

# DOCUMENTATION FOR FRONTEND

## **A. Title:**

**Assignment Title:** Better Associate Software Engineer (Python React)

**Assessment** – Frontend

**Name:** Khan Mohammed Ammaar

**Email:** khanmohammedammr@gmail.com

**Date:** August 7, 2025

## **B. Introduction:**

This document describes the frontend portion of the assessment. The React application was developed to interact with the Flask backend to perform CRUD operations on tasks and comments, presenting a simple and user-friendly interface.

## **C. Environment Setup:**

- **Operating System:** Windows 10
- **Node.js Version:** 20x
- **Package Manager:** npm
- **React Version:** 18.x
- **Other Libraries:** axios for HTTP requests (used optionally), react-router-dom if used

## **Setup Steps:**

- Navigated into the frontend directory (client).
- Installed Node.js from the official site.
- Ran npm install to install frontend dependencies.

## **D. Project Structure:**

**/client**

**|-- package.json**

**|-- src/**

**|-- components/**

**|-- App.js**

**|-- index.js**

**|-- node\_modules/**

- **package.json**: Contains frontend dependencies and scripts.
- **src/components/**: Contains React components like TaskList, TaskForm, CommentList.
- **App.js**: Main React component managing routes and state.
- **index.js**: Entry point of the React application.
- **node\_modules/**: Folder containing installed npm packages.

## **E. Solution Approach:**

### **Frontend (React):**

- Created React components for listing tasks, adding, editing, and deleting tasks and comments.
- Used fetch or axios to call backend Flask APIs asynchronously.
- Managed component state using React hooks such as useState and useEffect.
- Included error handling and provided user feedback on UI actions.
- Manually tested all CRUD operations to ensure smooth functionality and integration with backend.

### **Assumptions:**

- The UI was kept simple focusing on usability and clarity.
- Assumed backend API routes and payload structures as specified by the backend implementation.

## **F. How to Run the Backend:**

### **1. Navigate to the frontend folder inside your project:**

cd client

### **2. Install frontend dependencies:**

npm install

### **3. Run the React development server:**

npm start

4. The frontend UI will open automatically at <http://localhost:3000> in your default browser.

## **G. Screenshots:**

<C:\Users\Khan Mohammed Ammaar\OneDrive\Pictures\Screenshots 1>

## **J. Summary:**

The React frontend successfully communicates with the Flask backend to provide complete CRUD functionality. This project enhanced my understanding of React hooks, component design, and working with REST APIs. In future versions, I plan to enhance the UI styling and add more extensive form validations for better user experience.