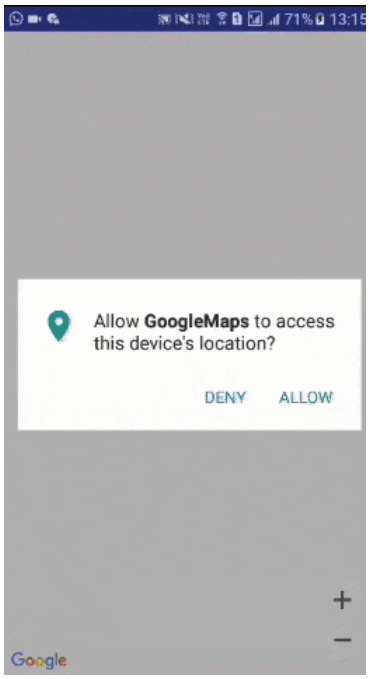
Google Map

- Android provides facility to integrate Google map in our application.
- Google map displays your current location, navigate location direction, search location etc.
- We can also customize Google map according to our requirement.
- You can show any location on the map , or can show different routes on the map e.t.c. You can also customize the map according to your choices.

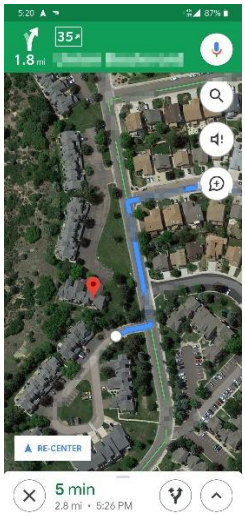


Types of Google Maps

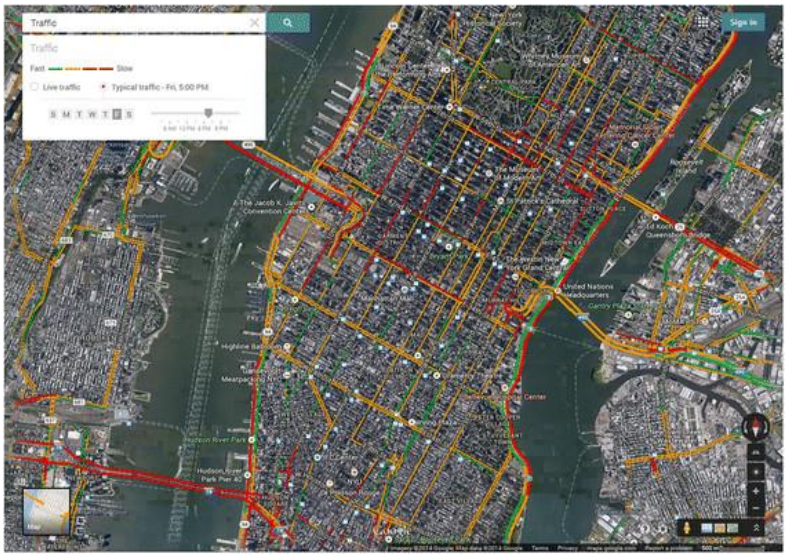
There are four different types of Google maps, as well as an optional to no map at all. Each of them gives different view on map. These maps are as follow:

- Normal: This type of map displays typical road map, natural features like river and some features build by humans.
  googleMap.setMapType(GoogleMap.MAP_TYPE_NORMAL);

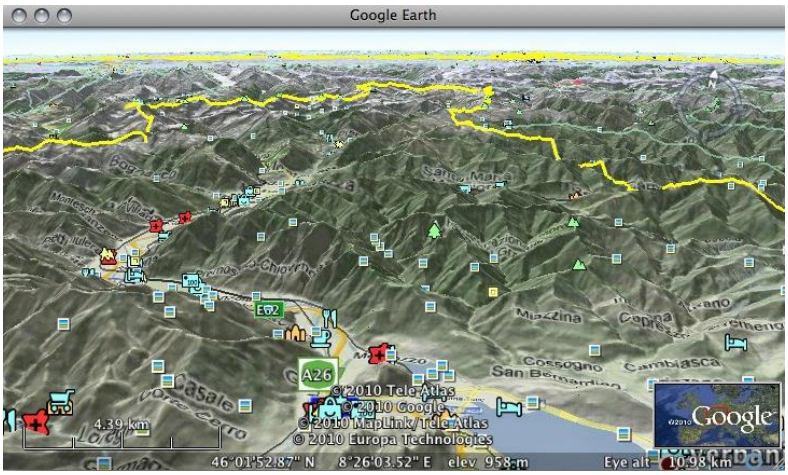
- Hybrid: This type of map displays satellite photograph data with typical road maps. It also displays road and feature labels.
  googleMap.setMapType(GoogleMap.MAP_TYPE_SATELLITE);



- Satellite: Satellite type displays satellite photograph data, but doesn't display road and feature labels.
  googleMap.setMapType(GoogleMap.MAP_TYPE_HYBRID);



- Terrain: This type displays photographic data. This includes colors, contour lines and labels and perspective shading.
  googleMap.setMapType(GoogleMap.MAP_TYPE_TERRAIN);



- None: This type displays an empty grid with no tiles loaded

Methods of Google Map

Methods	Description
addCircle(CircleOptions options)	This method add circle to map.
addPolygon(PolygonOptions options)	This method add polygon to map.
addTileOverlay(TileOverlayOptions options)	This method add tile overlay to the map.
animateCamera(CameraUpdate update)	This method moves the map according to the update with an animation.
clear()	This method removes everything from the map.
getMyLocation()	This method returns the currently displayed user location.
moveCamera(CameraUpdate update)	This method reposition the camera according to the instructions defined in the update.
setTrafficEnabled(boolean enabled)	This method set the traffic layer on or off.
snapshot(GoogleMap.SnapshotReadyCallback callback)	This method takes a snapshot of the map.
stopAnimation()	This method stops the camera animation if there is any progress.

What Permissions We Need to User Google Map API Services

Permissions:

1. **<uses-permission** android:name="android.permission.ACCESS_FINE_LOCATION" **/>**
2. **<uses-permission** android:name="android.permission.ACCESS_COARSE_LOCATION" **/>**

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

Dependencies

Add Dependency in build.gradle (Module File in dependencies { ... }):

- implementation 'com.google.android.gms:play-services-maps:17.0.1'
- implementation 'com.google.android.gms:play-services-location:17.0.0'

What To Add In Manifest.xml File

```
<meta-data
    android:name="com.google.android.geo.API_KEY"
    android:value="[redacted] />
```

```
<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" />
```

```
<meta-data
    android:name="com.google.android.gms.version"
    android:value="12451000" />

<uses-library
    android:name="org.apache.http.legacy"
    android:required="false" />
```

Add FrameLayout in MainActivity.xml

activity_main.xml | MainActivity.java | AndroidManifest.xml | build.gradle (:app)

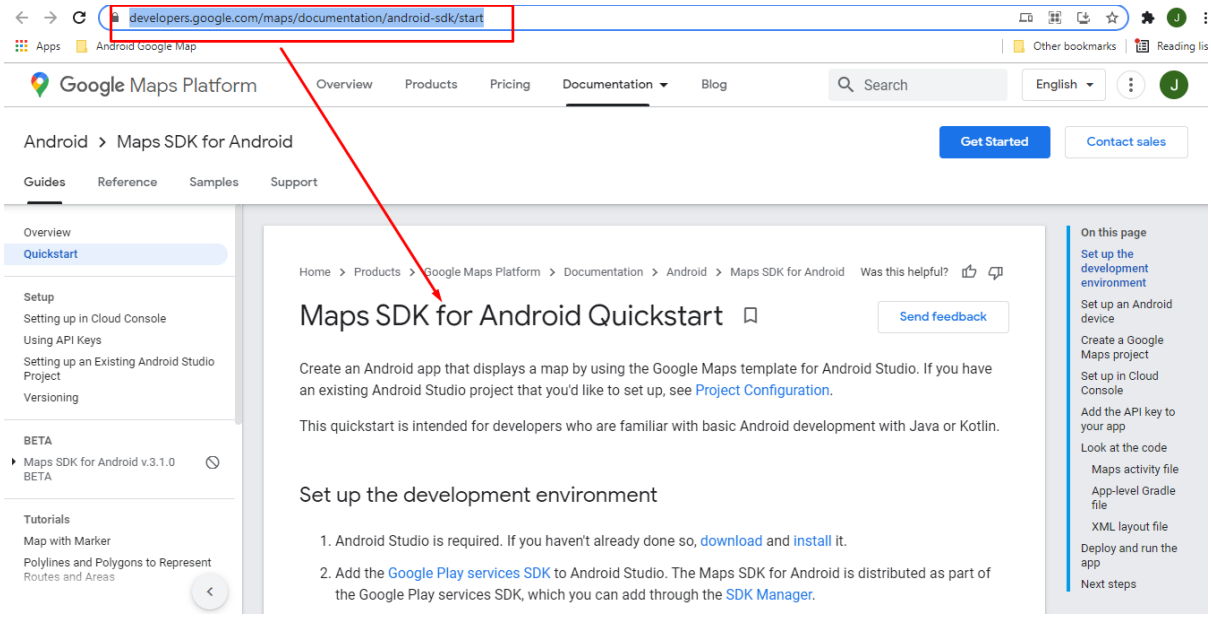
```xml
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".MainActivity">

    <fragment
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:id="@+id/maps"
        android:name="com.google.android.gms.maps.SupportMapFragment"/>

</RelativeLayout>
```

On MapReady() Method

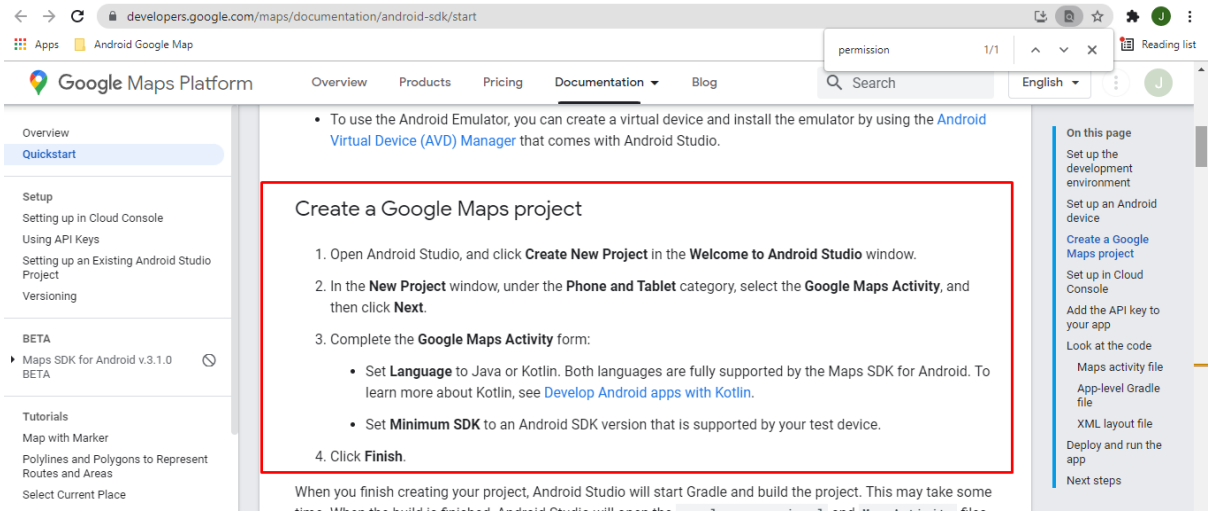
```
    @Override
    public void onMapReady(GoogleMap googleMap) {
        mMap = googleMap;
        //23.059663241662548, 72.51212682192856

        // Add a marker in Sydney and move the camera
        LatLng sydney = new LatLng( latitude: 23.059663241662548,  longitude: 72.51212682192856);
        mMap.addMarker(new MarkerOptions().position(sydney).title("ANDROID LIVE MARKER"));
        mMap.moveCamera(CameraUpdateFactory.newLatLng(sydney));
    }
}
```

HOW TO START WITH GOOGLE MAP API & INTEGRATE API KEY

- GO to the link : https://developers.google.com/maps/documentation/android-sdk/start

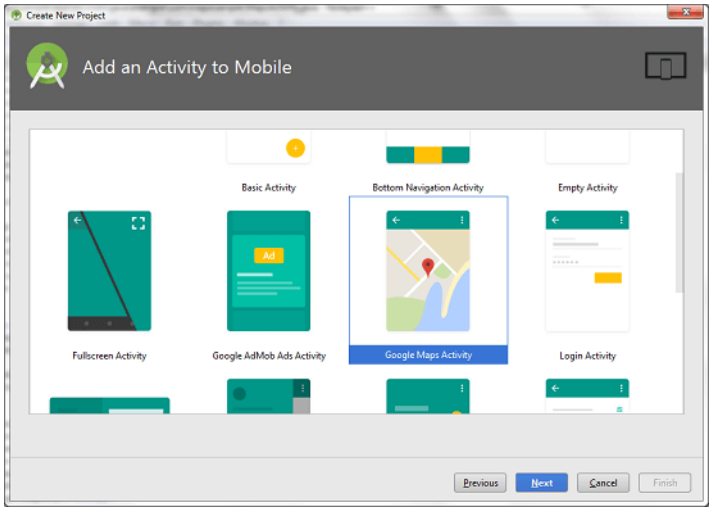


- Read the instructions & Follow them



- Create New Project in Android Studio

- Select Google Map Activity



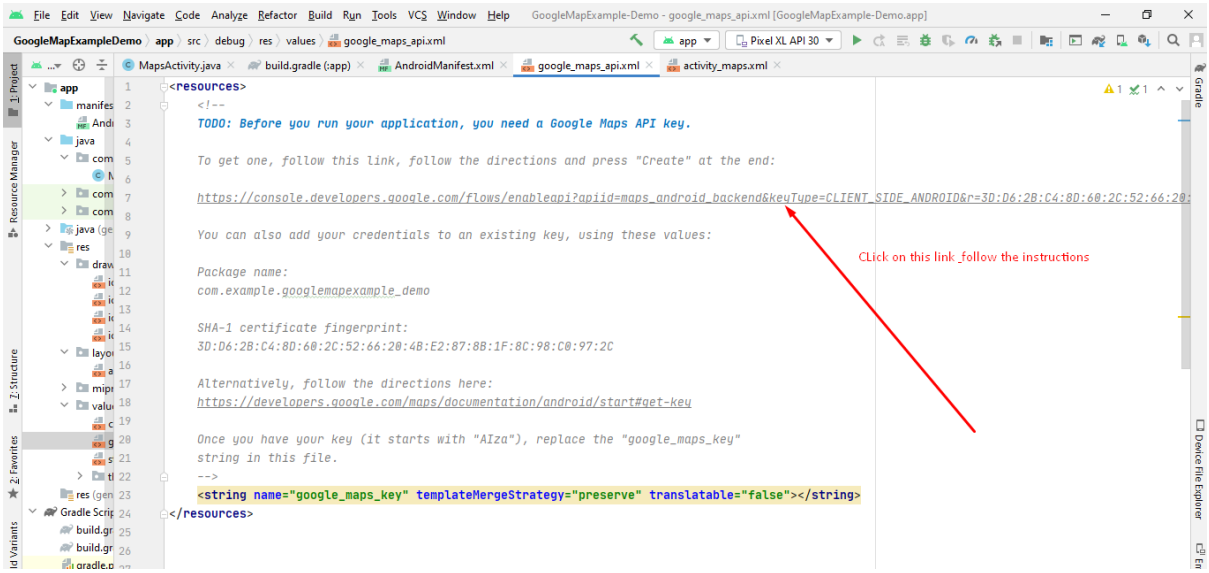
- Open the Project & Find **google_maps_api.xml** file
- Then copy URL of console.developers.google.com & follow the Instruction to Generate API Key :

https://console.developers.google.com/flows/enableapi?apiid=maps_android_backend&keyType=CLIENT_SIDE_ANDROID&r=3D:D6:2B:C4:8D:60:2C:52:66:20:4B:E2:87:8B:1F:8C:98:C0:97:2C%3Bcom.example.googlemapliveprojdemo1

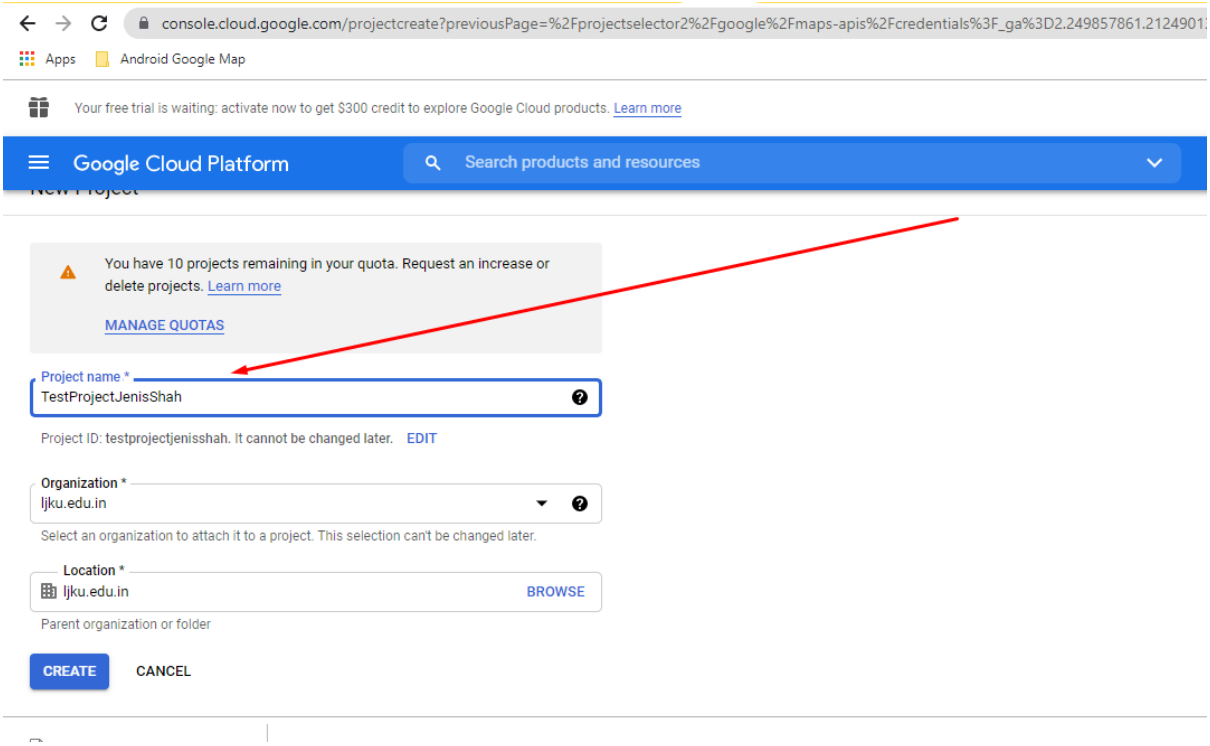
OR

Alternatively, follow the directions here:

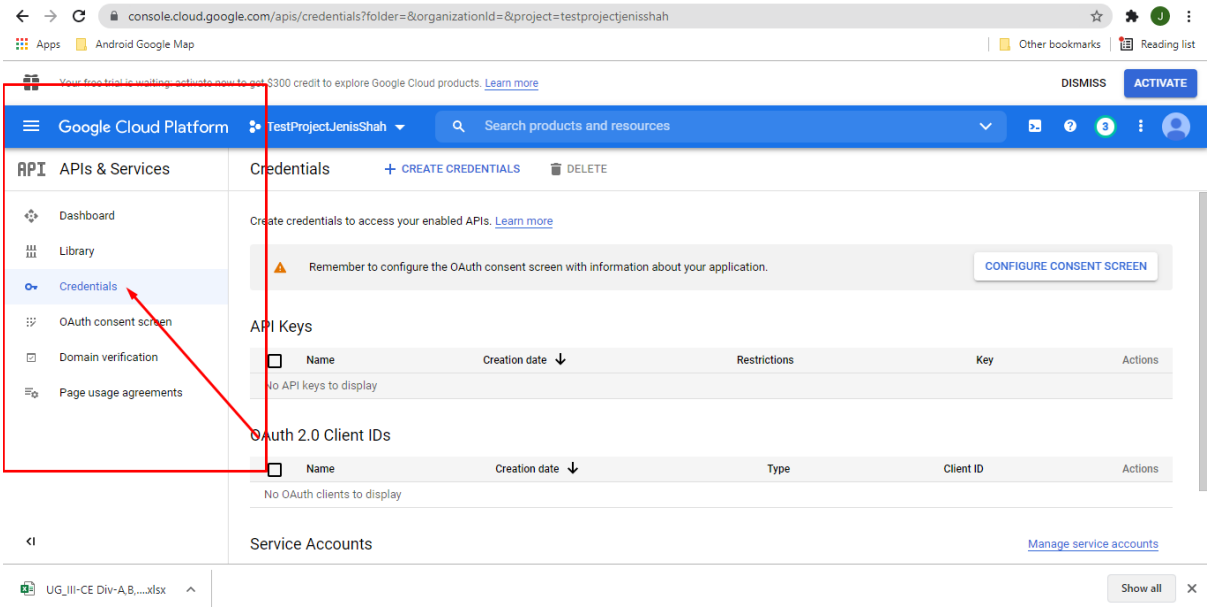
https://developers.google.com/maps/documentation/android/start#get-key



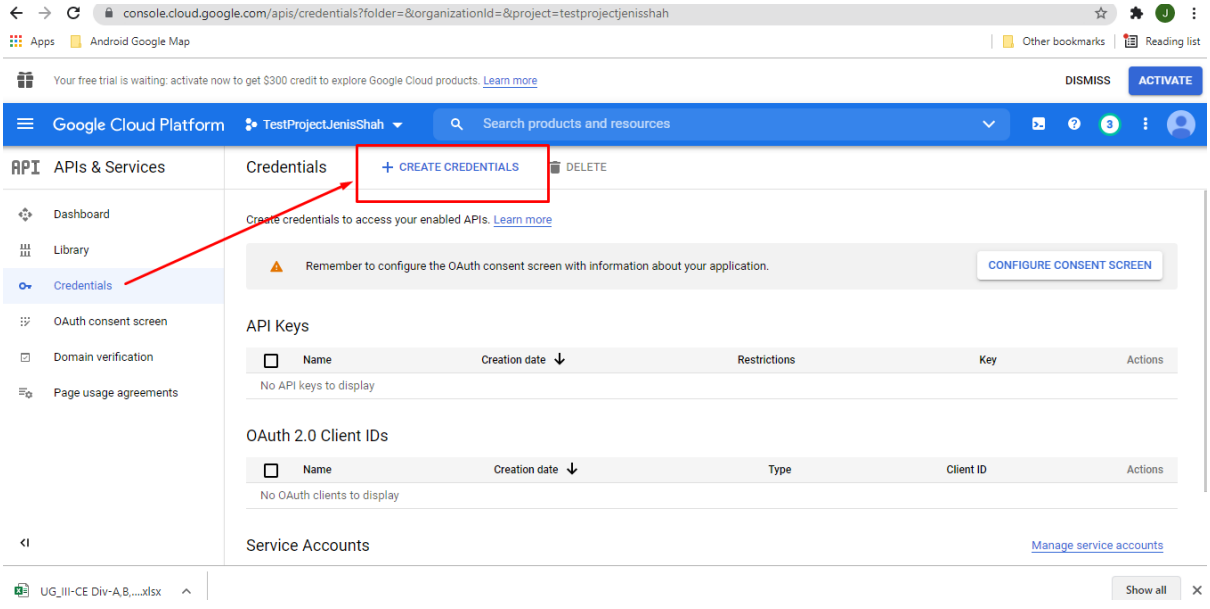
- Create a Project – Select Organization



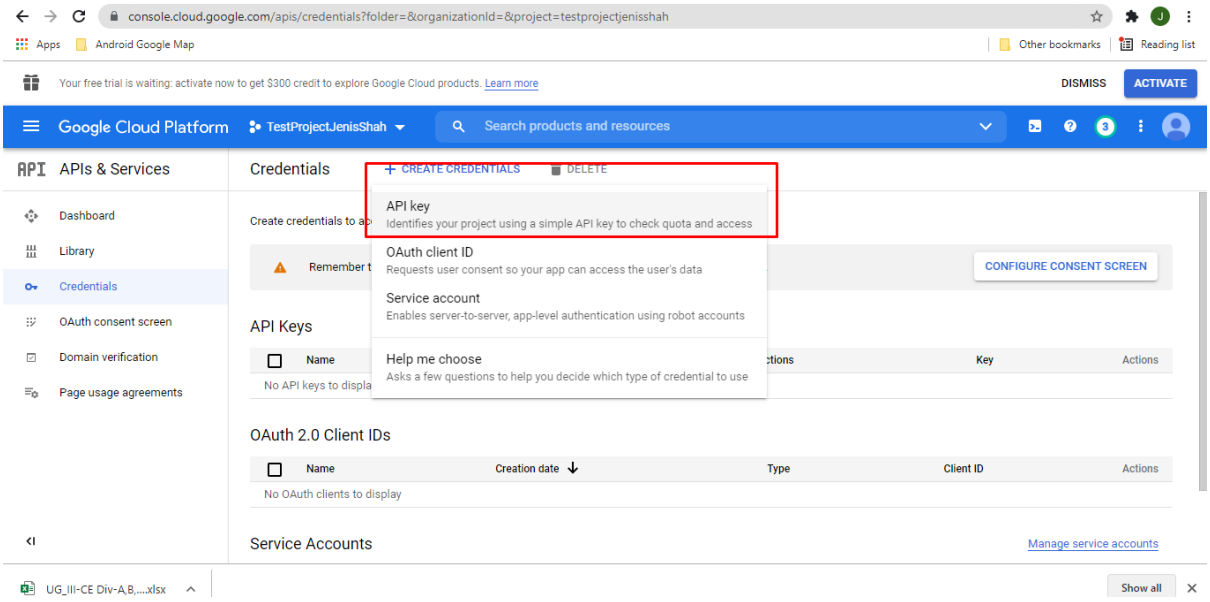
- Create Credentials



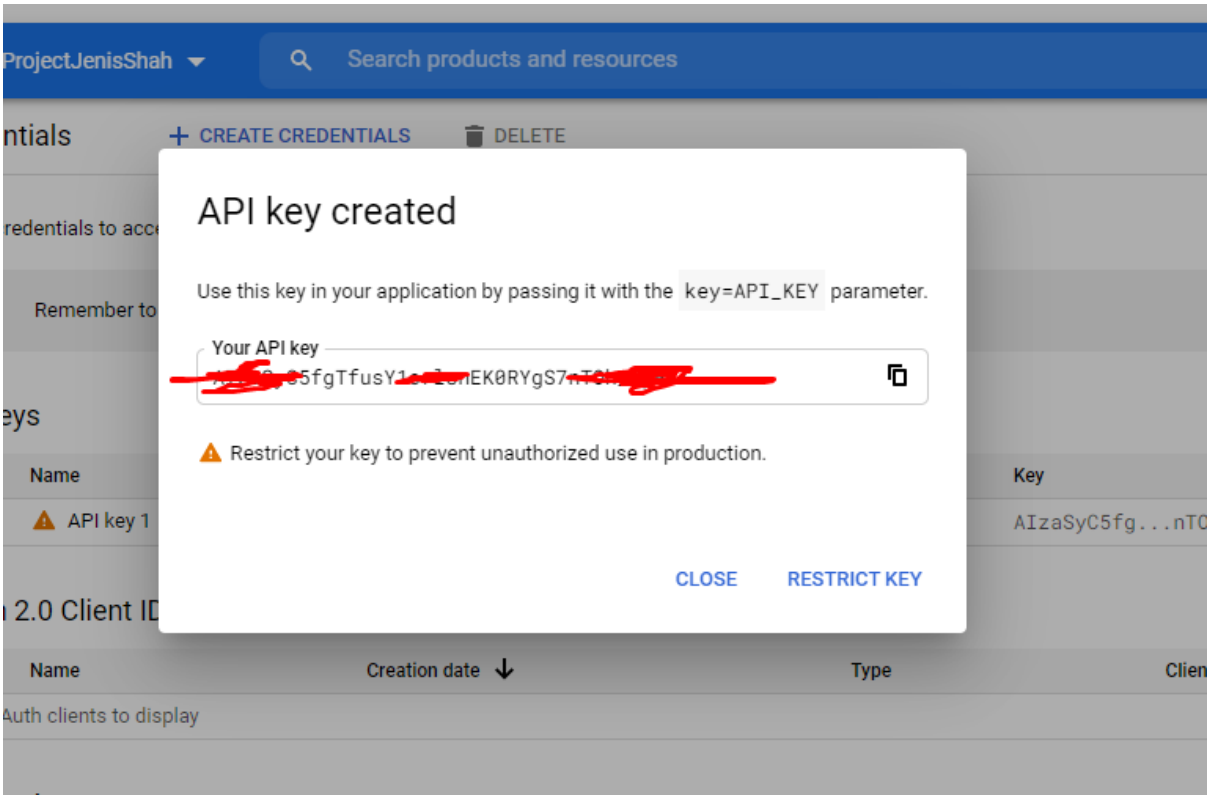
- Click on Create Credentials



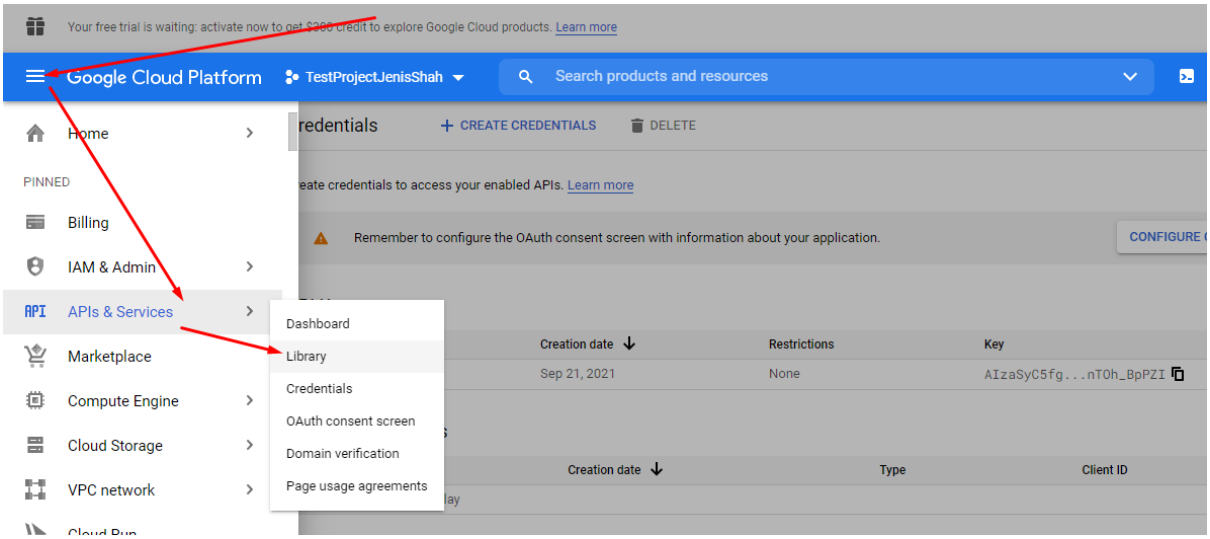
- Click on API Key



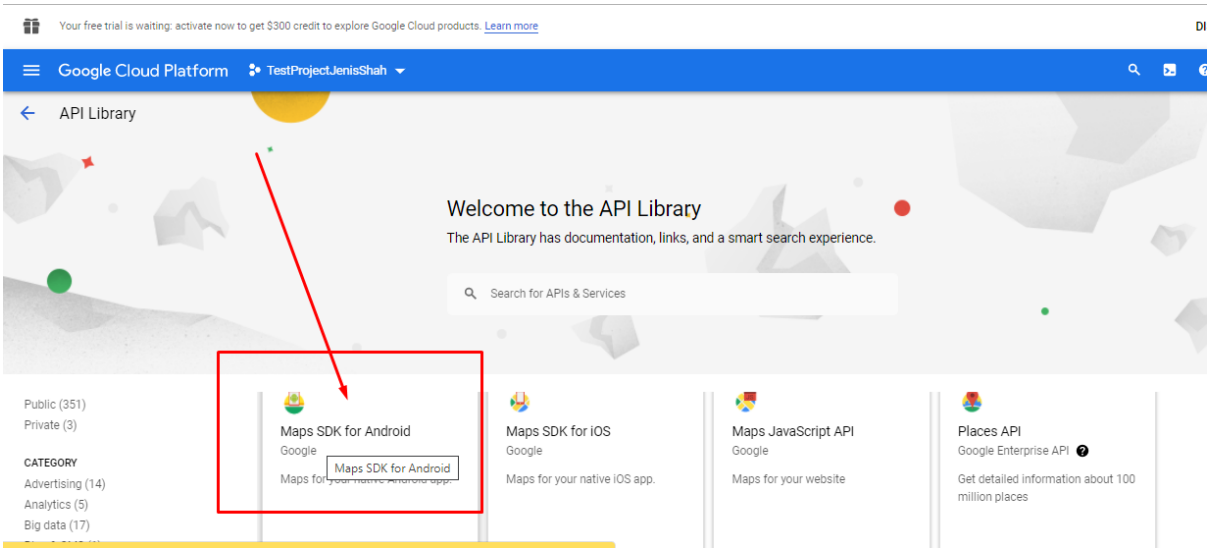
- Copy the API Key



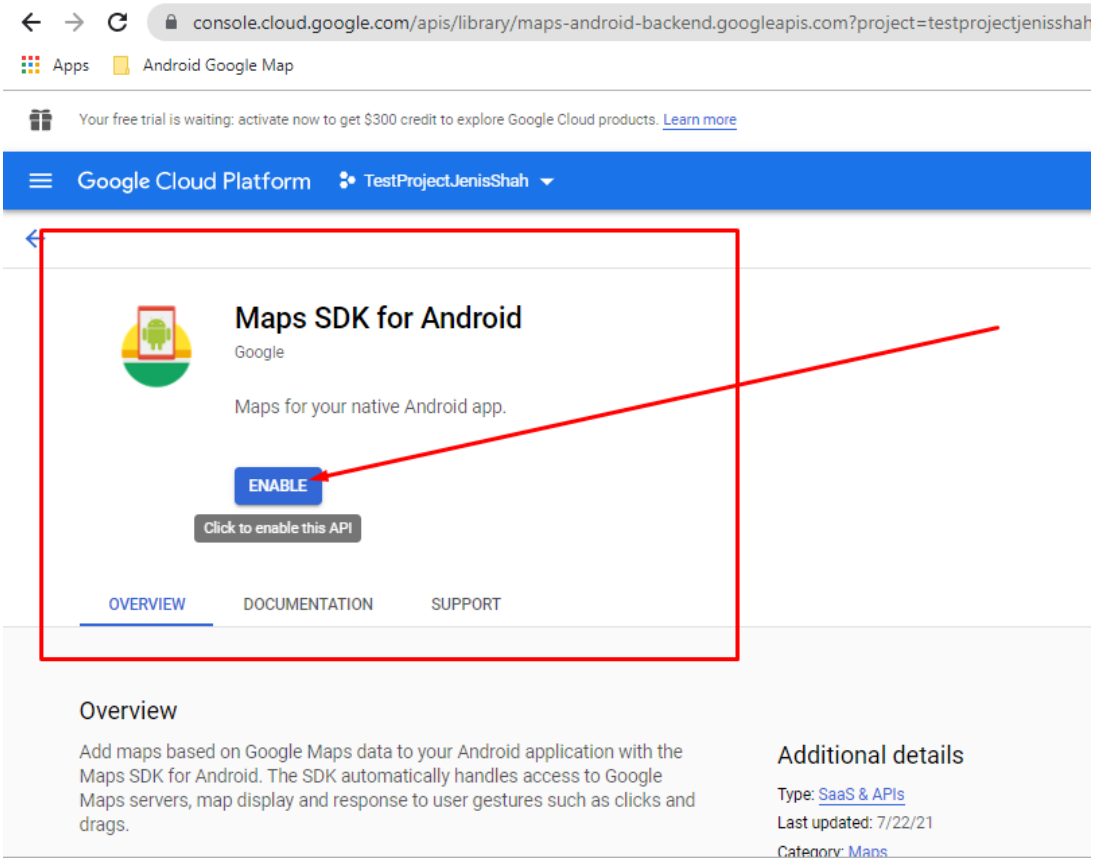
- Click on Library To Enable MAPS SDK FOR ANDROID



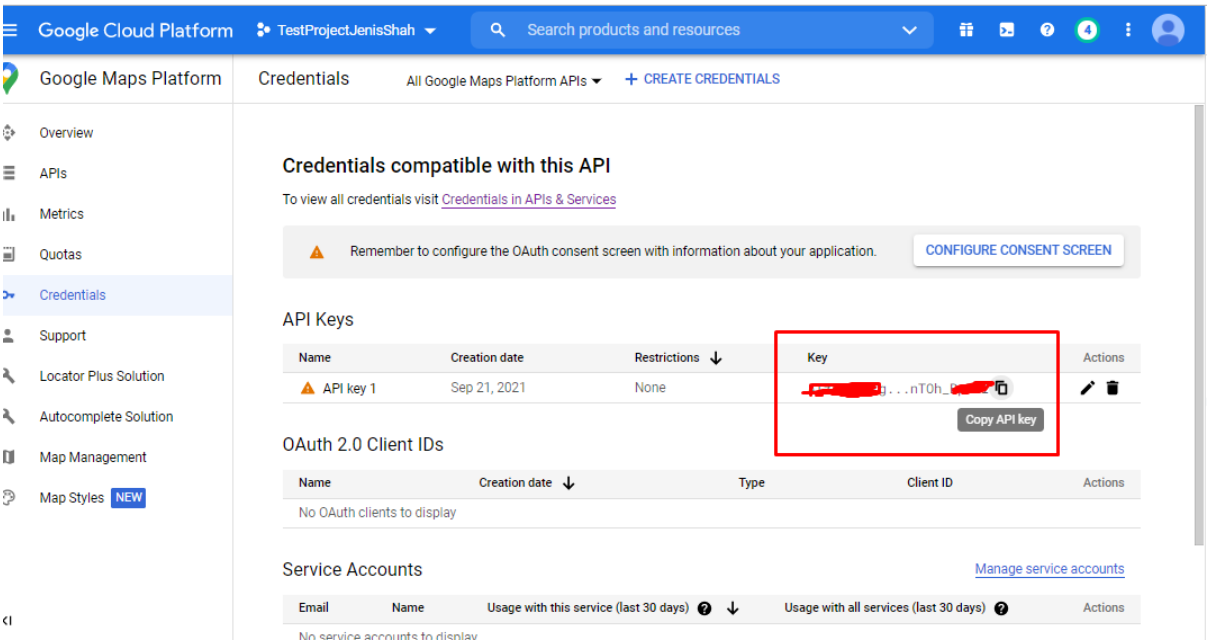
- Click on MAPS SDK FOR ANDROID



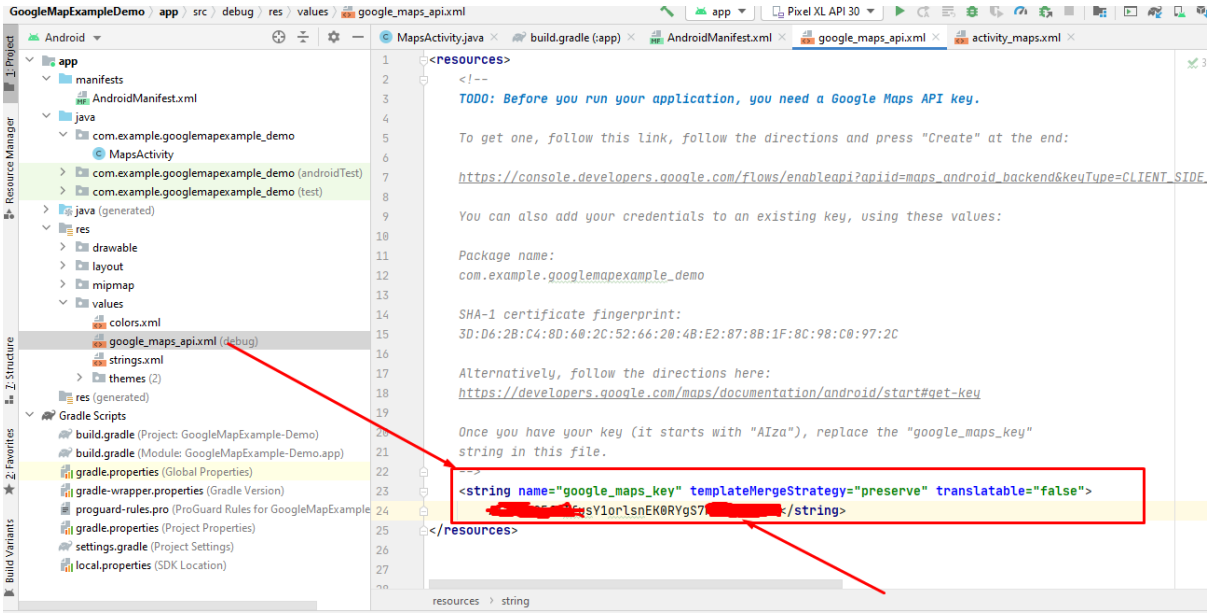
- ENABLE THE MAP SDK



- Copy the API Key



- Paste in to google_maps_api.xml



BINGO...!!! 😊 NOW You are READY .... !!!

Just Run the Application & Check the Output over there.

Location Based Services

- This becomes possible with the help of Google Play services, which facilitates adding location awareness to your app with automated location tracking, geofencing, and activity recognition.

- This tutorial shows you how to use Location Services in your APP to get the current location, get periodic location updates, look up addresses etc.

- The best live example is finding restaurants, petrol pumps or stores near you.

- Location-Based Services in Android provides us with this feature to help us in various ways. It enables us to create an application that is capable of detecting the current location of our devices. Android makes use of information from GPS and WiFi networks to get the location of the device on this Earth.

- So, LBS is the feature that Android provides us using the Location Framework. This framework provides us basically with certain classes and interfaces, which are key components. These components make it easier for us to implement the location feature in our application.

Components of Location-Based Services in Android

- LocationManager Class – It helps to get the location service of the System.
- LocationListener Interface – It receives notification from the Manager.
- LocationProvider – Devices provide a location from a set of providers.
- Location Class – It represents the geographic location at a particular time.

Location Object

- The location object stores the geographic location of the device in terms of Longitude, Latitude, Altitude, velocity, etc. Let us see some important methods that give locations specific information.

Methods of Location Object

1	**float distanceTo(Location dest)** Returns the approximate distance in meters between this location and the given location.
2	**float getAccuracy()** Get the estimated accuracy of this location, in meters.
3	**double getAltitude()** Get the altitude if available, in meters above sea level.
4	**float getBearing()** Get the bearing, in degrees.
5	**double getLatitude()** Get the latitude, in degrees.
6	**double getLongitude()** Get the longitude, in degrees.
7	**float getSpeed()** Get the speed if it is available, in meters/second over ground.
8	**boolean hasAccuracy()** True if this location has an accuracy.
9	**boolean hasAltitude()** True if this location has an altitude.
10	**boolean hasBearing()**

	True if this location has a bearing.
11	**boolean hasSpeed()** True if this location has a speed.
12	**void reset()** Clears the contents of the location.
13	**void setAccuracy(float accuracy)** Set the estimated accuracy of this location, meters.
14	**void setAltitude(double altitude)** Set the altitude, in meters above sea level.
15	**void setBearing(float bearing)** Set the bearing, in degrees.
16	**void setLatitude(double latitude)** Set the latitude, in degrees.
17	**void setLongitude(double longitude)** Set the longitude, in degrees.
18	**void setSpeed(float speed)** Set the speed, in meters/second over ground.
19	**String toString()** Returns a string containing a concise, human-readable description of this object

Get the Current Location

- To get the current location, create a location client which is LocationClient object, connect it to Location Services using connect() method, and then call its getLastLocation() method.
- This method returns the most recent location in the form of Location object that contains latitude and longitude coordinates and other information as explained above.
- To have location based functionality in your activity, you will have to implement two interfaces –

- GooglePlayServicesClient.ConnectionCallbacks
- GooglePlayServicesClient.OnConnectionFailedListener
- These interfaces provide following important callback methods, which you need to implement in your activity class

1	**abstract void onConnected(Bundle connectionHint)** This callback method is called when location service is connected to the location client successfully. You will use **connect()** method to connect to the location client.
2	**abstract void onDisconnected()** This callback method is called when the client is disconnected. You will use **disconnect()** method to disconnect from the location client.
3	**abstract void onConnectionFailed(ConnectionResult result)** This callback method is called when there was an error connecting the client to the service.
4	**abstract void onLocationChanged(Location location)** This callback method is used for receiving notifications from the LocationClient when the location has changed.

Location Quality of Service

- The LocationRequest object is used to request a quality of service (QoS) for location updates from the LocationClient.
- There are following useful setter methods which you can use to handle QoS.
- There are equivalent getter methods available which you can check in Android official documentation.

Methods of LOS

1	**setExpirationDuration(long millis)** Set the duration of this request, in milliseconds.
2	**setExpirationTime(long millis)** Set the request expiration time, in millisecond since boot.
3	**setFastestInterval(long millis)** Explicitly set the fastest interval for location updates, in milliseconds.
4	**setInterval(long millis)** Set the desired interval for active location updates, in milliseconds.
5	**setNumUpdates(int numUpdates)** Set the number of location updates.

Geocoding & Reverse Geocoding In Android

- Android Geocoder class is used for Geocoding as well as Reverse Geocoding.
- Geocoding refers to **transforming street address or any address into latitude and longitude.**
- Reverse Geocoding refers to **transforming latitude and longitude into its corresponding street address**
- **Address class** helps in fetching the **street address, locality, sub-locality, city, country, landmark etc.** features of the location.
- Using the above two classes we'll be fetching the current marker address on the Google Maps in our application.

How to Add Marker

```
LatLng latLng = new
LatLng(location.getLatitude(),location.getLongitude());

MarkerOptions markerOptions = new
MarkerOptions().position(latLng).title("LJIET DEMO Locaiton");
googleMap.addMarker(markerOptions).showInfoWindow();
```

How to Change Map Type

```
googleMap.setMapType(GoogleMap.MAP_TYPE_NORMAL);
googleMap.setMapType(GoogleMap.MAP_TYPE_HYBRID);
googleMap.setMapType(GoogleMap.MAP_TYPE_SATELLITE);
googleMap.setMapType(GoogleMap.MAP_TYPE_TERRAIN);
```

How to Enable/Disable Zoom

```
googleMap.getUiSettings().setZoomGesturesEnabled(true);
```

How to enable camera animation for Zoom up to 14

```
googleMap.animateCamera(CameraUpdateFactory.newLatLngZoom(latLng
,14));
```

Practical 1:

AIM:

Create the Google Map Project By Selecting the **Empty Activity.**

We get current location & set our Marker over there with animate camera effect.

Note: Here the process of generating the Google MAP API key are already mentioned above from Page no : (7 – 13)

Add Permissions & Necessary Key as well as uses-Library & gms.version

Dependencies like :

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

implementation 'com.google.android.gms:play-services-location:17.0.0'

Permissions Like:

```
<uses-permission android:name="android.permission.ACCESS_FINE_LOCATION"/>
<uses-permission android:name="android.permission.INTERNET"/>
<uses-permission android:name="android.permission.ACCESS_COARSE_LOCATION"/>
```

Manifest.xml File

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.example.googlemapexample1">
```

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

    <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/Theme.GoogleMapExample1">
        <activity android:name=".MainActivity">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />

                <category android:name="android.intent.category.LAUNCHER"
/>
            </intent-filter>
        </activity>

        <meta-data
            android:name="com.google.android.gms.version"
            android:value="@integer/google_play_services_version" />

        <uses-library
            android:name="org.apache.http.legacy"
            android:required="false" />
        <meta-data
            android:name="com.google.android.geo.API_KEY"
            android:value=" Write Your Key Here --ASY74BvEgC1v0btV—Write
Your Key Here" />
    </application>

</manifest>
```

Activity_main.xml File

Creating a Relative layout & take Fragment which have **`com.google.android.gms.maps.SupportMapFragment`** in it.

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".MainActivity">

    <fragment
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:id="@+id/maps"
        android:name="com.google.android.gms.maps.SupportMapFragment"/>
```

```
</RelativeLayout>
```

MainActivity.Java File

```
package com.example.googlemapexample1;

import androidx.annotation.NonNull;
import androidx.appcompat.app.AppCompatActivity;
import androidx.core.app.ActivityCompat;

import android.Manifest;
import android.content.pm.PackageManager;
import android.location.Location;
import android.os.Bundle;
import android.widget.Toast;

import com.google.android.gms.location.FusedLocationProviderClient;
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.OnSuccessListener;
import com.google.android.gms.tasks.Task;

public class MainActivity extends AppCompatActivity {


    private SupportMapFragment supportMapFragment;
    private FusedLocationProviderClient client;
    private int Request_Code = 110;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        supportMapFragment = (SupportMapFragment)
getSupportFragmentManager().findFragmentById(R.id.maps);
        client =
LocationServices.getFusedLocationProviderClient(MainActivity.this);


        if (ActivityCompat.checkSelfPermission(MainActivity.this,
Manifest.permission.ACCESS_FINE_LOCATION) ==
PackageManager.PERMISSION_GRANTED) {
            getCurrentLocation();
        }
        else{

            ActivityCompat.requestPermissions(MainActivity.this,new
String[]{Manifest.permission.ACCESS_FINE_LOCATION},Request_Code);
        }
    }
```

```
    private void getCurrentLocation() {
        if (ActivityCompat.checkSelfPermission(this,
Manifest.permission.ACCESS_FINE_LOCATION) !=
PackageManager.PERMISSION_GRANTED &&
                ActivityCompat.checkSelfPermission(this,
Manifest.permission.ACCESS_COARSE_LOCATION) !=
PackageManager.PERMISSION_GRANTED) {
            // TODO: Consider calling
            //    ActivityCompat#requestPermissions
            // here to request the missing permissions, and then overriding
            //   public void onRequestPermissionsResult(int requestCode,
String[] permissions,
            //                                          int[] grantResults)
            // to handle the case where the user grants the permission. See
the documentation
            // for ActivityCompat#requestPermissions for more details.
            return;
        }

        Task<Location> task = client.getLastLocation();

        task.addOnSuccessListener(new OnSuccessListener<Location>() {
            @Override
            public void onSuccess(Location location) {
                if(location != null)
                {
                    supportMapFragment.getMapAsync(new OnMapReadyCallback()
{
                        @Override
                        public void onMapReady(GoogleMap googleMap) {

                            LatLng latLng = new
LatLng(location.getLatitude(),location.getLongitude());
                            MarkerOptions markerOptions = new
MarkerOptions().position(latLng).title("LJIET DEMO Locaiton");

                            // for zoom map

googleMap.animateCamera(CameraUpdateFactory.newLatLngZoom(latLng,14));


googleMap.addMarker(markerOptions).showInfoWindow();

                        }
                    });
                }
            }
        });
    }

    @Override
    public void onRequestPermissionsResult(int requestCode, String[]
permissions, int[] grantResults) {

        super.onRequestPermissionsResult(requestCode, permissions,
grantResults);
        if (requestCode == Request_Code) {
            if (grantResults.length > 0 && grantResults[0] ==
PackageManager.PERMISSION_GRANTED) {
                getCurrentLocation();
            }
```

```
        }
        else{
            Toast.makeText(MainActivity.this,"Permission
denied",Toast.LENGTH_LONG).show();
        }
    }
}
```

Output

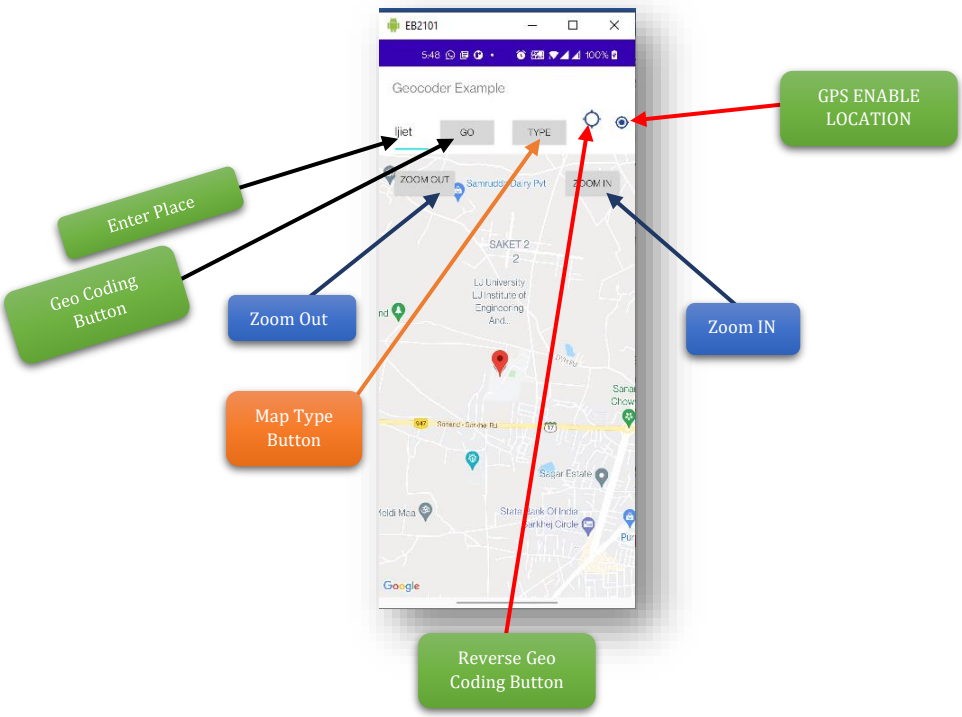
Practical 2:

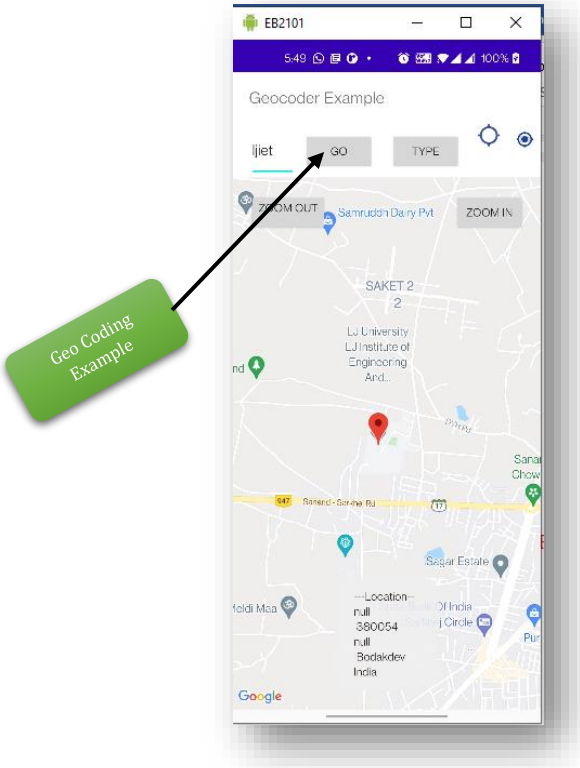
AIM : Create Project By Selecting the **Google Map Activity.**

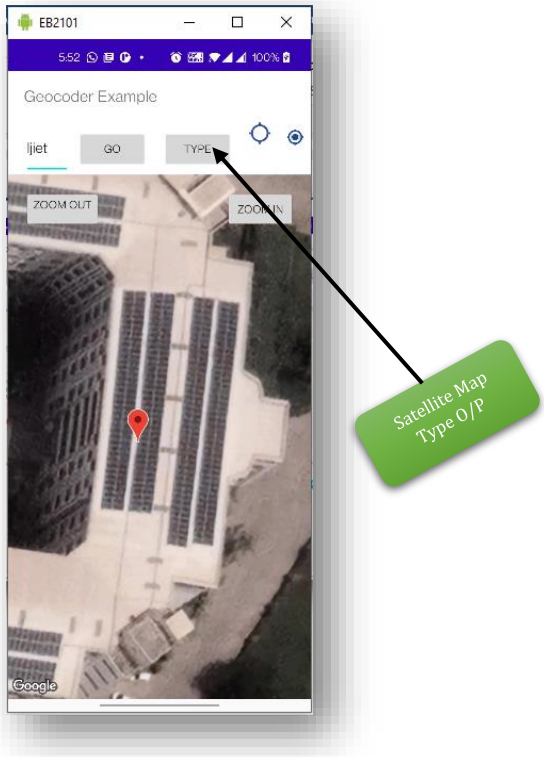
Here we are going to implement following functionality in a single program to understand every concept of Google Map.

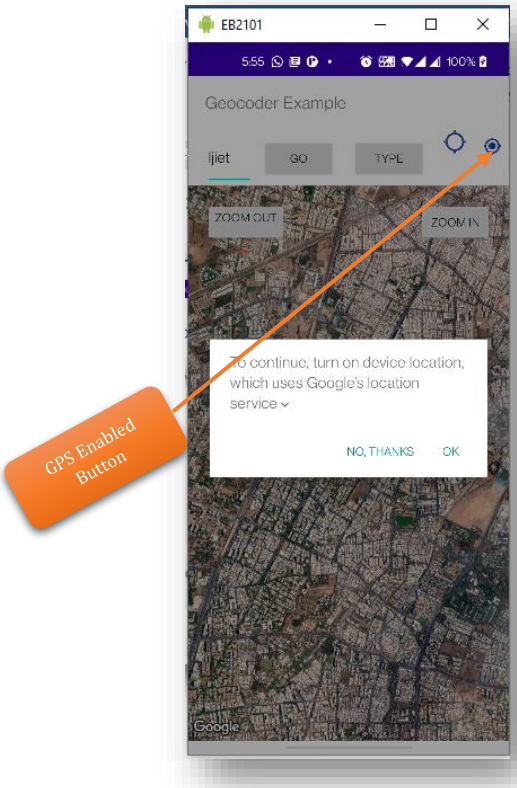
(Mostly All Concepts of Google Map Geo Coding Reverse Geo Coding Location Based Services and GPS have been covered in this SINGLE PROGRAM)

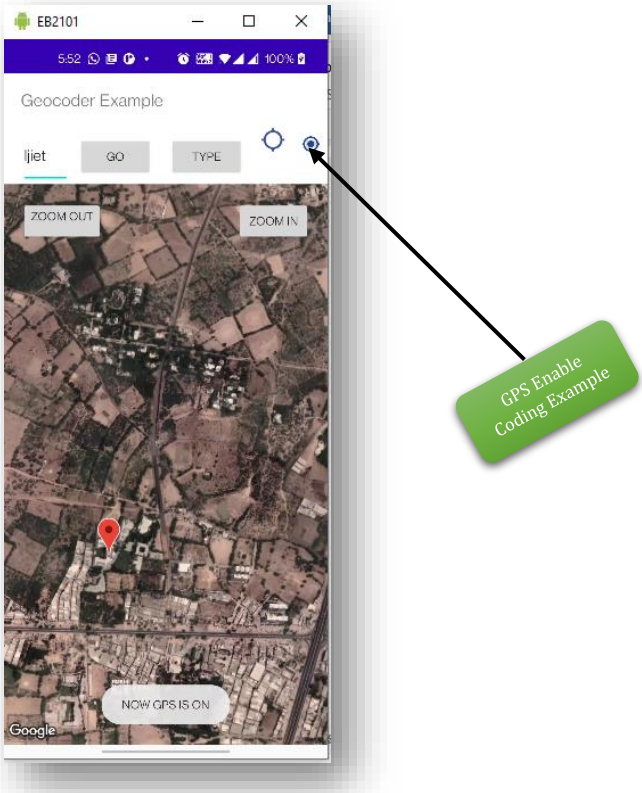
- Get Current Location
- Apply Zoom IN
- Apply Zoom Out
- Geocoding: (Get LatLong From Address) Here we need to Write Address & Fetch the Lat Long & Set Our Marker Over There On MAP
- Reverse Geo Coding: (Get Address From LatLong) I have already set the Iskon Cross Road LatLong over there in code. So, When we click on button It will give us City and Country Name.
- GPS Location Enabled & Get Current Location From It.
- Type of Map: We can change Map Type from Normal To Satellite & Vice Versa,
- **What we get as Output**




EB2101
Geocoder Example
ljiet
GO
TYPE
ZOOM OUT
ZOOM IN
Samruddh Dairy Pvt
SAKET 2
LJ University
LJ Institute of Engineering And...
Sagar Estate
Location
null
380054
null
Bodakdev
India
Google
Geo Coding Example


EB2101
Geocoder Example
ljiet
GO
TYPE
ZOOM OUT
ZOOM IN
Google
Satellite Map Type O/P


EB2101
Geocoder Example
ljiet
GO
TYPE
ZOOM OUT
ZOOM IN
To continue, turn on device location, which uses Google's location service
NO, THANKS
OK
Google
GPS Enabled Button


EB2101
Geocoder Example
ljiet
GO
TYPE
ZOOM OUT
ZOOM IN
NOW GPS IS ON
Google
GPS Enable Coding Example

Note: Here the process of generating the Google MAP API key are already mentioned above from Page no : (7 – 13)

Steps To Go With Google Map Activity as Default Activity File

A. Generate the API Key & Add it in google_maps_api.xml

B. Add Dependencies in Gradle File

C. Add Permissions to Manifest.xml File

D. Activity_maps.xml File

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

<LinearLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="vertical"
    android:layout_height="match_parent"
    android:layout_width="match_parent"
    >
    <TextView
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="Geocoder Example"
        android:textSize="20dp"
        android:layout_centerHorizontal="true"
        android:layout_margin="20dp"/>

    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:orientation="horizontal"
        >
        <EditText
            android:layout_width="0dp"
            android:hint="Enter Here "
            android:layout_height="wrap_content"
            android:layout_weight="1"

            android:layout_marginStart="20dp"
            android:id="@+id/searchEditText"/>
        <Button
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:id="@+id/goButton"
            android:layout_marginStart="10dp"
            android:layout_marginEnd="20dp"
            android:text="GO"/>
```

```
            <Button
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:text="type"
                android:layout_marginEnd="20dp"
                android:id="@+id/typeButton"
                />
            <ImageView android:layout_gravity="right"
                android:layout_width="30dp"
                android:layout_height="30dp"
                android:layout_marginEnd="20dp"
                android:layout_marginTop="10dp"
                android:id="@+id/getNameImageView"
                android:src="@drawable/ic_baseline_location_searching_24"
                />

            <ImageButton
                android:layout_width="20dp"
                android:layout_height="20dp"
                android:layout_marginTop="10dp"
                android:id="@+id/gpsImageButton"
                android:src="@drawable/ic_baseline_my_location_24"
                android:layout_marginEnd="10dp"
                />
        </LinearLayout>
        <FrameLayout
            android:layout_width="match_parent"
            android:layout_height="match_parent">

            <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" />

            <Button
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:text="Zoom In"
                android:layout_gravity="right"
                android:id="@+id/zoomInButton"
                android:layout_marginEnd="20dp"
                android:layout_marginTop="20dp"/>

            <Button
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:text="Zoom Out"
                android:id="@+id/zoomOutButton"
                android:gravity="right"
                android:layout_marginStart="20dp"
                android:layout_marginTop="20dp"/>

        </FrameLayout>

    </LinearLayout>
```

E. Basic Codes For Our Functionality

- Function for Geo Coding (Button)
  When User Clicks on Go Button. If user enter any data in Edit Text then It will Call The functions of GEO CODING. It shows the address of the Location Name that You have entered in the Edit Text.

```
goBtn.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        showAddress();
    }
});



private void showAddress() {

    String data = searchEdt.getText().toString();
    List<Address> addressList = null;

    if(data!=null || !data.equals("")){
        Geocoder geocoder = new Geocoder(this);
        try {
            addressList = geocoder.getFromLocationName(data,1);
        } catch (IOException e) {
            e.printStackTrace();
        }
        Address address = addressList.get(0);
        LatLng latLng = new
LatLng(address.getLatitude(),address.getLongitude());

 MarkerOptions markerOptions= new
MarkerOptions().position(latLng).title("Resulted Location
Marker");
        mMap.addMarker(markerOptions);

mMap.animateCamera(CameraUpdateFactory.newLatLngZoom(latLng,14)
);

    }
}
```

- Function for ZoomOut (Button)

```
zoomOutBtn.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        zoomOUT();
    }
});


private void zoomOUT()
{
    mMap.animateCamera(CameraUpdateFactory.zoomOut());
}
```

- Function for ZoomIn (Button)

```
zoomInBtn.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        zoomIN();
    }
});


private void zoomIN()
{
    mMap.animateCamera(CameraUpdateFactory.zoomIn());
}
```

- Function for MapType

```
typeBtn.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {

        mapType();
    }
});

private void mapType()
{
    if(mMap.getMapType()==GoogleMap.MAP_TYPE_NORMAL){
        mMap.setMapType(GoogleMap.MAP_TYPE_SATELLITE);
    }
    else{
        mMap.setMapType(GoogleMap.MAP_TYPE_NORMAL);
    }
}
```

- Function For Reverse Geo Coding (ImageView)

```
getName.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        Geocoder geocoder = new Geocoder(MapsActivity.this);
        try {
            List<Address> addressList =
geocoder.getFromLocation(23.02849078664996, 72.50681763309379,
3);
            if (addressList.size() > 0) {
                String countryName =
addressList.get(0).getCountryName();
                String addressLine2 =
addressList.get(0).getAddressLine(2);
                String address =
addressList.get(0).getAddressLine(1);
                String postalcode =
addressList.get(0).getPostalCode();
                String locality =
addressList.get(0).getSubLocality();
                Toast.makeText(MapsActivity.this, "---Location-
- \n"+addressLine2 + "\n "+ postalcode+ "\n" + address + "\n "
+ locality + "\n" + countryName, Toast.LENGTH_LONG).show();
            }
        } catch (IOException e) {
            e.printStackTrace();
        }
    }
});
```

- Functionality for GPS Button : To Enable the GPS Connection Without Opening the Setting of Phone & Get the Current Location as well as Set the Marker Over there.

```
locationRequest = LocationRequest.create();
locationRequest.setPriority(LocationRequest.PRIORITY_HIGH_ACCUR
ACY);
locationRequest.setInterval(5000);
locationRequest.setFastestInterval(3000);

gpsButton.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {

        getCurrentLocation();

    }
});
```

// function for Get Current Location

```
private void getCurrentLocation() {

    if (ActivityCompat.checkSelfPermission(MapsActivity.this,
Manifest.permission.ACCESS_FINE_LOCATION) ==
PackageManager.PERMISSION_GRANTED) {

        if(isGPSEnabled())
        {


LocationServices.getFusedLocationProviderClient(MapsActivity.thi
s)
                    .requestLocationUpdates(locationRequest, new
LocationCallback(){
                        @Override
                        public void
onLocationResult(LocationResult locationResult) {

super.onLocationResult(locationResult);


LocationServices.getFusedLocationProviderClient(MapsActivity.thi
s)
                                    .removeLocationUpdates(this);
                            if(locationResult!=null &&
locationResult.getLocations().size()>0)
                            {

                                int index =
locationResult.getLocations().size()-1;

                                double latitude  =
locationResult.getLocations().get(index).getLatitude();
                                double longitude =
locationResult.getLocations().get(index).getLongitude();

                                LatLng latLng = new
LatLng(latitude,longitude);
                                MarkerOptions marker = new
MarkerOptions().position(latLng).title("NEW CURRENT LOCATION");


mMap.animateCamera(CameraUpdateFactory.newLatLngZoom(latLng,14))
;

mMap.addMarker(marker).showInfoWindow();
                                Log.i("TEXT","Test");

                            }

                        }
                    }, Looper.getMainLooper());
        }
        else{
            turnOnGPS();
        }
    }
```

```
    else{
        if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.M) {
            requestPermissions(new
String[]{Manifest.permission.ACCESS_FINE_LOCATION},1);
        }
    }

}
```

// Function for Turn on GPS that has used on Above Function

```
private void turnOnGPS() {
    LocationSettingsRequest.Builder builder = new
LocationSettingsRequest.Builder()
            .addLocationRequest(locationRequest)
            .setAlwaysShow(true);
    Task<LocationSettingsResponse> result =
LocationServices.getSettingsClient(getApplicationContext())
            .checkLocationSettings(builder.build());

    result.addOnCompleteListener(new
OnCompleteListener<LocationSettingsResponse>() {
        @Override
        public void onComplete(Task<LocationSettingsResponse>
task) {

            try {
                LocationSettingsResponse
locationSettingsResponse = task.getResult(ApiException.class);
                Toast.makeText(MapsActivity.this,"GPS
ENABLED",Toast.LENGTH_LONG).show();
            } catch (ApiException e) {
                if(e.getStatusCode() ==
LocationSettingsStatusCodes.RESOLUTION_REQUIRED)
                {
                    ResolvableApiException
resolvableApiException = (ResolvableApiException) e;
                    try {

resolvableApiException.startResolutionForResult(MapsActivity.th
is,100);
                    } catch (IntentSender.SendIntentException
sendIntentException) {
                        sendIntentException.printStackTrace();
                    }
                }
                if(e.getStatusCode() ==
LocationSettingsStatusCodes.SETTINGS_CHANGE_UNAVAILABLE)
                {
                    Toast.makeText(MapsActivity.this,"GPS
SETTING NOT AAVAILABLE",Toast.LENGTH_LONG).show();

                }
            }
        }
    });
```

```
}
```

// Functions for checking is GPS enable or Not ?

```
private boolean isGPSEnabled() {
    LocationManager locationManager = null;
    boolean isEnabled = false;
    if(locationManager == null)
    {
        locationManager = (LocationManager)
getSystemService(LOCATION_SERVICE);

    }
    isEnabled =
locationManager.isProviderEnabled(LocationManager.GPS_PROVIDER)
;
    return  isEnabled;
}
```

//Function to check Permission

```
@Override
public void onRequestPermissionsResult(int requestCode,
String[] permissions, int[] grantResults) {
    super.onRequestPermissionsResult(requestCode, permissions,
grantResults);
    if(requestCode == 1)
    {
        if(grantResults[0] ==
PackageManager.PERMISSION_GRANTED)
        {
            if(isGPSEnabled())
            {
                getCurrentLocation();
            }
            else{
                turnOnGPS();
            }
        }
    }
}
```

// On Activity Result

```
@Override
protected void onActivityResult(int requestCode, int
resultCode,  Intent data) {
    super.onActivityResult(requestCode,resultCode,data);
    if(requestCode == 100){
```

```
            if(resultCode==RESULT_OK)
            {
                getCurrentLocation();
                Toast.makeText(this,"NOW GPS IS
ON",Toast.LENGTH_SHORT).show();

            }
            else {
                Toast.makeText(this,"DENIED GPS
PERMISSION",Toast.LENGTH_SHORT).show();
            }

        }
    }
```

FINAL CODE FOR MAPSACTIVITY.JAVA

```
package com.example.googlemapexample_demo;

import androidx.annotation.NonNull;
import androidx.annotation.Nullable;
import androidx.core.app.ActivityCompat;
import androidx.fragment.app.FragmentActivity;

import android.Manifest;
import android.app.AlertDialog;
import android.content.DialogInterface;
import android.content.Intent;
import android.content.IntentSender;
import android.content.pm.PackageManager;
import android.location.Address;
import android.location.Geocoder;
import android.location.Location;
import android.location.LocationManager;
import android.location.LocationProvider;
import android.os.Build;
import android.os.Bundle;
import android.os.Looper;
import android.provider.Settings;
import android.util.Log;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.ImageButton;
import android.widget.ImageView;
import android.widget.Toast;

import com.google.android.gms.common.api.ApiException;
import com.google.android.gms.common.api.ResolvableApiException;
import com.google.android.gms.location.FusedLocationProviderClient;
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.location.LocationSettingsRequest;
import com.google.android.gms.location.LocationSettingsResponse;
import com.google.android.gms.location.LocationSettingsStates;
```

```
import com.google.android.gms.location.LocationSettingsStatusCodes;
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.Marker;
import com.google.android.gms.maps.model.MarkerOptions;
import com.example.googlemapexample_demo.databinding.ActivityMapsBinding;
import com.google.android.gms.tasks.OnCompleteListener;
import com.google.android.gms.tasks.Task;

import java.io.IOException;
import java.util.List;

public class MapsActivity extends FragmentActivity implements
OnMapReadyCallback {

    private GoogleMap mMap;
    EditText searchEdt;
    ImageView getName;
    ImageButton gpsButton;
    Button goBtn, zoomInBtn, zoomOutBtn, typeBtn;
    private ActivityMapsBinding binding;
    LocationRequest locationRequest;


    @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);
        searchEdt = findViewById(R.id.searchEditText);
        goBtn = findViewById(R.id.goButton);
        getName = findViewById(R.id.getNameImageView);
        gpsButton = findViewById(R.id.gpsImageButton);
        zoomInBtn = findViewById(R.id.zoomInButton);
        zoomOutBtn = findViewById(R.id.zoomOutButton);
        typeBtn = findViewById(R.id.typeButton);

        goBtn.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                showAddress();
            }
        });

        typeBtn.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {

                mapType();
            }
```

```
        });

        zoomOutBtn.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                zoomOUT();
            }
        });

        zoomInBtn.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                zoomIN();
            }
        });

        getName.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                Geocoder geocoder = new Geocoder(MapsActivity.this);
                try {
                    List<Address> addressList =
geocoder.getFromLocation(23.02849078664996, 72.50681763309379, 3);
                    if (addressList.size() > 0) {
                        String countryName =
addressList.get(0).getCountryName();
                        String addressLine2 =
addressList.get(0).getAddressLine(2);
                        String address =
addressList.get(0).getAddressLine(1);
                        String postalcode =
addressList.get(0).getPostalCode();
                        String locality =
addressList.get(0).getSubLocality();
                        Toast.makeText(MapsActivity.this, "---Location--
\n"+addressLine2 + "\n "+ postalcode+ "\n" + address + "\n " + locality +
"\n" + countryName, Toast.LENGTH_LONG).show();
                    }
                } catch (IOException e) {
                    e.printStackTrace();
                }
            }
        });

        locationRequest = LocationRequest.create();

locationRequest.setPriority(LocationRequest.PRIORITY_HIGH_ACCURACY);
        locationRequest.setInterval(5000);
        locationRequest.setFastestInterval(3000);

        gpsButton.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {

                getCurrentLocation();

            }
        });

    }
```

```
    private void getCurrentLocation() {

        if (ActivityCompat.checkSelfPermission(MapsActivity.this,
Manifest.permission.ACCESS_FINE_LOCATION) ==
PackageManager.PERMISSION_GRANTED) {

            if(isGPSEnabled())
            {


LocationServices.getFusedLocationProviderClient(MapsActivity.this)
                        .requestLocationUpdates(locationRequest, new
LocationCallback(){
                            @Override
                            public void onLocationResult(LocationResult
locationResult) {
                                super.onLocationResult(locationResult);


LocationServices.getFusedLocationProviderClient(MapsActivity.this)
                                        .removeLocationUpdates(this);
                                if(locationResult!=null &&
locationResult.getLocations().size()>0)
                                {

                                    int index =
locationResult.getLocations().size()-1;

                                    double latitude  =
locationResult.getLocations().get(index).getLatitude();
                                    double longitude =
locationResult.getLocations().get(index).getLongitude();

                                    LatLng latLng = new
LatLng(latitude,longitude);
                                    MarkerOptions marker = new
MarkerOptions().position(latLng).title("NEW CURRENT LOCATION");


mMap.animateCamera(CameraUpdateFactory.newLatLngZoom(latLng,14));
                                    mMap.addMarker(marker).showInfoWindow();
                                    Log.i("TEXT","Test");

                                }

                            }
                        }, Looper.getMainLooper());
            }
            else{
                turnOnGPS();
            }
        }
        else{
            if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.M) {
                requestPermissions(new
String[]{Manifest.permission.ACCESS_FINE_LOCATION},1);
            }
        }

    }
```

```
    private void turnOnGPS() {
        LocationSettingsRequest.Builder builder = new
LocationSettingsRequest.Builder()
                .addLocationRequest(locationRequest)
                .setAlwaysShow(true);
        Task<LocationSettingsResponse> result =
LocationServices.getSettingsClient(getApplicationContext())
                .checkLocationSettings(builder.build());

        result.addOnCompleteListener(new
OnCompleteListener<LocationSettingsResponse>() {
            @Override
            public void onComplete(Task<LocationSettingsResponse> task) {

                try {
                    LocationSettingsResponse locationSettingsResponse =
task.getResult(ApiException.class);
                    Toast.makeText(MapsActivity.this,"GPS
ENABLED",Toast.LENGTH_LONG).show();
                } catch (ApiException e) {
                    if(e.getStatusCode() ==
LocationSettingsStatusCodes.RESOLUTION_REQUIRED)
                    {
                        ResolvableApiException resolvableApiException =
(ResolvableApiException) e;
                        try {

resolvableApiException.startResolutionForResult(MapsActivity.this,100);
                        } catch (IntentSender.SendIntentException
sendIntentException) {
                            sendIntentException.printStackTrace();
                        }
                    }
                    if(e.getStatusCode() ==
LocationSettingsStatusCodes.SETTINGS_CHANGE_UNAVAILABLE)
                    {
                        Toast.makeText(MapsActivity.this,"GPS SETTING NOT
AAVAILABLE",Toast.LENGTH_LONG).show();

                    }
                }
            }
        });

    }

    private boolean isGPSEnabled() {
        LocationManager locationManager = null;
        boolean isEnabled = false;
        if(locationManager == null)
        {
            locationManager = (LocationManager)
getSystemService(LOCATION_SERVICE);

        }
        isEnabled =
locationManager.isProviderEnabled(LocationManager.GPS_PROVIDER);
        return  isEnabled;
    }

    @Override
```

```
    protected void onActivityResult(int requestCode, int resultCode,
Intent data) {
        super.onActivityResult(requestCode,resultCode,data);
        if(requestCode == 100){
            if(resultCode==RESULT_OK)
            {
                getCurrentLocation();
                Toast.makeText(this,"NOW GPS IS
ON",Toast.LENGTH_SHORT).show();

            }
            else {
                Toast.makeText(this,"DENIED GPS
PERMISSION",Toast.LENGTH_SHORT).show();
            }

        }
    }

    @Override
    public void onRequestPermissionsResult(int requestCode,  String[]
permissions, int[] grantResults) {
        super.onRequestPermissionsResult(requestCode, permissions,
grantResults);
        if(requestCode == 1)
        {
            if(grantResults[0] == PackageManager.PERMISSION_GRANTED)
            {
                if(isGPSEnabled())
                {
                    getCurrentLocation();
                }
                else{
                    turnOnGPS();
                }
            }
        }
    }

  @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));

    }


    private void showAddress() {

        String data = searchEdt.getText().toString();
        List<Address> addressList = null;

        if(data!=null || !data.equals("")){
            Geocoder geocoder = new Geocoder(this);
```

```
            try {
                addressList = geocoder.getFromLocationName(data,1);
            } catch (IOException e) {
                e.printStackTrace();
            }
            Address address = addressList.get(0);
            LatLng latLng = new
LatLng(address.getLatitude(),address.getLongitude());
            MarkerOptions markerOptions= new
MarkerOptions().position(latLng).title("Resulted Location Marker");
            mMap.addMarker(markerOptions);

mMap.animateCamera(CameraUpdateFactory.newLatLngZoom(latLng,14));

        }
    }

    private void mapType()
    {
        if(mMap.getMapType()==GoogleMap.MAP_TYPE_NORMAL){
            mMap.setMapType(GoogleMap.MAP_TYPE_SATELLITE);
        }
        else{
            mMap.setMapType(GoogleMap.MAP_TYPE_NORMAL);
        }
    }

    private void zoomIN()
    {
        mMap.animateCamera(CameraUpdateFactory.zoomIn());
    }

    private void zoomOUT()
    {
        mMap.animateCamera(CameraUpdateFactory.zoomOut());
    }
}
```

Reference Links :

1. https://developer.android.com/
2. https://console.cloud.google.com/
3. https://www.tutorialspoint.com/android/android_google_maps.htm#:~:text=Android%20allows%20us%20to%20integrate,map%20according%20to%20your%20choices.
4. https://www.javatpoint.com/android-google-map
5. https://www.tutlane.com/tutorial/android/android-google-maps-api-with-examples
6. https://console.cloud.google.com/projectselector2/apis/dashboard?supportedpurview=project
7. https://support.google.com/googleapi/answer/6158862?hl=en
8. https://www.tutorialspoint.com/android/android_location_based_services.htm