

Project Work Documentation:

List of features (target points):

Feature	Points
Mandatory requirements (basic app and docs)	25
Frontside framework utilization (React)	5
Usage of a highlighting library for code snippets (react-syntax-highlighter)	2
Pagination	2
Vote Buttons	3
Search	3
sum of points	40

Technology choices:

Server: Node.js (mandatory), Express (most familiar with this)

Client: React (useful tool for non-static web development with its wide range of supported packages)

Database: mongoDB (familiar and easy to use database structure)

Frontend styling: Materialize (easy to use and helps with the responsive design of the page)

Installation:

This project requires node.js and node package manager.

Client-side packages (cd client and npm install ...)	Server-side packages (cd server and npm install ...)
react and react-dom	bcryptjs
react-scripts	express and express-validator
react-router-dom	dotenv
react-syntax-highlighter	jsonwebtoken
	jwt-decode
	lodash.isempty
	mongodb and mongoose
	nodemon
	passport and passport-jwt

Run npm install on both the client-and the server-side, it should install atleast the packages listed above (required for the app to work as a whole)

Running the client: In the root folder, run "npm run client"

Running the server: In the root folder, run "npm run server"

User Manual:

To create any posts, user must register and login. Without logging in, user can view already submitted posts and comments but can't interact with them (vote, downvote, comment). Any user can also search for posts. If there aren't any posts submitted, then the pages are pretty much empty (because there isn't any template content).

Registration:

Email should be a valid email (xxxxxx@xxx.xxx is considered as valid)

Username should be unique

Password should be strong (at least 8 characters, must include upper and lowercase letters, numbers and at least one symbol)

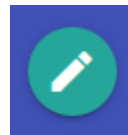
Login:

Done with the email with which the registration has been conducted

Password must match perfectly (as usual)

After logging in:

User can create their own posts through this button



Here user can create their code-snippet or plaintext post (Both textfields are required)

Create a Post

Title

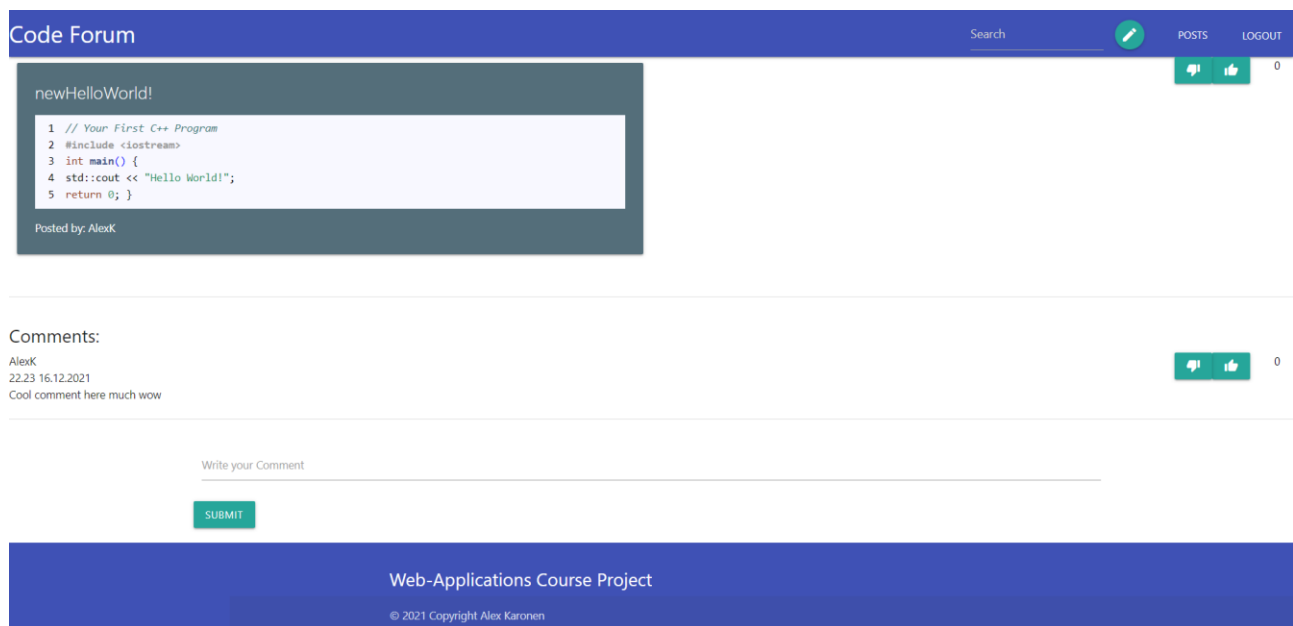
Write your Post here

SUBMIT

After submitting the post, the user is shown all the posts that have been created on the page.

(If there are over 10 posts -> the page is divided with pagination)

POSTS

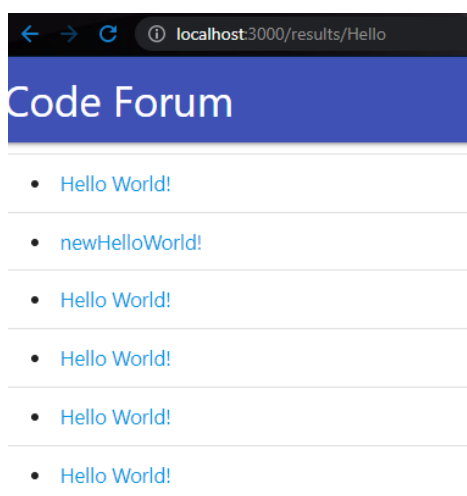


A post looks something like this, there's a card-panel showing the content of the post, then there are vote buttons on the right for the post and comments. Comments are shown only when there are comments for the post. Commenting and voting are disabled if the user hasn't logged in but comments and votes are shown if there are those.

SEARCH



Search returns posts that contain the search input when the user presses the "Enter"-key.



The search results are shown the same way as all the posts on the page like shown in the picture here.