# Full Stack To-Do List Application

**Overview**

This project is a full-stack web application that allows users to create and manage a personalized to-do list. It uses **React.js** for the frontend and **Node.js with Express** for the backend, with **MongoDB** serving as the database for user data and tasks. The application includes a user authentication system (sign-up and sign-in) and provides a personalized experience for each user.

**Technologies**

* **Frontend**: React.js (Single Page Application)
* **Backend**: Node.js with Express (JavaScript)
* **Database**: MongoDB
* **Authentication**: User authentication using JWT with personalized tasks
* **Deployment**: Docker for containerization (optional) or any cloud service

**Project Features**

1. **User Authentication**: Users can sign up and log in using email and password, with JWT-based authentication for secure access.
2. **Personalized To-Do List**: Each user will have their own list of tasks.
3. **CRUD Operations**: Users can **Create**, **Read**, **Update**, and **Delete** tasks.
4. **API-Driven**: The application uses a REST API to communicate between the React.js frontend and the Node.js backend.
5. **Single Page Application (SPA)**: Frontend interaction with minimal page reloads for smooth user experience.

**Project Structure**

**Frontend (React.js)**

The frontend renders the user interface, manages routes, and sends API requests to the backend.

* **React Components**: Manage different UI elements (e.g., TaskList, TaskItem, AuthForm).
* **React Router**: Manages client-side routing between login, signup, and task pages.
* **Axios**: For sending HTTP requests to the backend for authentication and task management.

**Backend (Node.js with Express)**

The backend serves the API, handles user authentication, and stores user data and tasks.

* **Express.js**: Handles API requests (GET, POST, PUT, DELETE) for the to-do list and user accounts.
* **JWT (JSON Web Token)**: Used for secure user authentication.
* **MongoDB with Mongoose**: Used as the database for storing user information and tasks.

**Database (MongoDB)**

* **User Data**: Email, password (hashed with bcrypt), and session information stored securely.
* **To-Do Tasks**: Task name, status (complete/incomplete), and timestamps for each user.