CW - 03
CSC 4360/6370 - Mobile App Development
Due- 10/21/2025 @11:59 pm

Objective:

In this assignment, you will create a Flutter app that manages tasks using StatefulWidget and setState for state management. The app will allow users to maintain a list of tasks and perform basic **CRUD** (*Create, Read, Update, Delete*) operations on those tasks

Requirements:

- Design a user interface that includes the following elements:
  - A text input field where users can enter task names.
  - An "Add" button to add tasks to the list.
  - A list view to display the task list.
  - Each task in the list should have the following:
    - A checkbox to mark the task as completed.
    - A "Delete" button to remove the task.
- **Implement Task Management**
  - Create a StatefulWidget (e.g., TaskListScreen) that represents the main screen of your app.
  - Inside the TaskListScreen widget, define a list of tasks (each represented as an object with a name and completion status) as an instance variable.
  - Implement methods for adding, completing, and removing tasks within the TaskListScreen widget.
- **Use setState for State Management:**
  - Inside the methods for adding, completing, and removing tasks, use the setState method to update the state of the task list.
- **Display the Task List:**
  - Build a widget to display the list of tasks on the screen.
  - Ensure that marking tasks as completed and deleting tasks update the UI using setState.
- **Add Functionality:**
  - Implement the functionality to add tasks to the list when the user taps the "Add" button.
  - Implement the functionality to mark tasks as completed or incomplete when the user toggles the checkbox.
  - Implement the functionality to remove tasks from the list when the user taps the "Delete" button.

Extra Task for graduate students: Task Priority Feature

Introduce a priority system to the task management app, allowing users to assign and view task priorities.

Requirements:

- Add a priority dropdown or selector to the task input field where users can select a priority level (e.g., Low, Medium, High) when adding a new task.
- Display the priority level next to each task in the list.
- Sort the tasks based on their priority level, with high-priority tasks appearing at the top of the list.
- Allow prioty changes after creation.
-

(Extra Features) – All must do

- Persistence: Save tasks locally (shared_preferences, hive, or sqflite).
- Theming: Light/Dark mode toggle.

Guidelines:

- Extend the task data model to include a priority field.
- Update the UI to show the priority of each task next to its name.
- Implement functionality to sort tasks by priority.
- Ensure that priority selection and sorting are properly handled and reflected in the UI.

Testing:

- Run the app on an emulator or physical device.
- Verify that users can add, complete, and delete tasks, and that the UI updates accordingly.

How to Submit Your Project for Grading

To have your work graded, you must provide **both** of the following:

1. **The APK File:** Upload the final .apk file of your application.
2. **A GitHub Link:** Provide a link to your project's GitHub repository.

Important: Your GitHub repository must show a history of regular commits throughout your development process. We will check your commit history to see your progress. Simply uploading the final code at the last minute is not sufficient for a grade.