google_maps_api.xml

```
<resources>
    <!--
    TODO: Before you run your application, you need a Google Maps API key.

    To get one, follow this link, follow the directions and press "Create" at the
end:


https://console.developers.google.com/flows/enableapi?apiid=maps_android_backend&k
eyType=CLIENT_SIDE_ANDROID&r=89:AB:73:69:27:2D:3A:69:97:D2:8B:90:7C:EE:26:7E:F6:C9
:34:08%3Bcom.example.locationfinder

    You can also add your credentials to an existing key, using these values:

    Package name:
    com.example.locationfinder

    SHA-1 certificate fingerprint:
    89:AB:73:69:27:2D:3A:69:97:D2:8B:90:7C:EE:26:7E:F6:C9:34:08

    Alternatively, follow the directions here:
    https://developers.google.com/maps/documentation/android/start#get-key

    Once you have your key (it starts with "AIza"), replace the "google_maps_key"
    string in this file.
    -->
    <string name="google_maps_key" templateMergeStrategy="preserve"
translatable="false">AIzaSyAtRtZ3eW8hCmRZQwmJpqjyx3gq8HQ_Il8</string>
</resources>
```

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" />
```

MapsActivity.java

```
package com.example.locationfinder;

import androidx.appcompat.app.AppCompatActivity;
import androidx.fragment.app.FragmentActivity;

import android.os.Bundle;
import android.view.Menu;
import android.view.MenuInflater;
import android.view.MenuItem;

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;

public class MapsActivity extends AppCompatActivity implements OnMapReadyCallback
{

    private GoogleMap mMap;

    @Override
    protected void onCreate(Bundle savedInstanceState)
    {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_maps);
        // 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 Sydney and move the camera
        LatLng MyHouse = new LatLng(19.476814, 74.578380);
        mMap.addMarker(new MarkerOptions().position(MyHouse).title("Marker in Guha
to MyHouse"));
        mMap.moveCamera(CameraUpdateFactory.newLatLngZoom(MyHouse,10F));
    }

    @Override
    public boolean onCreateOptionsMenu(Menu menu)
    {
        MenuInflater inflater = getMenuInflater();
        inflater.inflate(R.menu.map_options, menu);
        return true;
    }

    @Override
    public boolean onOptionsItemSelected(MenuItem item)
    {
```

```
        // Change the map type based on the user's selection.
        switch (item.getItemId())
        {
            case R.id.normal_map:
                mMap.setMapType(GoogleMap.MAP_TYPE_NORMAL);
                return true;
            case R.id.hybrid_map:
                mMap.setMapType(GoogleMap.MAP_TYPE_HYBRID);
                return true;
            case R.id.satellite_map:
                mMap.setMapType(GoogleMap.MAP_TYPE_SATELLITE);
                return true;
            case R.id.terrain_map:
                mMap.setMapType(GoogleMap.MAP_TYPE_TERRAIN);
                return true;
            default:
                return super.onOptionsItemSelected(item);
        }
    }

}
```

menu
map_options.xml

```
<?xml version="1.0" encoding="utf-8"?>
<menu xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto">
    <item android:id="@+id/normal_map"
        android:title="@string/normal_map"
        app:showAsAction="never"/>
    <item android:id="@+id/hybrid_map"
        android:title="@string/hybrid_map"
        app:showAsAction="never"/>
    <item android:id="@+id/satellite_map"
        android:title="@string/satellite_map"
        app:showAsAction="never"/>
    <item android:id="@+id/terrain_map"
        android:title="@string/terrain_map"
        app:showAsAction="never"/>
</menu>
```

string.xml

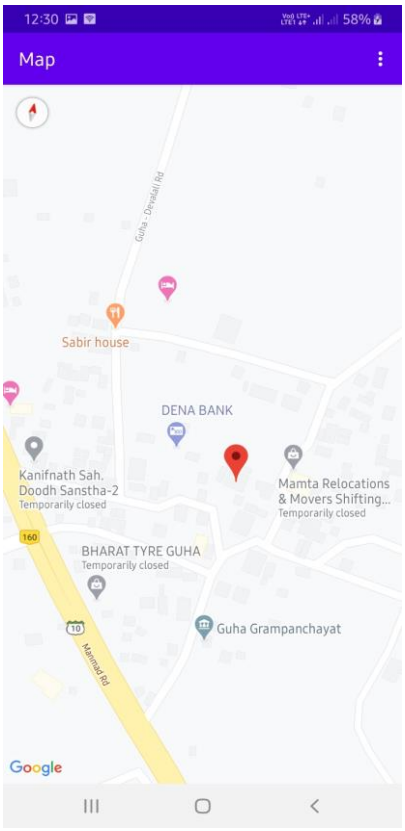
```
<resources>
    <string name="app_name">LocationFinder</string>
    <string name="title_activity_maps">Map</string>
    <string name="normal_map">Normal Map</string>
    <string name="hybrid_map">Hybrid Map</string>
    <string name="satellite_map">Satellite Map</string>
    <string name="terrain_map">Terrain Map</string>
</resources>
```

AndroidManifest.xml

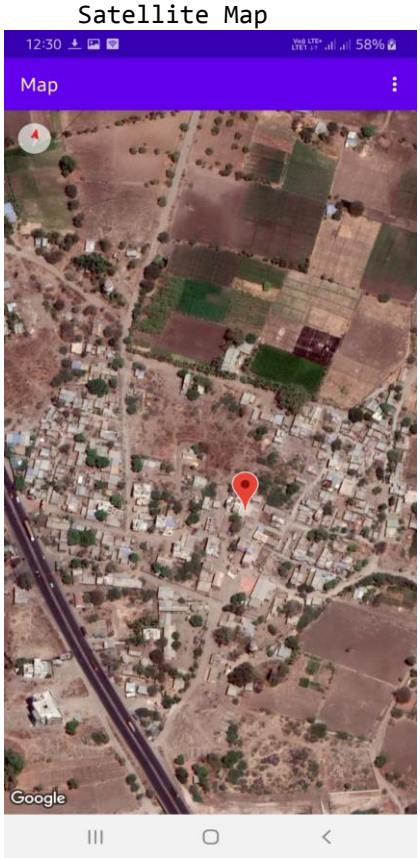
```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.example.locationfinder">

    <!--
         The ACCESS_COARSE/FINE_LOCATION permissions are not required to use
         Google Maps Android API v2, but you must specify either coarse or fine
         location permissions for the 'MyLocation' functionality.
    -->
    <uses-permission android:name="android.permission.ACCESS_FINE_LOCATION" />

    <application
        android:allowBackup="true"
        android:icon="@mipmap/ic_launcher"
        android:label="@string/app_name"
        android:roundIcon="@mipmap/ic_launcher_round"
        android:supportsRtl="true"
        android:theme="@style/AppTheme">

        <!--
             The API key for Google Maps-based APIs is defined as a string
resource.
             (See the file "res/values/google_maps_api.xml").
             Note that the API key is linked to the encryption key used to sign
the APK.
             You need a different API key for each encryption key, including the
release key that is used to
             sign the APK for publishing.
             You can define the keys for the debug and release targets in
src/debug/ and src/release/.
        -->
        <meta-data
            android:name="com.google.android.geo.API_KEY"
            android:value="@string/google_maps_key" />

        <activity
            android:name=".MapsActivity"
            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>
```

Normal Map

With Zoom



Satellite Map



Terrain Map