Objective

Students will design and develop an Android application of their choice that incorporates essential Android development competencies. The application should reflect real-world usability and functionality by meeting the minimum requirements below.

Project Requirements

1. Application Idea and Purpose

- Select an application theme that has real-world utility (e.g., health tracker, task manager, travel assistant).
- Write a brief project proposal outlining the app's purpose, main functionalities, and intended user base.

2. Core Functionalities

1. **Minimum of Three Screens**:
    a. **Main Screen**: Should welcome users and navigate to other screens.
    b. **Action Screen**: Allows users to perform a primary function of the app (e.g., logging an activity or entering data).
    c. **Display Screen**: Shows stored or real-time data to the user, presenting relevant information in a readable format.
2. **User Interface (UI)**:
    a. Use **at least three different layout managers** (e.g., Constraint Layout, Linear Layout, Table Layout).
    b. Integrate **four distinct UI components** (e.g., Buttons, TextView, EditText, ImageView, ToggleButton).
3. **Responsive Design**:
    a. Ensure that your app adapts well to different screen sizes and orientations.

3. Data Management

- **Local Data Storage**:
    - Store and retrieve user inputs using **SQLite** or **SharedPreferences**.
    - The data should persist across app sessions, meaning it remains available when the app is reopened.

4. Application Logic

1. **User Interactions**:
    a. Program meaningful interactions (e.g., button clicks, data validations) that are central to the app's purpose.
2. **Background Task**:
    a. Implement at least one background task using **AsyncTask** or other threading mechanisms for operations like data fetching or image loading.

5. Web Service Integration

- **API Usage**:
    - Integrate a public API relevant to your app's theme (e.g., weather, maps, currency exchange).
    - Display dynamic information from the API on one of the app screens.

6. Quality Assurance

1. **Testing**:
    a. Develop a simple test plan with at least **two functional tests** and **two UI tests** to ensure reliable app performance.
2. **Version Control**:
    a. Use Git for version control, with clear commit messages describing code changes.

7. Documentation

- **Project Documentation**:
    - Provide a README file that includes:
        - Project overview
        - Setup instructions
        - Usage instructions
        - Known issues or limitations

8. Presentation

- **5-Minute Presentation**:
    - Prepare a short presentation to showcase the main features, challenges, and solutions implemented in the project.