|  |
| --- |
| **BATCH AND ROLL NO: Q7 42430** |
| **EXPERIMENT NO.10** |
| **TITLE:** Design a mobile app using Google Map and GPS to trace the location. |
| **DATE OF PERFORMANCE:** |
| **DATE OF SUBMISSION:** |

**Title:** Design a mobile app using Google Map and GPS to trace the location.

**Requirements:**

1 Android studio

2.Google Play service Packages

**Theory:**

**Introduction**

In the ever-connected world of mobile applications, harnessing the power of location-based services has become essential for creating dynamic and context-aware applications. This lab focuses on designing a mobile application that integrates Google Maps and Global

Positioning System (GPS) functionalities, enabling users to trace their location and visualize it on a map. The fusion of Google Maps and GPS empowers developers to craft applications that provide real-time location-based information, fostering an enriched user experience.

**Objective of the Lab:** The primary objective of this lab is to guide you through the process of designing a mobile application that leverages Google Maps and GPS technology. By the end of this lab, you should be adept at implementing features such as obtaining real-time location updates, displaying the user's location on a Google Map, and incorporating additional functionalities to enhance the overall location tracking experience.

**Components of the Application:** 1. **Google Maps Integration:**

o The application will integrate Google Maps, allowing users to view and interact with a map interface. o Developers will utilize the Google Maps API to embed the map and leverage its rich features for location-based interactions.

2. **GPS Location Tracking:**

* The application will utilize the device's GPS functionality to trace and update the user's real-time location.
* GPS data will be used to dynamically update the user's marker on the Google Map.

**Lab Prerequisites:**

* Basic understanding of mobile application development concepts.
* Familiarity with the chosen development environment (e.g., Android Studio).  Prior knowledge of programming languages such as Java (for Android)

**Steps:**

**Step 1: Set Up Your Development Environment**

* Ensure that you have Android Studio installed and configured on your machine.  Create a new project in Android Studio.

**Step 2: Obtain Google Maps API Key**

* Obtain a Google Maps API key from the Google Cloud Console.  Enable the "Maps SDK for Android" for your project.

**Step 3: Add Google Maps SDK to Your Project**

* Open the build.gradle file (Module: app) and add the following dependency:

implementation 'com.google.android.gms:play-services-maps:17.0.1'

**Step 4: Design the User Interface**

* Open the XML layout file associated with your main activity (e.g., activity\_main.xml).
* Add a SupportMapFragment or MapView element to your layout to display the Google Map.

**Step 5: Implement Google Maps Integration**

* Open the Java file associated with your main activity (e.g., MainActivity.java).
* Initialize the Google Map and set up its features, such as zoom controls and markers.

**Step 6: Implement GPS Location Tracking**

* Request permission for accessing the device's location in the AndroidManifest.xml.  Implement a LocationListener to receive location updates.

**Step 7: Test Your Application**

* Run your application on an emulator or a physical device.

* Verify that the Google Map is displayed, and the user's location is updated on the map as they move.

**XML Code:**

**AndroidManifest.xml**

**<?xml version="1.0" encoding="utf-8"?>**

**<manifest xmlns:android="http://schemas.android.com/apk/res/android" xmlns:tools="http://schemas.android.com/tools"**

**>**

**<uses-permission android:name="android.permission.ACCESS\_FINE\_LOCATION"/>**

**<uses-permission**

**android:name="android.permission.ACCESS\_COARSE\_LOCATION"/>**

**<uses-permission android:name="android.permission.INTERNET"/>**

**<uses-feature android:name="android.hardware.location.gps"/>**

**<application**

**android:allowBackup="true"**

**android:dataExtractionRules="@xml/data\_extraction\_rules" android:fullBackupContent="@xml/backup\_rules" android:icon="@mipmap/ic\_launcher" android:label="@string/app\_name" android:supportsRtl="true" android:theme="@style/Theme.Lab10"**

**tools:targetApi="31">**

**<meta-data**

**android:name="com.google.android.geo.API\_KEY"**

**android:value="AIzaSyCBlap-jqb0uC3vp7eBrzJn8iiTKJpxtgM" />**

**<activity**

**android:name=".MapsActivity" android:exported="true"**

**android:label="@string/title\_activity\_maps">**

**<intent-filter>**

**<action android:name="android.intent.action.MAIN" />**

**<category android:name="android.intent.category.LAUNCHER"**

**/>**

**</intent-filter>**

**</activity>**

**</application>**

**</manifest>**

**Activity\_maps.xml**

**<?xml version="1.0" encoding="utf-8"?>**

**<fragment xmlns:android="http://schemas.android.com/apk/res/android" xmlns:map="http://schemas.android.com/apk/res-auto" xmlns:tools="http://schemas.android.com/tools"**

**android:id="@+id/map"**

**android:name="com.google.android.gms.maps.SupportMapFragment" android:layout\_width="match\_parent" android:layout\_height="match\_parent"**

**tools:context=".MapsActivity" /**

**Java Code: MainActivity.java**

**package com.example.expt10AD;**

**import androidx.annotation.NonNull;**

**import androidx.core.app.ActivityCompat;**

**import androidx.fragment.app.FragmentActivity;**

**import android.content.pm.PackageManager;**

**import android.location.Location;**

**import android.location.LocationListener;**

**import android.location.LocationManager;**

**import android.os.Bundle; import android.Manifest;**

**import com.google.android.gms.maps.CameraUpdateFactory; import com.google.android.gms.maps.GoogleMap;**

**import com.google.android.gms.maps.OnMapReadyCallback; import com.google.android.gms.maps.SupportMapFragment; import com.google.android.gms.maps.model.LatLng;**

**import com.google.android.gms.maps.model.MarkerOptions; import com.example.lab10.databinding.ActivityMapsBinding; public class MapsActivity extends FragmentActivity implements OnMapReadyCallback**

**{**

**private GoogleMap mMap;**

**private ActivityMapsBinding binding;**

**private LocationListener locationListener;**

**private LocationManager locationManager;**

**private final long MIN\_DIST=5;**

**private final long MIN\_TIME=1000;**

**private LatLng latLng; @Override**

**protected void onCreate(Bundle savedInstanceState) { super.onCreate(savedInstanceState);**

**binding = ActivityMapsBinding.inflate(getLayoutInflater());**

**setContentView(binding.getRoot());**

**// Obtain the SupportMapFragment and get notified when the map is ready to be used.**

**SupportMapFragment mapFragment = (SupportMapFragment) getSupportFragmentManager() .findFragmentById(R.id.map); mapFragment.getMapAsync(this);**

**ActivityCompat.requestPermissions(this, new**

**String[]{Manifest.permission.ACCESS\_FINE\_LOCATION},**

**PackageManager.PERMISSION\_GRANTED);**

**ActivityCompat.requestPermissions(this, new**

**String[]{Manifest.permission.ACCESS\_COARSE\_LOCATION},**

**PackageManager.PERMISSION\_GRANTED);**

**}**

**@Override**

**public void onMapReady (GoogleMap googleMap){ mMap = googleMap;**

**// Add a marker in Sydney and move the camera LatLng sydney = new LatLng(-34, 151);**

**mMap.addMarker(new**

**MarkerOptions().position(sydney).title("Marker in Sydney")); mMap.moveCamera(CameraUpdateFactory.newLatLng(sydney));**

**locationListener = new LocationListener() {**

**@Override**

**public void onLocationChanged(@NonNull Location location)**

**{**

**latLng = new**

**LatLng(location.getLatitude(),location.getLongitude()); mMap.addMarker(new MarkerOptions().position(latLng**

**).title("My position"));**

**mMap.moveCamera(CameraUpdateFactory.newLatLng(latLng));**

**} };**

**locationManager = (LocationManager)**

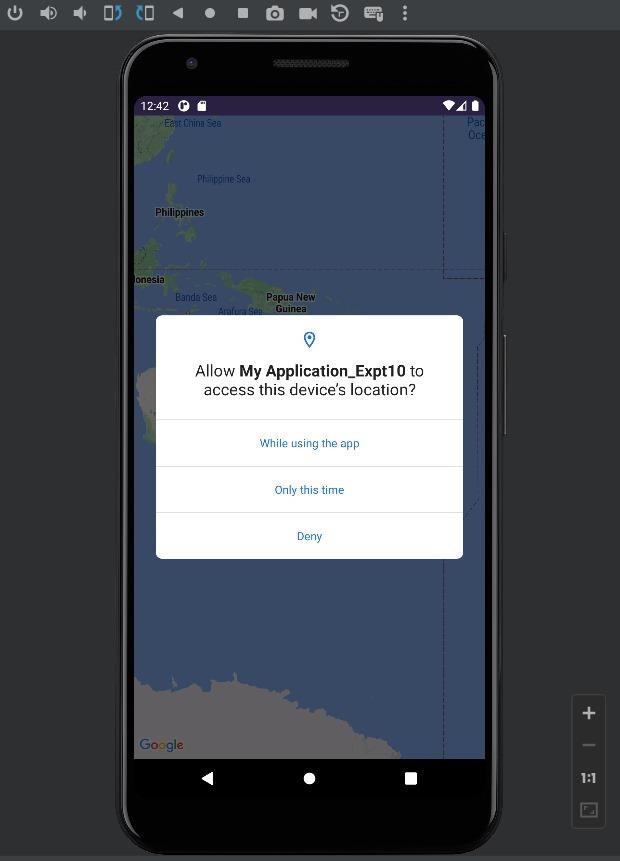
**getSystemService(LOCATION\_SERVICE); try{**

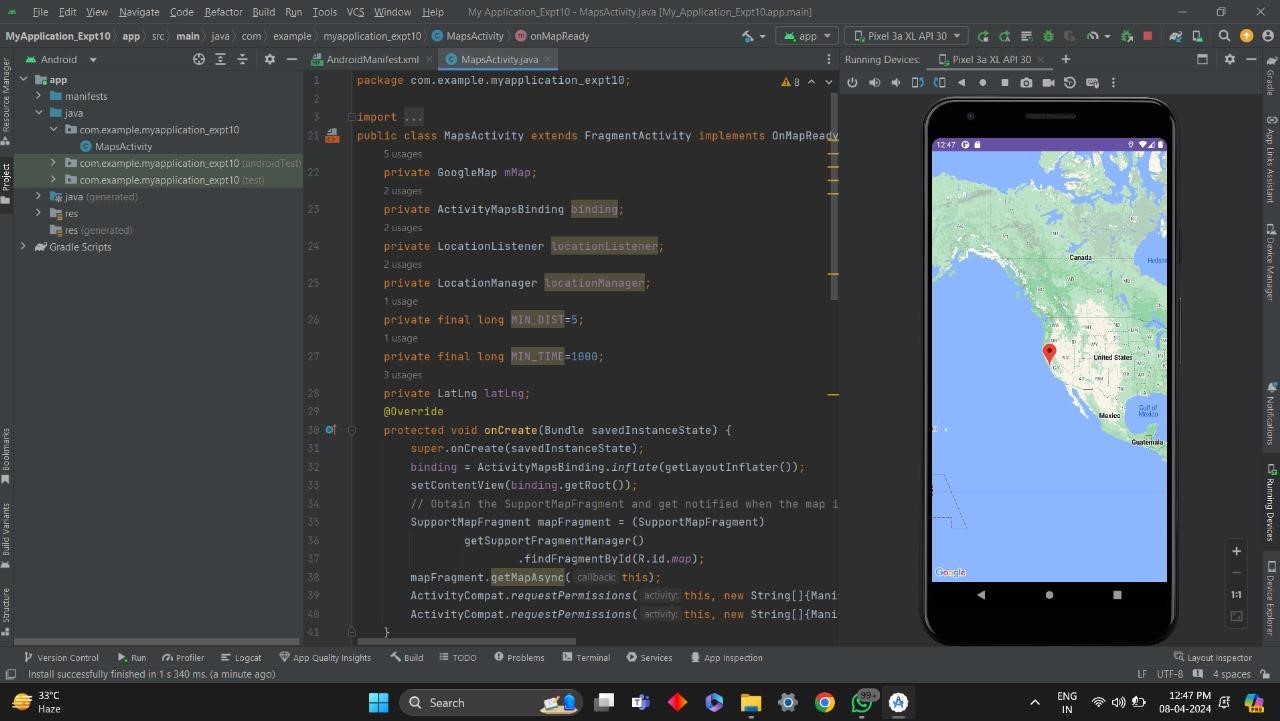
**locationManager.requestLocationUpdates(LocationManager.GPS\_PROVIDER,MIN\_TI**

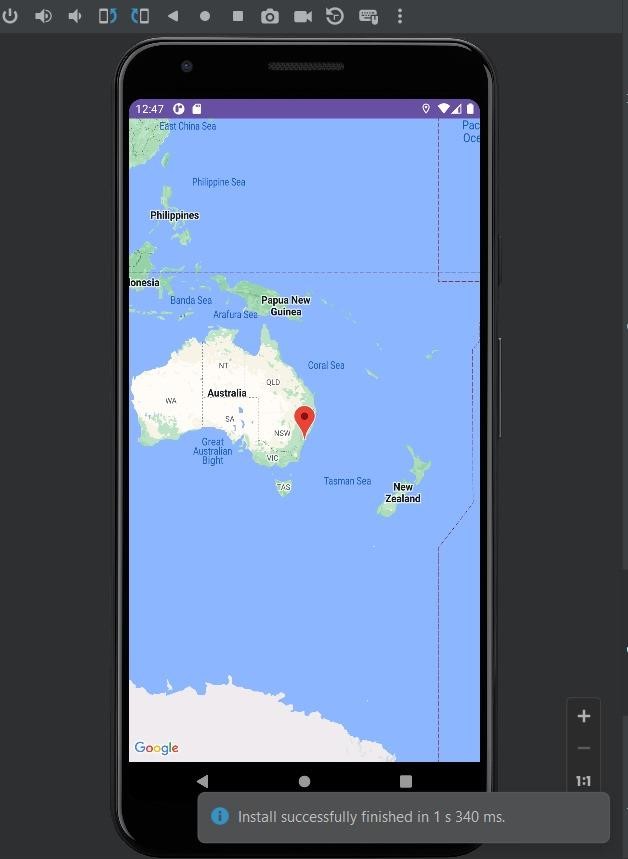
**ME,MIN\_DIST,locationListener);} catch (SecurityException e){ e.printStackTrace();**

**}**

**} }**







**Conclusion: ……………………………………………………………………………………………**

**……………………………………………………………………………………………**

**……………………………………………………………………………………………**