KSHITIJ WEB DEVELOPMENT WORKSHOP

Assignment 4

Deadline: 03rd Aug 2021, 23:59:59

OVERVIEW

Now you all have already built in the frontend of your own Social Media Website, now we need to create a backend for your website, so that it turns out to be a real social Website. So now the next task will be first creating a backend for your website using NodeJS as the runtime environment, Express.js to create a server, and MongoDB as the DataBase. Not only that, you also need to wire up these things with your frontend code to make it a fully functional full stack website.

SECTIONS OF WEBSITE

These will be the major sections to work upon in the Assignment:

1. Signup Page
2. Login Page
3. The Main page.

DETAILS

1. SIGNUP PAGE: (75)

a. Backend:

1. You need to create a post route when the user signs up. The first task is to add validation to data you receive from the route. The second task is to also check whether there already exists a user with similar email credentials. (10)
2. The password should not be exposed in your database. Hence the third task is to encrypt the password. The next task is to initiate a new UserModel and post the data to the database. (20)
3. You can also add JWT authentication which will return a token. (This might not be of significant use on your website, as we don't really have a concept of protected routes in the website.) [BONUS] (5)

b. Wiring with Frontend:

4. Now assuming that there is already validation for all the input fields, you need to go ahead and then post the data to your api route with axios and then if it faces any issues render an error message, and in case of success you need to redirect them to the home page. (20)

c. BONUS TASK:

1. With the increase in the number of fake accounts on every platform the importance of unique username is growing. So now to implement this feature, that everyone has a unique username, you have to create a get route that accepts a username from params and then check from the database whether this username exists or not. (10)
2. Then in the frontend, every time the user types in the username you need to check whether that exists or not. And only allow the user to have a unique username. This can be done by useEffect hook where every time the username value changes you have to fetch and check whether the username exists or not.(10)

2. LOGIN PAGE : (60)

a. Backend :

1. You need to create a post route when the user signs in. The first task is to check if email exists. If it does then check the password (compare with the encrypted one). If there is an error again send back the error. (10)
2. You can also add JWT authentication which will return a token every time you login and add an expiry time. [BONUS] (5)
3. The third task will be to also add a logout functionality. (10)

b. Wiring with Frontend:

4. Again here we already have validation for all the input fields, you need to go ahead and then if it faces any issues, render an error message, and in case of success you need to redirect them to the home page. (20)

c. BONUS:

1. So if you receive a token from jwt, you can create a middleware function that will verify the token and then add this token to localStorage. So now after this token expires you can add the functionality to auto logout the user. (10)
2. You can also add the functionality to reset password. Try to watch a tutorial online to understand how this is done. (10)

MAIN PAGE : (105)

a. Backend :

1. The first task will be to create a post route when the user creates a new post. Again add validation and add the post to the database. (10)
2. Now you also have to display all posts on the main page. So create a get route for fetching all posts. (5)
3. The next two routes will be to update the post and to also delete the post. Here these routes will be only accessed by the author of the post. (20)

b. Wiring with Frontend:

4. Here you have to connect all your frontend UI to the proper functionality. Now all the buttons should be working, and you should be able to display the posts, update and delete your own posts, and also create new posts. (40)

c. BONUS TASK:

1. Now as earlier said all the buttons will be functional except the like button. So you can now add routes to like a post, and then also wire up the like button from the frontend to make it functional. This will work something like you have a likes array as a property on the PostModel and then whenever a user clicks on the button you send a request to update the likes array with the userId of the user who liked. (20)
2. Note that you cannot just increment the number of likes every time someone clicks without keeping a track of who liked it because the same user can login and logout many times to like the same post. (10)

RESOURCES

•MERN Stack Front To Back Full Stack React, Redux & Node.js

•Youtube:

- freeCodeCamp
- Nodejs tutorial in Hindi
- MERN STACK in Hindi 2021
- Full Stack MERN Project

P.S. You are free to follow other resources as well but analyze from time to time as to whether the content you are consuming is necessary at that point of time.

WORDS OF ADVICE :

- **IMPORTANT:** Make sure that you properly organise this entire MERN STACK project into different folders with proper segregations depending on their functionalities.

  For a detailed workflow and folder structure, you can refer to this link.

  For an overall idea of the file structure refer to this link.

P.S most of the “words of advice” given since the 1st assignment holds good here as well