Student Task Scheduler App - Project Plan

# 1. Requirements and Planning

Features:

- Subject Management: Add, delete, and manage subjects.

- Schedule Management: Create, view, and manage the schedule.

- Task/Assignment Tracker: Track tasks, deadlines, and assignments.

- Notifications: Notify students about upcoming classes or deadlines.

- Calendar View: Display weekly or monthly schedules.

Technology Stack:

- Frontend: Flutter

- Backend: SQLite (since it's simple and easy to integrate with Flutter)

- Database: SQLite database for storing schedules, subjects, and tasks.

# 2. Database Design

Tables:

1. Students:

- student\_id (Primary Key)

- name

- email

- phone\_number

2. Subjects:

- subject\_id (Primary Key)

- subject\_name

- student\_id (Foreign Key referencing Students)

3. Schedules:

- schedule\_id (Primary Key)

- subject\_id (Foreign Key referencing Subjects)

- date

- time

- location

4. Tasks:

- task\_id (Primary Key)

- subject\_id (Foreign Key referencing Subjects)

- task\_description

- deadline

- status (e.g., pending, completed)

# 3. Flutter UI Design

Home Screen:

- Display a list of subjects and upcoming tasks.

Subject Detail Screen:

- Show the schedule and tasks related to a specific subject.

Schedule Screen:

- A calendar view showing all classes and tasks for the week/month.

Task Detail Screen:

- View, add, edit, and delete tasks/assignments.

Notifications:

- Implement notifications to remind users of upcoming classes or tasks.

# 4. Backend Integration

Setting up SQLite:

- Define the schema based on the database design.

- Integrate SQLite into your Flutter app using the sqflite package.

CRUD Operations:

- Implement Create, Read, Update, and Delete operations for subjects, schedules, and tasks.

Data Fetching:

- Fetch data from SQLite and display it in the UI.

Notification Setup:

- Use Flutter's flutter\_local\_notifications package to set up reminders and notifications.

# 5. Testing

- Test the app on different devices.

- Ensure data integrity and proper functioning of CRUD operations.

- Verify that notifications are working correctly.

# 6. Deployment and Documentation

- Deploy the app to a mobile device for presentation or submission.

- Document the code and the database schema.

- Prepare a project report detailing the app's functionality, database design, and technology stack.

# 7. Optional Enhancements

User Authentication:

- Allow students to sign in and manage their schedules.

Data Backup:

- Implement backup and restore functionality using cloud storage.

Dark Mode:

- Add a dark mode for the app.