Task Management App with TypeScript

In this TypeScript Task Management Application project, you'll embark on a journey to develop a robust task management tool using TypeScript and React. This project will enhance your understanding of TypeScript fundamentals while introducing you to the practical application of TypeScript in building real-world web applications.

To successfully create the Task Management Application with TypeScript, you'll need to meet the following project requirements:

1. **Task Management Features**:
    - **Task Dashboard Page**:
        - Implement a dashboard interface for managing tasks, including features like task lists, creation, editing, and deletion.
    - **Task Details Display**:
        - Design a detailed view for individual tasks, displaying task information and allowing users to modify task details.
    - **Task Creation and Editing Pages**:
        - Develop forms for creating and editing tasks, incorporating TypeScript types for data validation and error handling.
    - **Authentication and Authorization Pages**:
        - Implement user authentication with Auth0 and authorization pages, including registration and login.
2. **TypeScript Integration**:
    - Utilize TypeScript to enforce type safety and enhance code readability and maintainability throughout the project.
    - Define TypeScript Interfaces or Type Aliases for data shapes and enforce type checking in React components and utility functions.
3. **State Management with Typed Hooks**:
    - Utilize React's useState hooks with TypeScript to manage application state effectively.
4. **Context API for Global State Management**:
    - Implement the Context API in React with TypeScript for managing global application state and sharing data between components.
5. **Authentication and Authorization with Auth0**:
    - Integrate Auth0 authentication services into the application for secure user authentication and authorization.
    - Configure TypeScript types for Auth0 user data.
6. **Error Handling and Validation**:
    - Implement error handling and form validation using TypeScript types and React components to provide a seamless user experience.

7. **GitHub Repository**:
    - Create a GitHub repository for the project and commit code regularly.
    - Maintain a detailed `README.md` file in the repository, providing clear instructions on project setup, installation, and usage.
    - Include documentation on project features, architecture, and implementation details.