another place for user to submit their own comment to a snippet. “Back to snippets” button takes back to main snippets page.



Jarno

print("Hello world!")

Jarno: My first comment!
Foo: Help! This doesn't work on my web application

Comment sent to server.

Figure 6. Commenting

New build to connect front-end with back-end can be created with npm run build in the client folder. Build version in '/client/build' should be included in the upload so this step might not be necessary.

```
cd client; npm run build; cd..;
```

The application can be launched by installing modules and starting. The application is running on localhost:3000. Please note the application needs mongoDB to work correctly and is configured to "const mongoDB = "mongodb://127.0.0.1:27017/testdb"" on app.js line:18.

```
npm install; npm start;
```

from the main folder. This launches the script from package.json:

```
 "scripts": {
   "start": "nodemon ./bin/www"
 },
```

CT30A3204 Advanced Web Applications

Based on the criteria I would argue the following scoring for the assignment :

Feature	Max points
Basic features (as stated in the previous chapter) with well written documentation	25
Utilization of a frontside framework, such as React	5