## PRACTICAL NO.08

Exercise - Create android application to demonstrate google maps and locations Implementation:

## **Program:**

```
activity_maps.xml
```

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
  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:layout_width="match_parent"
  android:layout_height="match_parent"
   android:orientation="vertical">
  <androidx.fragment.app.FragmentContainerView
    android:id="@+id/map"
    android:name="com.google.android.gms.maps.SupportMapFragment"
    android:layout_width="match_parent"
    android:layout_height="675dp"
    tools:context=".MapsActivity"/>
    <LinearLayout
      android:layout_width="wrap_content"
      android:layout_height="92dp"
       android:orientation="horizontal">
       <Button
         android:id="@+id/button"
         style="?android:attr/buttonBarButtonStyle"
         android:layout_width="200dp"
         android:layout_height="wrap_content"
         android:layout_weight="1"
         android:background="#F3E889"
         android:onClick="onZoom"
         android:text="@string/zoomin"
```

```
android:textSize="16sp"/>
      <Button
        android:id="@+id/button2"
        style="?android:attr/buttonBarButtonStyle"
        android:layout_width="200dp"
        android:layout_height="wrap_content"
        android:layout_weight="1"
        android:background="#F3E889"
        android:onClick="onZoom"
        android:text="@string/zoomout"
        android:textSize="16sp"/>
    </LinearLayout>
</LinearLayout>
MapsActivity.java
package com.example.map;
import androidx.fragment.app.FragmentActivity;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
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.map.databinding.ActivityMapsBinding;
public class MapsActivity extends FragmentActivity implements OnMapReadyCallback {
  private GoogleMap mMap;
  private ActivityMapsBinding binding;
```

@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);
}
/**
* Manipulates the map once available.
* This callback is triggered when the map is ready to be used.
* This is where we can add markers or lines, add listeners or move the camera. In this case,
* we just add a marker near Sydney, Australia.
* If Google Play services is not installed on the device, the user will be prompted to install
* it inside the SupportMapFragment. This method will only be triggered once the user has
* installed Google Play services and returned to the app.
*/
@Override
public void onMapReady(GoogleMap googleMap) {
  mMap = googleMap;
  // Add a marker in ratnagiri and move the camera
  LatLng ratnagiri = new LatLng(16.992500, 73.294197);
  mMap.addMarker(new MarkerOptions().position(ratnagiri).title("Marker in Ratnagiri"));
  mMap.moveCamera(CameraUpdateFactory.newLatLng(ratnagiri));
  mMap.animateCamera(CameraUpdateFactory.newLatLngZoom(ratnagiri,12));
  mMap.getUiSettings().setZoomControlsEnabled(true);
  // Add a marker in mumbai and move the camera
  LatLng mumbai = new LatLng(19.0760, 72.8777);
  mMap.addMarker(new MarkerOptions().position(mumbai).title("Marker in Mumbai"));
  mMap.moveCamera(CameraUpdateFactory.newLatLng(mumbai));
  mMap.animateCamera(CameraUpdateFactory.newLatLngZoom(mumbai,12));
```

```
mMap.getUiSettings().setZoomControlsEnabled(true);
  }
  public void onZoom(View view){
    if(view.getId()==R.id.button){
      mMap.animateCamera(CameraUpdateFactory.zoomIn());
    }
    if(view.getId()==R.id.button2){
      mMap.animateCamera(CameraUpdateFactory.zoomOut());
AndroidManifest.xml
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"</pre>
  xmlns:tools="http://schemas.android.com/tools">
  <uses-permission android:name="android.permission.INTERNET"/>
  <uses-permission android:name="android.permission.ACCESS_NETWORK_STATE"/>
  <uses-permission android:name="android.permission.ACCESS_FINE_LOCATION"/>
  <uses-permission android:name="android.permission.ACCESS_COARSE_LOCATION"/>
  <uses-feature android:glEsVersion="0x00020000" android:required="true"/>
  <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:roundIcon="@mipmap/ic_launcher_round"
    android:supportsRtl="true"
    android:theme="@style/Theme.Map"
    tools:targetApi="31">
    <!--
```

```
TODO: Before you run your application, you need a Google Maps API key.
       To get one, follow the directions here:
         https://developers.google.com/maps/documentation/android-sdk/get-api-key
       Once you have your API key (it starts with "AIza"), define a new property in your
       project's local.properties file (e.g. MAPS_API_KEY=Aiza...), and replace the
       "YOUR_API_KEY" string in this file with "${MAPS_API_KEY}".
    -->
    <meta-data
       android:name="com.google.android.geo.API_KEY"
       android:value="AIzaSyAbud2J76YA1NK9D8k3p9gErjaH-P8G9gA" />
    <meta-data
       android:name="com.google.android.gms.version"
       android:value="@integer/google_play_services_version"/>
    <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>
```

## **Output:**







