Campus Life Assistant App - Final Lab Task

Covered CLOs (4,5) : Develop advanced mobile applications with multiple screens, persistent storage, and API integration.

Dear Students,

As you embark on this exciting journey of building your **Campus Life Assistant App**, I want to wish you the very best of luck! This project is not only an opportunity to enhance your coding and problem-solving skills, but also a chance to create something that could truly benefit student life.

Remember, this is a learning process—take your time to explore, experiment, and challenge yourself. Don't hesitate to ask for help when needed, and always strive for clean, well-structured code. Each commit you make brings you closer to mastering real-world app development, and the hard work you put in will pay off.

Stay focused, stay curious, and most importantly, enjoy the process!

Good luck and have fun! (**Teacher: Muhammad Abdullah**)

Project Task Breakdown and Marks Distribution (Commit-by-Commit)

Phase 1: Setup and Firebase Integration

- **Objective:** Set up the basic project structure, initialize the GitHub repository, and integrate Firebase for user authentication.

Task 1: GitHub Repository Creation and Initial Commit

- **Description:**
  - Create a new repository on GitHub for the app.
  - Set up the project directory and include basic files (README, .gitignore).
  - Commit the initial setup to GitHub.
  - Share your repo link with and Save in this Excel sheet front of your rollnumebr
  - Share your repo on this email privately: githubprojectmine@gmail.com
- **Marks Distribution:**
  - **Repository Creation & Initial Commit:** 5 Marks

Task 2: Firebase Authentication Integration

- **Description:**
  - Set up a Firebase in the Flutter project.
  - Implement Firebase Authentication (sign-up, login, and profile management).
  - Use Firebase's free tier for authentication (email/password).
  - Commit changes after successful Firebase authentication integration.
- **Marks Distribution:**
  - **Firebase Authentication:** 15 Marks

Phase 2: Class Schedule Management & Local Storage

- **Objective:** Implement the class schedule management system and integrate local storage for offline access.

Task 3: Class Schedule Management

- **Description:**
  - Implement the functionality to add, edit, and delete classes.
  - Sync data with **Cloud Firestore** for cross-device access.
  - Store class schedules in **SQFlite** for offline use.
  - Commit once the schedule management feature is working properly.
- **Marks Distribution:**
  - **Class Schedule Management:** 20 Marks

Task 4: Implement Offline Storage (SQFlite)

- **Description:**
  - Implement **SQFlite** for local data storage (schedules, events).
  - Ensure offline support for when the user has no internet connection.
  - Commit once the offline functionality is tested and working.
- **Marks Distribution:**
  - **Offline Storage Integration:** 10 Marks

Phase 3: Event Notifications and Assignment Tracker

- **Objective:** Add event notifications using Firebase Cloud Messaging (FCM) and an assignment tracker with deadlines.

Task 5: Event Notifications (FCM)

- **Description:**
  - Set up **Firebase Cloud Messaging** (FCM) to send push notifications for upcoming events, assignments, and deadlines.

    - Implement notifications for events like deadlines, exam dates, and class schedules.
    - Commit after successful implementation and testing of event notifications.
- **Marks Distribution:**
    - **Event Notifications:** 15 Marks

Task 6: Assignment Tracker and Deadline Reminders

- **Description:**
    - Implement an **assignment tracker** where students can add assignments and set deadlines.
    - Use **flutter_local_notifications** for deadline reminders.
    - Sync assignments with **Cloud Firestore** and **SQFlite** for offline access.
    - Commit once the assignment tracker is functioning and integrated.
- **Marks Distribution:**
    - **Assignment Tracker & Reminders:** 15 Marks

Phase 4: Study Group Finder and Feedback System

- **Objective:** Enable study group creation and implement a feedback collection system.

Task 7: Study Group Finder

- **Description:**
    - Implement functionality for students to create, join, or leave study groups.
    - Store data using **Firestore** or **Real-time Database**.
    - Commit after the study group feature is functional.
- **Marks Distribution:**
    - **Study Group Finder:** 10 Marks

Task 8: Feedback System

- **Description:**
    - Implement a feedback system where students can rate courses, professors, or campus services.
    - Store feedback in **Cloud Firestore**.
    - Commit once the feedback system is working.
- **Marks Distribution:**
    - **Feedback System:** 10 Marks

Phase 5: Final Integration, Testing, and Documentation

- **Objective:** Finalize the app, fix bugs, optimize, and add documentation.

Task 9: Final Integration & UI Enhancements

- **Description:**
    - Finalize the integration of all features: Authentication, Schedule Management, Notifications, etc.
    - Improve the **UI/UX** for a polished, user-friendly experience.
    - Commit once all features are integrated and UI is finalized.
- **Marks Distribution:**
    - **Final Integration & UI Enhancements:** 10 Marks

Task 10: Testing & Documentation

- **Description:**
    - Conduct thorough testing of the app to identify and fix bugs.
    - Write the app's **documentation**, including instructions on how to use the app and the features implemented.
    - Commit the final version and .apk file and each Creditionals on the Github, including the documentation and testing details.
- **Marks Distribution:**
    - **Testing & Documentation:** 10 Marks

Total Marks: 100 Marks

Commit-by-Commit Marks Distribution:

Task	**Marks**
1. Repository Creation & Initial Commit	5 Marks
2. Firebase Authentication	15 Marks
3. Class Schedule Management	20 Marks

4. Offline Storage Integration (SQFlite)	10 Marks
5. Event Notifications (FCM)	15 Marks
6. Assignment Tracker & Reminders	15 Marks
7. Study Group Finder	10 Marks
8. Feedback System	10 Marks
9. Final Integration & UI Enhancements	10 Marks
10. Testing & Documentation & .APK File	10 Marks
Total	**100 Marks**

Guidelines for GitHub Commits:

1. **Frequent Commits:** Students should commit their code after completing each task (e.g., after integrating Firebase Authentication, or after completing the schedule management feature).
2. **Descriptive Commit Messages:** Each commit should have a descriptive message indicating what has been accomplished in that commit (e.g., "Added Firebase Authentication with email/password login").
3. **Branching (Optional):** If working in teams, consider using branches for each feature and merging them into the main branch once the feature is completed and tested.
4. **Regular Pushes:** Ensure code is pushed to GitHub regularly to maintain backup and version control.
