CS385

Mobile Application Development

Group Project Division

Project Title: CS Student Social Web

Github link： *https://github.com/TymeoL1/CS385_Team_Project*

Project Idea Introduction:

The primary goal of this project is to enhance communication and mutual assistance among students in the same Computer Science major. We believe this idea aligns well with the theme of "Life Improvement."

For instance, the background image used in the main blog interface is a screenshot from the game *Dark Souls*. In my opinion, the process of people communicating and helping each other is akin to the protagonist's mission in the game—passing on the flame and carrying forward the spirit. Of course, the process of developing this project was as challenging and tormenting as the trials in the game itself :)

For instructions on how to start the project, you can check out our GitHub.

LI MEN (Student ID: 24250798 Email: li.men.2024@mumail.ie)

Contributions:

- Developed the login interface and implemented navigation.
- Implemented backend code and database design for user data and blog data.
- Managed data interactions between the backend and frontend.
- Enhanced the frontend interface design for improved user experience.

Technologies Used:

1. Axios:
    - Used to send HTTP requests to the server, like logging in users by sending their username and password.
2. UI Elements:
    - Input Fields: For users to enter their username and password.
    - Buttons: For actions like logging in and switching images.

3. CSS Styling:
    - Used external CSS files to make the UI look nice and consistent.
4. Component-Based Architecture:
    - Broke the app into smaller, reusable pieces, each handling a specific part (like the login page).
5. Form Handling and Event Management:
    - Managed user input and form submissions (like logging in or updating state when inputs change).
6. External JSON Data Source:
    - Sent and received JSON data to communicate with the backend (like sending login credentials).
7. React Router:
    - Used useNavigate from react-router-dom to move between different pages after a successful login.
8. Parent-Child Communication:
    - Passed down the setIsLoggedIn prop from the parent to update the login status.
9. Express.js:
    - Created the server and defined routes to handle HTTP requests.
10. CORS (Cross-Origin Resource Sharing):
    - Allowed the frontend to make requests to the backend.
11. MongoDB:
    - Used as the database to store user info and posts.
12. APIs:
    - Exposed APIs for login, adding posts, and fetching posts.
13. JSON Data Interaction:
    - Handled JSON data for requests and responses.

SONGYAN LAI (Student ID: 24250371 Email: Songyan.Lai.2024@mumail.ie)

Contributions:

- Designed and developed the blog interface.
- Created and integrated the calendar and weather functionalities.
- Implemented blog posting and search functionalities.

Technologies Used:

1. React Hooks:
    - useState and useEffect: These are used a lot to keep track of what's happening in the app and react to changes, like when fetch data or switch images.
2. Axios:
    - This is for making HTTP requests, like getting data from a server or sending new blog posts.
3. React Calendar:
    - Using a calendar component (react-calendar) that lets users pick dates. It's styled with some CSS to look nice.
4. API Integration:
    - using a weather API (open-meteo.com) to get weather data and a geocoding API (nominatim.openstreetmap.org) to turn location names into coordinates.
5. Conditional Rendering:
    - This means showing different parts of the app based on certain conditions, like switching between the search page and the blog posts.

6. Component-Based Architecture:
    - The app is split into multiple components like Header, Sidebar, PostArea, and Post, which makes the code more organized and easier to manage.
7. Form Handling and Event Management:
    - This is about managing user inputs and actions, like when you submit new posts or add comments.
8. Styling with CSS:
    - Using CSS to style the components and make everything look good. Some styles are in a CSS file, and some are inline.
9. Data Interaction:
    - Managing how data moves between the frontend (what the user sees) and the backend (the server) to make sure everything works smoothly.