

INTERNET WEB SYSTEMS-1

Project Report: To-Do List App

PRATHAP RAMACHANDRA

(#02129543)

Features Completed:

1. Server Configuration:

- The `server.js` file is set up, likely handling the server-side logic using Express.js.

2. Frontend Setup:

- The **public** folder contains `index.html`, `style.css`, and `app.js`, indicating that the basic structure for the frontend of the To-Do List app is in place.
- `index.html` provides the main HTML layout.
- `style.css` handles the app's styling.
- `app.js` likely manages the frontend logic for adding, deleting, and interacting with tasks.

3. Database Integration:

- The **config** folder contains `db.js`, which likely handles MongoDB connections, confirming the MongoDB integration.

4. Task Model:

- The **models** folder includes `taskModel.js`, which defines the schema for tasks, outlining how tasks are structured in the database (likely attributes like task name, completion status, and timestamps).

5. Task Routes:

- The **routes** folder has `taskRoutes.js`, which likely manages the routing for creating, reading, updating, and deleting tasks (CRUD operations).

Pending Tasks:

1. Frontend Functionality:

- The app may need additional JavaScript functionality (in `app.js`) for handling user interactions such as editing tasks, marking them as complete/incomplete, and real-time updates on the UI.
- 2. **Error Handling:**
 - Ensure proper error handling is in place on both the server and client sides, especially for database operations.
- 3. **User Authentication:**
 - If the app is intended to have multiple users, user authentication (e.g., login/signup) is missing. This could involve adding authentication middleware and frontend logic to manage user sessions.
- 4. **Styling Enhancements:**
 - While basic CSS is in place, the UI could benefit from more sophisticated styling for better user experience and aesthetics (e.g., adding animations, responsive design).
- 5. **Deployment:**
 - The app needs to be deployed to a cloud platform like Heroku or Vercel, with the MongoDB database hosted on a service like MongoDB Atlas.

Plan for Completion:

1. **Complete Frontend Logic:**
 - Finish writing the JavaScript code in `app.js` to ensure all user interactions (adding, deleting, editing tasks) are functioning smoothly.
2. **Implement User Authentication:**
 - Add authentication features using Passport.js or JWT for secure user login/signup.
 - Modify `taskModel.js` and `taskRoutes.js` to associate tasks with specific users.
3. **Enhance Styling:**
 - Improve the visual appeal of the app by refining the `style.css` file with better layouts, fonts, and responsiveness.
4. **Error Handling & Testing:**
 - Test the app thoroughly, handle edge cases, and ensure server-client communication is robust. Add error messages for failed operations (e.g., database errors, validation errors).
5. **Deployment:**
 - Once all functionality is implemented and tested, deploy the app to a cloud platform, ensuring the database connection works in the production environment.