Before move on with the application development we have to add dependencies in the gradle.

Refer **developer.android.com**

Manifest file

*<?***xml version="1.0" encoding="utf-8"***?>*<**manifest xmlns:android="http://schemas.android.com/apk/res/android"  
 package="com.example.location\_example"**>  
  
 *<!--  
 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"** />  
 <**uses-permission android:name="android.permission.ACCESS\_COARSE\_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"**>  
 <**intent-filter**>  
 <**action android:name="android.intent.action.MAIN"** />  
  
 <**category android:name="android.intent.category.LAUNCHER"** />  
 </**intent-filter**>  
 </**activity**>  
 </**application**>  
  
</**manifest**>

Google\_map\_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&keyType=CLIENT\_SIDE\_ANDROID&r=DD:14:EB:48:9C:AA:B7:5C:2D:C9:6D:27:9B:9D:62:5B:2E:B4:66:12%3Bcom.example.location\_example  
  
 You can also add your credentials to an existing key, using these values:  
  
 Package name:  
 com.example.location\_example  
  
 SHA-1 certificate fingerprint:  
 DD:14:EB:48:9C:AA:B7:5C:2D:C9:6D:27:9B:9D:62:5B:2E:B4:66:12  
  
 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"**>AIzaSyDfWpWq7ENSiUJU052VRo8Q\_mqaikEJdfM</**string**>  
</**resources**>

Java file

**package** com.example.location\_example;  
  
**import** androidx.annotation.NonNull;  
**import** androidx.core.app.ActivityCompat;  
**import** androidx.fragment.app.FragmentActivity;  
  
**import** android.Manifest;  
**import** android.content.pm.PackageManager;  
**import** android.location.Location;  
**import** android.os.Bundle;  
**import** android.util.Log;  
  
**import** com.google.android.gms.location.FusedLocationProviderClient;  
**import** com.google.android.gms.location.LocationAvailability;  
**import** com.google.android.gms.location.LocationCallback;  
**import** com.google.android.gms.location.LocationRequest;  
**import** com.google.android.gms.location.LocationResult;  
**import** com.google.android.gms.location.LocationServices;  
**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.google.android.gms.tasks.OnFailureListener;  
**import** com.google.android.gms.tasks.OnSuccessListener;  
  
**public class** MapsActivity **extends** FragmentActivity **implements** OnMapReadyCallback {  
  
 **private** GoogleMap **mMap**;  
 **private static final int *UPDATE\_INTERVEL*** = 5000; *// update the location after 5 seconds* FusedLocationProviderClient **locationProviderClient**;  
 LocationRequest **locationRequest**;  
 LocationCallback **locationCallback**;  
 **private** Location **currentlocation**;  
  
 @Override  
 **protected void** onCreate(Bundle savedInstanceState) {  
 **super**.onCreate(savedInstanceState);  
 setContentView(R.layout.***activity\_maps***);  
 **locationProviderClient** = LocationServices.*getFusedLocationProviderClient*(**this**);  
 **locationRequest** = LocationRequest.*create*();  
 **locationRequest**.setPriority(LocationRequest.***PRIORITY\_HIGH\_ACCURACY***);  
 **locationRequest**.setInterval(***UPDATE\_INTERVEL***);  
 **locationCallback** = **new** LocationCallback() {  
 @Override  
 **public void** onLocationAvailability(LocationAvailability locationAvailability) {  
 **super**.onLocationAvailability(locationAvailability);  
 **if** (locationAvailability.isLocationAvailable())  
 Log.*i*(**"Harry"**, **"Location is available"**);  
 **else** Log.*i*(**"Harry"**, **"Location is not available"**);  
 }  
  
 @Override  
 **public void** onLocationResult(LocationResult locationResult) {  
 **super**.onLocationResult(locationResult);  
 Log.*i*(**"Harry"**, **"Location result is available"**);  
 }  
 };  
 startGettingLocation();  
  
 }  
  
 **private void** startGettingLocation()  
 {  
 **if**(ActivityCompat.*checkSelfPermission*(**this**, Manifest.permission.***ACCESS\_FINE\_LOCATION***)== PackageManager.***PERMISSION\_GRANTED*** &&  
 ActivityCompat.*checkSelfPermission*(**this**, Manifest.permission.***ACCESS\_COARSE\_LOCATION***)== PackageManager.***PERMISSION\_GRANTED***) {  
 **locationProviderClient**.requestLocationUpdates(**locationRequest**,**locationCallback**,MapsActivity.**this**.getMainLooper());  
 **locationProviderClient**.getLastLocation().addOnSuccessListener(**new** OnSuccessListener<Location>() {  
 @Override  
 **public void** onSuccess(Location location) {  
 **currentlocation**=location;  
 *// Obtain the SupportMapFragment and get notified when the map is ready to be used.* SupportMapFragment mapFragment = (SupportMapFragment) getSupportFragmentManager()  
 .findFragmentById(R.id.***map***);  
 mapFragment.getMapAsync(MapsActivity.**this**);  
  
 }  
 });  
 **locationProviderClient**.getLastLocation().addOnFailureListener(**new** OnFailureListener() {  
 @Override  
 **public void** onFailure(@NonNull Exception e) {  
 Log.*i*(**"Harry"**,**"Exception while getting location "**);  
 }  
 });  
  
 }  
  
 }  
  
 *//@Override  
 //public void onRequestPermissionsResult(int requestCode, @NonNull String[] permissions, @NonNull int[] grantResults) {  
 // super.onRequestPermissionsResult(requestCode, permissions, grantResults);  
 // startGettingLocation();  
 //}* @Override  
 **public void** onMapReady(GoogleMap googleMap) {  
 **mMap** = googleMap;  
  
 *// Add a marker in Sydney and move the camera* LatLng current\_place= **new** LatLng(**currentlocation**.getLatitude(),**currentlocation**.getLongitude());  
 **mMap**.addMarker(**new** MarkerOptions().position(current\_place).title(**"Marker"**));  
 **mMap**.animateCamera(CameraUpdateFactory.*newLatLngZoom*(current\_place,15.0f));  
 }  
}

output screen:

