Full Stack Web Development Project

Project Title: To-Do List Web App

What is This Project About?
This project focuses on building and deploying a full-stack Todo List web application using React.js for the frontend and Node.js with Express and MongoDB for the backend. The goal is to provide hands-on experience in designing, developing, and deploying a complete MERN (MongoDB, Express, React, Node) stack application.

Purpose:
- To apply the concepts learned in the classroom in a real-world project.
- To understand the process of deploying a full-stack application to platforms like Vercel, Render, or Railway.
- To demonstrate the ability to connect frontend and backend effectively.

Learning Outcomes:
- Implement CRUD (Create, Read, Update, Delete) operations with REST APIs.
- Connect React frontend with Node.js backend.
- Use MongoDB for data persistence.
- Learn the basics of deployment.
- Improve UI/UX skills and debugging techniques.

Skills Developed:
- Frontend development with React
- Backend development with Node.js & Express
- API integration
- Database management (MongoDB)
- Full-stack deployment
- Version control with Git and GitHub

What You Have to Do (Step-by-Step):

1. **Plan the Project:**
- Decide on the features (Add, Delete, Edit, Mark as Done)
- Create wireframes or a rough UI layout.

2. **Develop the Backend:**
- Set up Node.js with Express.
- Connect to MongoDB using Mongoose.
- Create RESTful APIs for CRUD operations.

3. **Develop the Frontend:**
- Set up the React project.
- Create components for todo input, todo list, and edit functionality.
- Connect the frontend to the backend using Axios or Fetch.

4. **Deploy the Application:**
- Deploy the backend to Render/Railway or similar.
- Deploy the frontend to Vercel/Netlify.
- Make sure the deployed frontend is successfully communicating with the deployed backend.

5. **Create a Demo Video:**
- Record a 2–3 minute video demonstrating:
- Website overview
- Adding, updating, deleting todos
- Deployment URLs
- Key code explanation (brief)

What to Submit:

- *GitHub Repository Link* (with separate folders for client and server)
- *Live Deployed URL* for both Frontend (e.g., Vercel) and Backend (e.g., Render)
- *Demo Video Link* (Uploaded to Google Drive or YouTube, make sure it's accessible)
- *README File* in the repository with Project Description, Technologies Used, Setup Instructions and Deployment Links

Tips for Students:

- Keep your UI clean and responsive using basic CSS or frameworks like Tailwind or Bootstrap.
- Use .env files to secure your database and API keys.
- Ensure CORS is properly configured in your backend for frontend-backend communication.
- Test your API using Postman before integrating with frontend.
- Use screen recording tools like OBS or Loom for creating your demo.
- Use meaningful commit messages and maintain clean code structure.
- Make sure your video has clear audio and visuals—explain each part briefly but confidently.