CMPS 312 Mobile Application Development
Lab 4- User Interface (UI) development: Components and Layouts

Objective

The objective of the lab is to practice how to design and implement user interfaces using Jetpack Compose. In this lab, you will practice the following:

- Implement composable functions, modifiers, and layouts to create user interfaces.
- Customize the look and feel of UI using modifiers and Material 3 Themes, Colors, Typography, and Shapes.
- Reuse composable functions to simplify your code.
- Test UI on different devices and screen sizes.

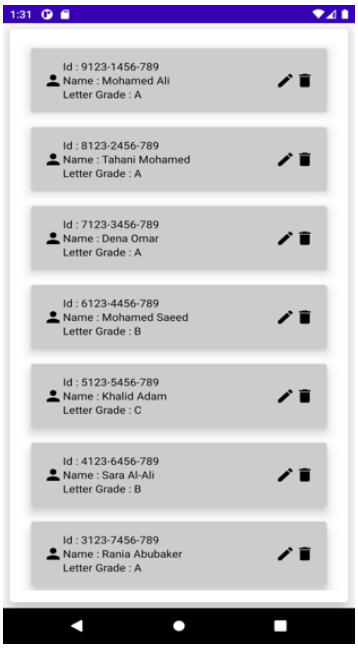
Overview

- **Part A -** Jetpack Compose Basics codelab.
- **Part B -** Implement StudentList app.
- **Part C -** Implement ToDo app.

Part A – Jetpack Compose Basics

Complete Jetpack Compose Basics codelab to practice using composable functions, modifiers, state hoisting, and layouts.

Part B – Implement StudentList app



Read the students details from `students.json` file and display them in a LazyColumn. Each row should be inside a card and have edit and delete buttons.

Implementation Details

1. Create a new project and name it **`StudentList`**
2. Create assets folder and add the `students.json` file
3. Add the necessary dependencies that allow you to serialize JSON files [Refer to the previous Lab 3]
4. Create a serializable data class called `Student` inside a package named `model`
5. Create a class called `StudentRepository` inside a package named repository. Implement **`getStudents()`** method to return the list of students from `students.json` file.
6. Create a Kotlin file named **StudentList** to implement **StudentList** composable that uses **StudentCard** composable**.** This composable should get the students list and display them in a `LazyColumn`.
7. Make sure you add **edit** and **delete** icons to each StudentCard.
8. Test the **StudentList** composable by creating a preview composable.

Part B – Implement ToDo app

Create a **ToDo App** that allows users to manage their to-do list. The app should be user-friendly with a visually appealing interface. Users should be able to perform the following actions:

1. **List Task:** Display the list of tasks sorted by priority and due date to allow the user to easily see the most urgent tasks.
2. **Add Task:** adds a new task to the to-do list. Each task should have a title, a description, a due date, and a priority level.
3. **Edit Task:** edit the details of an existing task including modifying the title, description, due date, and priority level.
4. **Delete Task:** remove a task from the to-do list when they are completed or are no longer relevant.
5. **Visual Appeal:** Design the app with an aesthetically pleasing interface. You are free to use any colors, fonts, and UI elements that you believe will enhance the user experience but ensure that the overall design is intuitive and cohesive.