

Step-by-Step Documentation for To-Do List Application

Objective:

Develop a To-Do List application that allows users to add, edit, delete tasks, set due dates, times, priorities, and receive reminders.

Technologies:

- **MongoDB:** Database for storing tasks.
- **Express.js:** Backend framework for handling server-side operations.
- **React.js:** Frontend library for building the user interface.
- **Node.js:** JavaScript runtime environment for executing backend logic.

1. Initial Setup

.1. Install Required Software:

1 Node.js and npm:

Install Node.js from the official website: [Node.js Download](#)

Verify installation by running:

```
node -v  
npm -v
```

2 MongoDB:

Install MongoDB from the official website: [MongoDB Download](#)

Verify MongoDB installation by running:

```
mongod --version
```

3 React.js:

React is a JavaScript library, so no separate installation is required.

2. Backend Setup with Node.js and Express

Open your terminal in the project directory and initialize a new Node.js project:

```
npm init -y
```

This creates a package.json file that will manage your project's dependencies.

2.2. Install Dependencies:

- **Install the necessary libraries for the backend:**
 - Express: Framework for building the server.
 - Mongoose: ODM (Object Data Modeling) library for MongoDB.

- Cors: Middleware to handle cross-origin requests.

`npm install express mongoose cors`

2.3. Setup Express Server:

- Create a `server.js` file in the root directory.
- In this file, you'll define routes for adding, editing, deleting tasks, and connecting to MongoDB.

2.4. MongoDB Connection:

- Set up a connection to MongoDB using Mongoose.

3. Frontend Setup with React.js

3.1. Create React Application:

- Create a React application within your project directory using the following command:

`npx create-react-app client`

This creates a client directory with all the necessary files and configurations for a React app.

3.2. Install Frontend Dependencies:

- Navigate to the client directory and install additional libraries:
 - **uuid**: For generating unique task IDs.
 - **axios or fetch**: For making HTTP requests to the backend.

3.3. Create Components:

- In the `src` folder of your React app, create components like `TodoForm.js` and `TodoList.js`.
- These components will handle task creation, editing, and display.

3.4. Connect to Backend:

- Use `axios` or `fetch` in React to make requests to your Express server, enabling task CRUD operations.

4. Database Setup with MongoDB

4.1. Define Schema:

- use Mongoose to define a schema for tasks, including fields like title, due date, time, priority, and reminder.

Start MongoDB:

- Ensure MongoDB is running on your system. If using the local MongoDB server:

Mongod

5. How to Run To-Do List application

- Navigate to the Directory where To-Do Application Server source code located. i.e
PS C:\Users\Latharani\Documents\TODO Application > cd server
- Run the Below Command
node server.js
- Navigate to the Directory where To-Do Application client source code located. i.e
PS C:\Users\Latharani\Documents\TODO Application > cd client
- Create React App in This Directory using Below Command
npx create-react-app .
- Navigate to the Directory where To-Do Application source code located. i.e
- PS C:\Users\Latharani\Documents\TODO Application \client\app> cd src
- Run the Below Command
- npm install
npm start
- Run the blow line in Chrome Browser(This point is optional).
- <http://localhost:3000/>

5.1. List of the Task Performed on Front-end

- Add Task
- Edit Task
- Delete Task
- View Task

5.2. Task CRUD Operations:

- Add Task: Allow users to create new tasks with details like title, due date, and priority.
- Edit Task: Enable users to modify existing tasks.
- Delete Task: Provide functionality to remove tasks.
- View Tasks: List all tasks, displaying key details like title, due date, and priority.

5.3. Task Prioritization:

- Implement a system where tasks can be marked with different priorities (High, Medium, Low).

- The user can set the priority when creating or editing a task.

5.4. Reminders and Notifications:

- Add a feature where users can set reminders for tasks based on their due dates.
- These reminders will trigger notifications or logs in the console (extendable to email or push notifications)

6 How To-Do Application Looks on Front-end

To-Do List

Add Task

Morning Standup Meeting

2024-09-02
23:58

High

Reminder: 9/2/2024, 11:00:00 PM

CompleteEditDelete

Coding

2024-09-02
23:58

High

Reminder: 9/2/2024, 12:30:00 PM

CompleteEditDelete

Automated Test Cases Dev/Review Test Cases For Current Sprint

2024-09-03
00:00

Medium

Reminder: 9/3/2024, 1:30:00 AM

CompleteEditDelete

Regression Testing

2024-09-03
00:00

Medium

Reminder: 9/3/2024, 2:01:00 AM

CompleteEditDelete

