# CareerPulse Source documentation:

## 1. Introduction :

Applying for job opportunities, especially internships, can be quite challenging. It's a time-consuming task, and the pressure of referrals and deadlines adds to the stress. Our app simplifies this process, enabling you to efficiently track and control your job applications without the need for complex Excel sheets.

Our application has the capacity for multifaceted utility, as it can serve both as a job tracker and a project management solution. It offers the flexibility to create dynamic boards and columns, making it an ideal tool for tracking updates and progress.

In the context of job tracking, individuals targeting particular roles can initiate custom boards with role-specific names. This enables them to efficiently manage their applications across various companies. They can easily monitor the stage of their applications in each company, simplifying the job application process. Users can effortlessly expand this approach by creating new boards for different roles, streamlining their job hunt.

Furthermore, our application is not limited to job tracking alone; it also seamlessly integrates project management capabilities. Teams can establish boards for their specific projects, allowing them to oversee and manage their tasks collectively. The option to add new columns enhances their ability to adapt to evolving project requirements, whether that involves introducing new tasks, modifying existing ones, or even removing tasks when necessary. This adaptability makes it an excellent choice for agile project management.

This report provides users with an extensive project summary, enabling them to perceive it as open-source software and contribute enhancements. Additionally, the report aids developers in grasping the code and acts as an initial reference for the project.

## 2. Tech Stack Used :

The development process relied on specific technologies, and it is advisable for the next team of developers working on this project to have these technologies installed and operational in advance to have a smooth experience.

1. FrontEnd:
    a. React
    b. Node.Js
    c. CSS
2. Backend:
    a. Python, Flask
    b. Pytest (Test Framework)
3. Database:

a. MongoDB

## 3. Code Functionalities :

We have implemented enhancements in both the frontend and backend. The following is the explanation of the changes incorporated.

**Front-end source files :**

**App.js -**

App Component is the main component in React which acts as a container for all other components. This is the root component which calls the child components.

**Instance variables :**

1. `*auth*`: It keeps track of the user's authentication status.
2. `*boardData*`: It stores the data related to boards.
3. `*currentPage*`: It represents the current page or component to be displayed.

**Functions Implemented in App.js :**

- *handleLogout*: This function handles the user's logout action. It removes the authentication token stored in the local storage, sets the auth state to false, and changes the current page to the login page (LoginPage).
- *useEffect with the AJAX request:* This useEffect is used to make an AJAX request to the server to fetch the board data. It sets the boardData state based on the response from the server.
- *useEffect for user authentication:* This useEffect is used to check if the user is authenticated by looking for an authentication token in local storage. If a token is found, it sets the auth state to true and changes the current page to the ApplicationPage with the initial data.

**headerDropDown.js** -

The HeaderDropDown component is a dropdown menu that provides access to boards, the ability to create new boards, and a dark mode toggle.

**Home.js -**

Defines a React component that includes a sidebar, columns, and a new column creation option and uses Redux to manage the application's state, such as boards and columns. The component dynamically adjusts the layout based on the window size and user interactions.

**Login.js -**

Defines a React component for logging in and signing up. It manages the form data, handles the authentication process through API calls, and provides a simple interface for switching between login and signup forms.

**Header.js** -

Defines a "Header" component that works for navigation and user interaction to manage boards, tasks, and other related actions.

**Task.js**:

Defines a React component that renders individual task cards within a specific column of a board. It is able to handle drag-and-drop functionality and allows users to view and interact with task details through a modal when clicked.

**emptyBoard.js:**

Defines a React component to display a message and a button when there is no content to show, specifically when there are no boards (or columns) available.

**BoardSlice.js -**

The boardsSlice reducer manages state for Kanban boards. It handles actions to fetch initial data, create/edit boards and columns, add and edit tasks, update task status and subtasks, drag tasks between columns, and delete boards and tasks. The reducer dispatches async thunks to persist changes to a backend API. Overall boardsSlice provides Redux actions and reducers to manage complex Kanban board state and integrate with a backend.

**AddEditBoardModal.js -**

The boardsSlice reducer manages state for Kanban boards. It handles actions to fetch initial data, create/edit boards and columns, add and edit tasks, update task status and subtasks, drag tasks between columns, and delete boards and tasks. The reducer dispatches async thunks to persist changes to a backend API. Overall boardsSlice provides Redux actions and reducers to manage complex Kanban board state and integrate with a backend.

**AddEditTaskModal.js -**

The AddEditTaskModal component displays a modal for adding or editing tasks. It manages local form state and dispatches Redux actions on submit to create/update the task. Validation ensures required fields are populated before allowing submit. The modal handles toggling between add and edit modes, prefilling edit fields from Redux state. Overall it provides a reusable form modal for adding and editing tasks with validation and Redux integration.

**DeleteModal.js -**

The DeleteModal component is a reusable confirmation modal used to delete items in a React application. It displays a message specific to the type of item to be deleted ("task" or "section") and offers "Delete" and "Cancel" buttons. Clicking "Delete" triggers the deletion action via the onDeleteBtnClick function, while "Cancel" closes the modal with

the help of setIsDeleteModalOpen. It's a versatile component for confirming and canceling deletion actions.

**taskModal.js -**

The TaskModal component displays a modal overlay for viewing and managing a task's details including its title, description, subtasks, and status. It uses React hooks to manage state locally, integrating with Redux for data and updates. TaskModal allows changing the task status, deleting the task, or opening nested edit and delete modals. It reuses common modal components for editing and deleting to avoid duplication. In summary, TaskModal provides a modular interface for viewing and modifying task details by rendering reusable modal components while synchronizing state and data with Redux.

**loginPage.js -**

The LoginPage component handles user authentication by showing tabs for login and signup. It manages local form state using React hooks and calls API functions to submit the form data. After successful login or signup, it dispatches the side effect callback passed as a prop. The component handles switching between tabs and rendering the correct form based on a useState hook for the active tab. Overall LoginPage provides the UI and local state management for authenticating users while integrating with external APIs for data processing.

**Column.js -**

The Column component displays a column containing tasks for a Kanban board. It gets data from Redux and maps over tasks to render Task components. Column handles drag and drop of tasks between columns by dispatching Redux actions on drop. A useEffect hook sets a random background color for the column. Overall Column renders a droppable column of tasks and enables dragging tasks between columns.

**Sidebar.js -**

The Column component displays a column containing tasks for a Kanban board. It gets data from Redux and maps over tasks to render Task components. Column handles drag and drop of tasks between columns by dispatching Redux actions on drop. A useEffect hook sets a random background color for the column. Overall Column renders a droppable column of tasks and enables dragging tasks between columns.

**Subtask.js -**

The Subtask component displays a checkbox for a task's subtask. It gets the subtask data from the Redux store and dispatches an action to toggle the subtask's completed state on change. The component shows the subtask title, crossing it out if completed. Overall Subtask renders a checkbox to mark a subtask as completed and integrates with Redux for state management.

**Backend source files:**

**App.py -**

This application serves as the backend for a program designed for managing job applications and boards.

**Functionalities***:*
*User Management:* The application offers robust user management features, including user registration and login capabilities, ensuring secure access to the system

*Board Management:* efficient organization by enabling users to create and manage boards for structuring job applications and tasks. Users can effortlessly add, edit, or delete boards to suit their specific needs. To streamline task management, users can set a board as active, ensuring focused attention. Within boards and columns, tasks can be added, edited, or removed, allowing for flexible task tracking and customization. Moreover, users have the flexibility to modify task statuses and subtask completions, enhancing the granularity of application and project management.

*Database Interaction*: Integration with MongoDB for storing user data, applications, and resumes.

## 4. Future Scope:

- Hash the password for security enhancement.
- Implementation of scrolling in the sidebar
- Enhancing UI of the login page
- Deploy the application on some online platform like Heroku.

## 5. Contributors: ~Team 59 Fall 2023.

- Sharan Jamanani
- Saketh Ruddarraju
- Hruthwik Krishnamurthy
- Nayan Bhushan K N