**ProjectSync - Frontend (Flutter)**

Objective:

Develop a user-friendly frontend application using Flutter for efficient project management and collaboration.

Requirements:

1. **User Authentication:**
   * Implement a secure user authentication system with features like registration, login, and logout.
2. **Dashboard:**
   * Create a dashboard to display an overview of projects, tasks, and deadlines.
   * Include widgets for quick access to project lists, tasks, notifications, etc.
3. **Project Management:**
   * Allow users to create, edit, and delete projects.
   * Enable users to assign tasks to project members with deadlines and priority levels.
4. **Task Management:**
   * Provide functionality for creating, editing, and deleting tasks within projects.
   * Include options to assign tasks to specific team members and set task deadlines.
5. **Collaboration Features:**
   * Implement real-time collaboration features such as comments, file attachments, and task status updates.
   * Enable users to communicate within the app through messaging or chat functionality.
6. **Notifications:**
   * Set up push notifications to alert users about task assignments, upcoming deadlines, and project updates.
7. **User Settings:**
   * Allow users to customize their profiles, including profile pictures, usernames, and notification preferences.
8. **Offline Support:**
   * Implement offline support to allow users to access and modify data even when offline, with data synchronization when reconnecting.
9. **Responsive Design:**
   * Ensure the application is responsive and optimized for various screen sizes and orientations.
10. **Security:**
    * Implement security measures to protect user data, including encryption of sensitive information and prevention of unauthorized access.

**Project Management Flow Diagram**

1. Project List View

* Displays a list of all projects the user has access to.
* Each project item may include basic details such as project name, description, and status.
* Clicking on a project item navigates the user to the Project Details View.

2. Project Details View

* Displays detailed information about a specific project, including tasks, members, deadlines, etc.
* Provides options to edit project details, delete the project, or navigate back to the Project List View.
* Includes a button to navigate to the Create Task View.

3. Create Project View

* Form for creating a new project.
* Only accessible to users with project lead roles.
* Fields include project name, description, start/end dates, and any other relevant project details.
* After creating the project, the user is redirected to the Project Details View.

4. Edit Project View

* Form for editing an existing project.
* Similar to the Create Project View but pre-populated with existing project details.
* Only accessible to users with project lead roles.
* After editing the project, the user is redirected to the Project Details View.

5. Add Team Members View

* Available within the Create Project View and Edit Project View.
* Allows the project lead to search for and add team members to the project.
* May include options to specify roles or permissions for each team member (e.g., project lead, contributor, viewer).
* After adding team members, the user returns to the Create/Edit Project View.

6. Task Management Views (Optional)

* Task List View, Task Details View, Create Task View, Edit Task View, etc. These views are accessible from the Project Details View and are used for managing tasks within the project.

**Authorization and Roles**

* Project lead roles have permissions to create, edit, and delete projects, as well as add and manage team members.
* Other users may have different levels of access based on their roles (e.g., contributors may have permission to view and edit tasks but not create projects).

**Error Handling**

* Error views should be displayed whenever there are validation errors (e.g., missing fields, invalid input) or authorization failures (e.g., user does not have permission to perform the requested action).