MOBILE PROGRAMMING

Tutorial 04

Activity 01: Weather App

You will build a React Native weather application that allows users to enter a city name and fetch weather data from the **Open-Meteo API**. The app will display the temperature, weather condition, and background image that matches the weather status of the searched location.

Requirements:

1. **Search for a location**
    - Users enter a city name in the search bar.
    - Use the **Geocoding API** to get the **latitude** and **longitude** of the city.
2. **Fetch weather data from the API**
    - Use **Open-Meteo API** to fetch weather data based on coordinates.
    - Display the temperature (°C) and weather condition.
3. **Update UI based on weather conditions**
    - Use `ImageBackground` to change the **background image** dynamically based on the weather condition.
    - Map **WMO Weather Codes** to corresponding background images.
4. **Enhance user experience**
    - Show an `ActivityIndicator` while fetching data.
    - Display an `Alert` message if an error occurs (e.g., city not found).

iOS

Android

Activity 02: To-Do List App

Create a simple **To-Do List App** using **React Native**. The app should allow users to **add, delete,** and **persist** tasks using `AsyncStorage`.

Install Required Dependencies

- Install `AsyncStorage` to store tasks persistently:

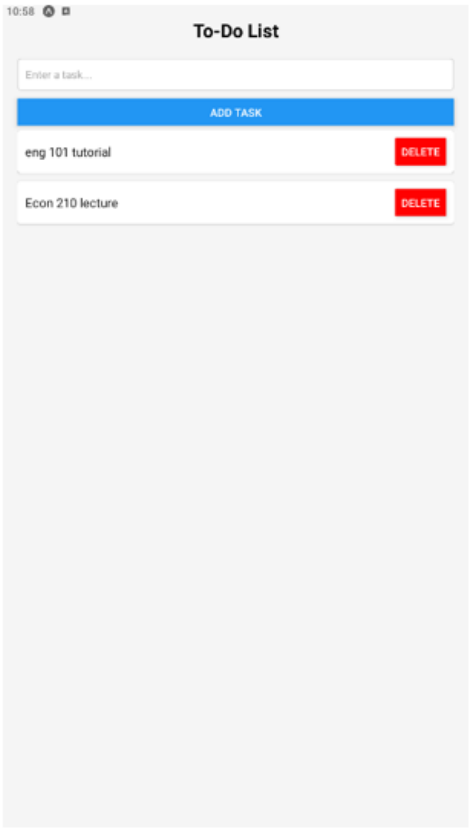
```
npm install @react-native-async-storage/async-storage
```

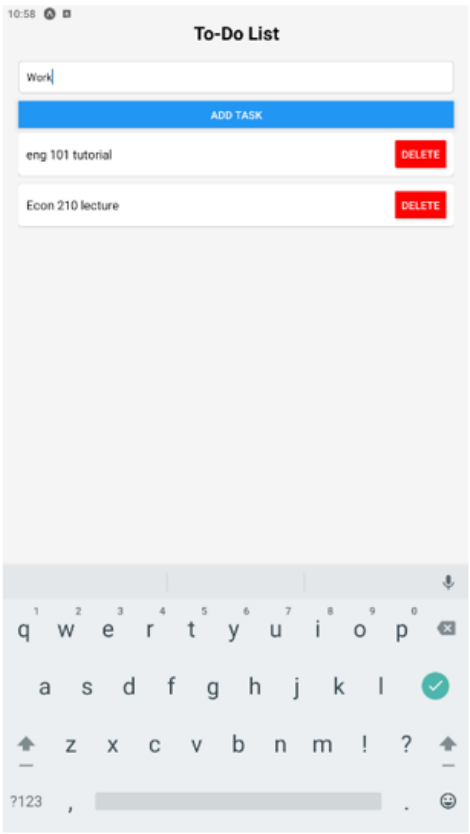
Requirements

1. Add new tasks
    - Users can type a task into a `TextInput` field.
    - When the user presses the "Add Task" button, the task appears in the list.
2. Mark tasks as completed

- Users can tap on a task to toggle completion status (e.g., strike-through text for completed tasks).

3. Delete tasks
   - Users can remove a task by pressing a delete button next to it.
4. Display tasks dynamically
   - The task list updates automatically when a task is added or removed.
5. Enhance user experience
   - Show an `ActivityIndicator` while processing changes (optional).
   - Use `AsyncStorage` to save tasks so they persist when the app restarts (bonus).





Submission

Compress your relevant project files (`App.js`, components, `assets` folder, etc.) into `zip` files (1 `zip` file per activity) and put all `zip` files into a single `zip` file.

Submit the final `zip` file to this tutorial's submission box on FIT LMS.