

Assignment Title: Building a Todo List App with User Authentication (Using Local Storage)

Assignment Description:

In this assignment, you will create a Todo List web application with user authentication features. Users should be able to sign up for an account, log in, and manage their tasks. Instead of a backend server and database, you can use local storage to simulate user accounts

Requirements:

1. User Authentication

Implement user registration (signup) and login functionality.

Users should be able to sign up with a username and password.

Users should be able to log in with their credentials.

Use forms for both registration and login.

2. Authentication State

Manage user authentication state using React state.

Implement protected routes to ensure that only authenticated users can access certain parts of the application (e.g., the todo list).

3. Task Management

Users should be able to create, edit, mark as completed, and delete tasks.

Tasks should be associated with the logged-in user.

Tasks should persist in local storage, so they are not lost on page refresh.

4. Logout

Implement a logout button that allows users to log out of their accounts.

5. User Interface

Create an intuitive user interface with a clean design for both the authentication and todo list parts of the application.

Display a list of tasks, each with options to edit, mark as completed, and delete.

6. Error Handling

Provide user-friendly error messages for scenarios such as incorrect login credentials, registration errors, or failed task updates.

Security Considerations:

7. User-Specific Data

Ensure that each user can only see and manage their own tasks after logging in.

Tasks should be associated with the user who created them.

Bonus Features (Optional):

- Add due dates to tasks.
- Implement task prioritisation (e.g., high, medium, low).
- Add the ability to filter tasks by completed and active.
- Implement sorting by due date or priority.

Database Integration (Optional):

- If you want to take it a step further, integrate a backend server and database (e.g., Node.js with Express and MongoDB) to store user accounts and tasks securely.
- Implement API endpoints for user registration, login, and task management.

Submission:

- Submit your code via a GitHub repository that includes both the front-end React code and, if applicable, the back-end code.
- Provide clear instructions on how to run the application and any necessary setup for the database (if used).
- Include a README explaining the authentication flow and any additional security measures you implemented.