**Android App Development 1 - ITE-5333-0NA**

**Lab 12 - Google Maps & Locations**

**Name : Linga Sai Srilaxmi**

**ID : N01653528**

**City Tourism Guide Setup Report**

**Introduction**

The City Tourism Guide is an Android application designed to help users explore a city by displaying both static and dynamic points of interest (POIs) on a Google Map. Built using Kotlin, the app integrates Google Maps SDK, Google Places API, and Volley for network communication. This document provides a concise guide on how to obtain the necessary API key from Google and how to set up and run the application using Android Studio.

**1. Getting a Google Maps API Key**

In order to integrate Google Maps and Places functionality into the app, you need to obtain an API key from the Google Cloud Console. This key allows your app to access Google’s mapping and location-based services.

**Step-by-Step Process:**

**1.1 Go to Google Cloud Console**  
Navigate to <https://console.cloud.google.com/>.

**1.2 Create a New Project**  
In the top menu, click the project dropdown and select “New Project.” Give it a name (e.g., "CityTourismGuide") and create it.

**1.3 Enable Required APIs**  
From the navigation panel, go to “APIs & Services” > “Library.” Search for and enable the following APIs:

* Maps SDK for Android
* Places API

These APIs are essential for map rendering and place data retrieval.

**1.4 Generate an API Key**  
Under “APIs & Services,” go to “Credentials.” Click on “Create Credentials” > “API Key.” A new key will be generated.

**1.5 Secure Your API Key**  
Click on the newly created key to open settings. Restrict the key to be used only by your Android application:

* Select “Application restrictions” > “Android apps”
* Add your package name and SHA-1 certificate fingerprint (can be found in Android Studio > Gradle > signingReport)
* Under “API restrictions,” allow access only to the “Maps SDK for Android” and “Places API” for security purposes.

**2. Setting Up the Project in Android Studio**

Once you have your API key, the next step is setting up the project in Android Studio and linking everything together.

**2.1 Create a New Project**

Open Android Studio and create a new project with the “Empty Activity” template. Use Kotlin as the programming language and set the minimum SDK version to API 24 (Android 7.0) or higher.

**2.2 Organize Project Structure**

Make sure you have the proper project layout:

* MainActivity.kt – Displays the map and POIs.
* PlaceDetailsActivity.kt – Shows details of a selected location.
* activity\_main.xml – Layout containing MapView and UI elements like Spinner.
* activity\_place\_details.xml – Displays name, location, and navigation options.

**2.3 Configure Permissions**

The app requires access to location and the internet. Ensure your project has permissions added for both:

* Location permission for accessing the user’s position.
* Internet permission for API calls to Google services.

**2.4 Add Dependencies and Manage with Version Catalogs**

Dependencies for Google Maps, Location Services, Places API, and Volley are required. Use a version catalog (libs.versions.toml) to organize versions and reference them in your Gradle configuration. Sync your project after setting up dependencies.

**3. Running the Project**

Once the project is set up and the dependencies are in place, you can move on to running the application.

**3.1 Initial Launch**

Connect a physical Android device or use an emulator with Google Play services enabled. Run the application using the "Run" button in Android Studio.

**3.2 Permissions Handling**

On first launch, the app will request access to your location. Make sure to allow it; otherwise, the map and nearby place features won’t work correctly.

**3.3 Testing Features**

Verify that the following features work as expected:

* The map loads correctly and displays your current location.
* Static markers for known POIs appear.
* When selecting a category (e.g., restaurants), dynamic markers from the Places API show up.
* Tapping a marker opens a detail screen with information and a button to navigate via Google Maps.

**Conclusion**

By following these steps, you can successfully generate a Google Maps API key, configure your Android Studio project, and run the City Tourism Guide application. This lab not only demonstrates practical skills in integrating location-based services but also introduces important concepts like dynamic data loading, permission handling, and user interface integration in real-world mobile development.