Name of Student: Pushkar Sane		
Roll Number: 45	**Lab Assignment Number: 7**	
Title of Lab Assignment: Android program to work with google maps and location and GPS		
DOP: 17-09-2024	**DOS: 19-10-2024**	
CO Mapped: **CO2, CO5**	**PO Mapped:** **PO2, PO3, PO5, PSO1, PSO2**	**Signature:**

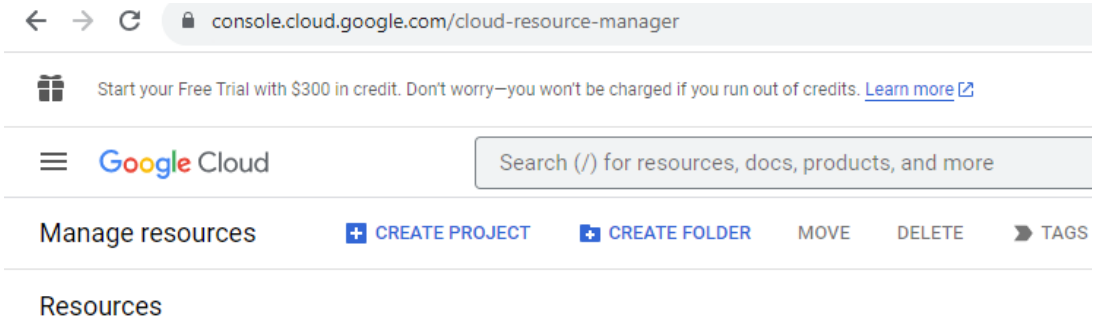
PRACTICAL 7

Aim: Android program to work with google maps and location "Add marker "method to be used in the application students are creating.

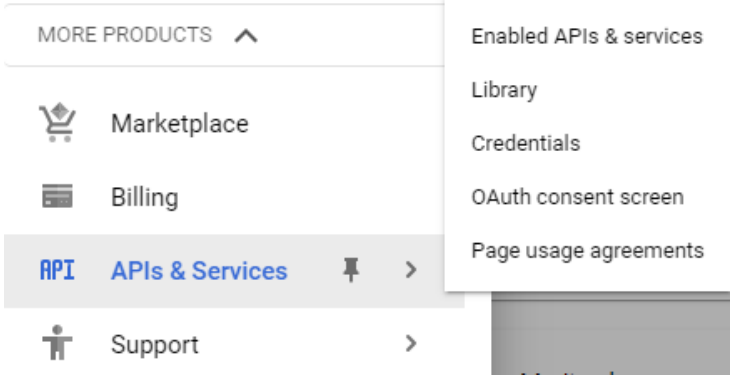
Theory:
Google Maps API: An API key is needed to access the Google Maps servers. This key is free, and you can use it with any of your applications. If you haven't created project, you can follow the below steps to get started:

Step 1: Open Google developer console and sign in with your gmail account: https://console.developers.google.com/project

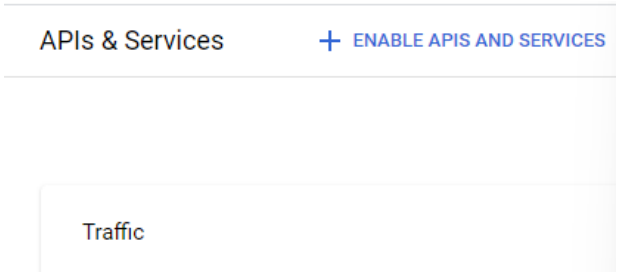
Step 2: Now create a new project. You can create a new project by clicking on the Create Project button and giving the name to your project.



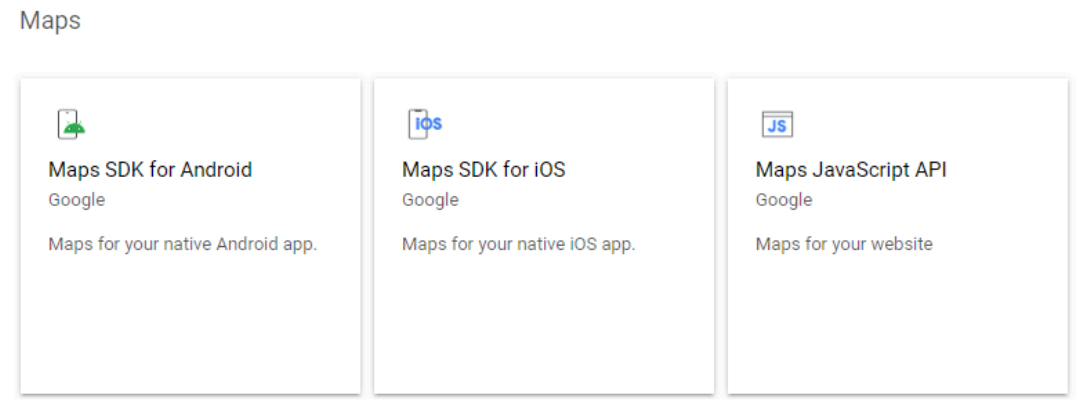
Step 3: Now click on APIs & Services and open Dashboard from it



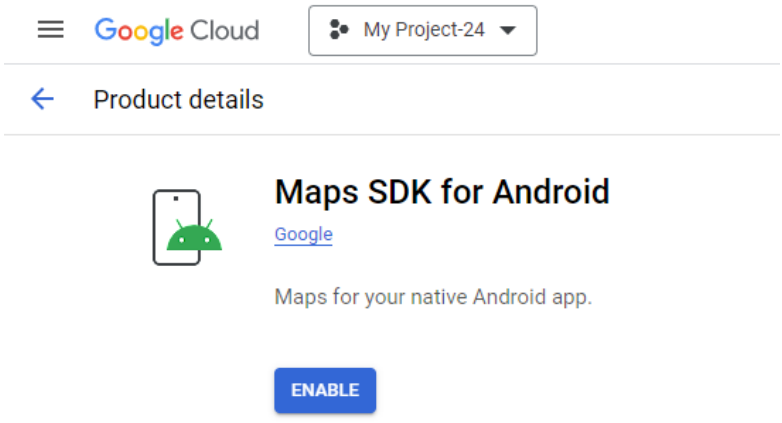
Step 4: In this open Enable APIS AND SERVICES



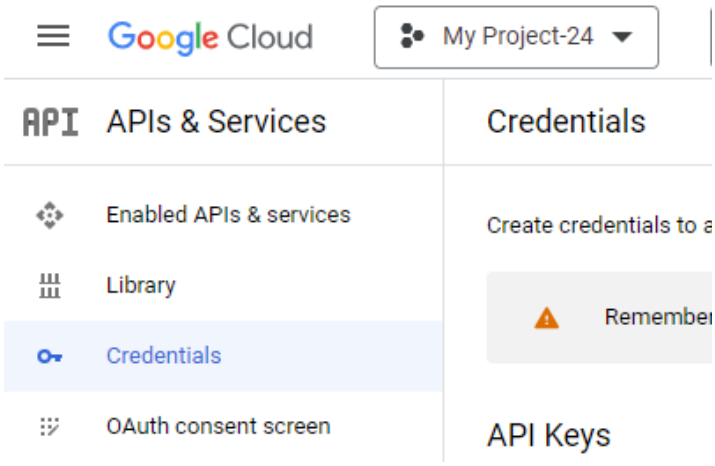
Step 5: Now open Map SDK for Android.



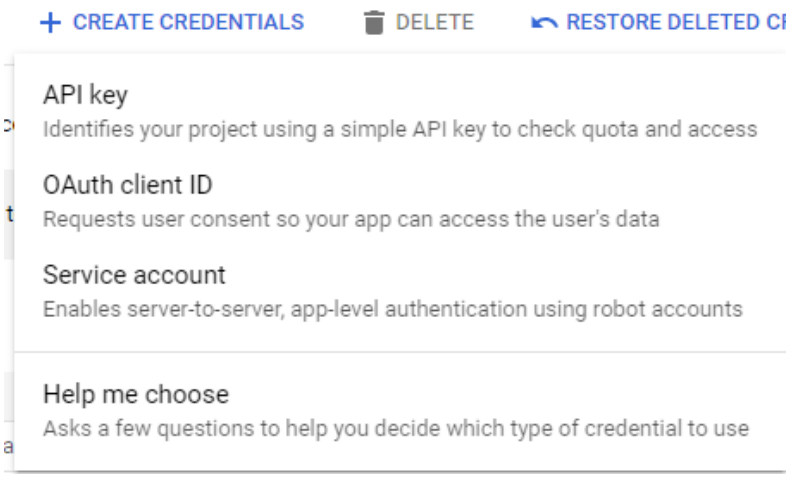
Step 6: Now enable the Google Maps Android API



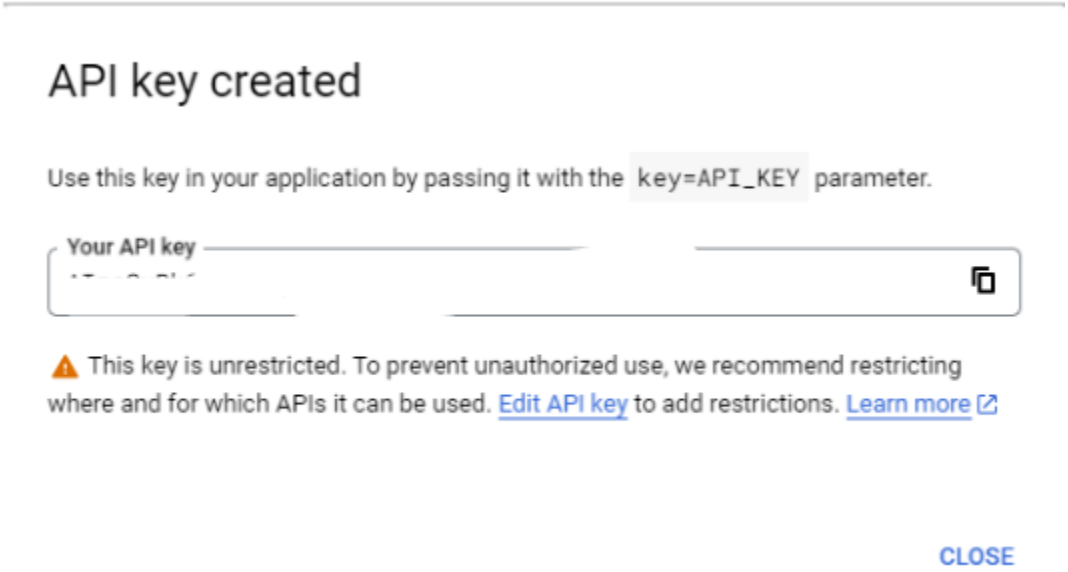
Step 7: Now go to Credentials



Step 8: Here click on Create credentials and choose API key



Step 9: Now your API key will be generated. Copy it and save it somewhere as we will need it when implementing Google Map in our Android project.



Strings.xml:

```
<resources>
  <string name="app_name">My Application_prac7</string>
  <string name="map_key"
translatable="false">AIzaSyBkQ7SsgYqle37tPs0BYLpXddFu6m0uCuk</string>
</resources>
```

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">
  <meta-data android:name="com.google.android.geo.API_KEY"
    android:value="@string/map_key"/>

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

```
  <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.MyApplication_prac7"
     tools:targetApi="31">
     <activity
        android:name=".MainActivity"
        android:exported="true">
        <intent-filter>
           <action android:name="android.intent.action.MAIN" />
           <category android:name="android.intent.category.LAUNCHER" />
        </intent-filter>
        <meta-data
           android:name="android.app.lib_name"
           android:value="" />
     </activity>
  </application>
</manifest>
```

Dependencies:

```
implementation 'com.google.android.libraries.maps:maps:3.1.0-beta'
implementation 'com.google.android.gms:play-services-maps:18.0.0'
implementation 'com.google.android.gms:play-services-location:18.0.0'
implementation 'com.google.maps.android:android-maps-utils:2.3.0'
```

activity_main.xml:

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
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:id="@+id/google_map"
    android:name="com.google.android.gms.maps.SupportMapFragment"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:ignore="MissingClass" />
</androidx.constraintlayout.widget.ConstraintLayout>
```

MainActivity.java:

```
package com.example.myapplication_prac7;

import androidx.appcompat.app.AppCompatActivity;
import android.Manifest;
import android.annotation.SuppressLint;
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;
import androidx.annotation.NonNull;
import androidx.appcompat.app.AppCompatActivity;
import android.os.Bundle;
import androidx.core.app.ActivityCompat;

public class MainActivity extends AppCompatActivity implements OnMapReadyCallback{
    Location currentLocation;
    GoogleMap gMap;
    FusedLocationProviderClient fusedLocationProviderClient;
    private static final int REQUEST_CODE = 101;

  @Override
  protected void onCreate(Bundle savedInstanceState) {
```

```
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);
    fusedLocationProviderClient = LocationServices.getFusedLocationProviderClient(this);
    fetchlastLocation();
  }
  private void fetchlastLocation() {
    if
(ActivityCompat.checkSelfPermission(this,Manifest.permission.ACCESS_FINE_LOCATION) !=
        PackageManager.PERMISSION_GRANTED &&
        ActivityCompat.checkSelfPermission(this,
            Manifest.permission.ACCESS_COARSE_LOCATION) !=
            PackageManager.PERMISSION_GRANTED) {
      ActivityCompat.requestPermissions(this, new String[]
          {Manifest.permission.ACCESS_FINE_LOCATION},REQUEST_CODE);
      return;
    }

    Task<Location> task = fusedLocationProviderClient.getLastLocation();
    task.addOnSuccessListener(new OnSuccessListener<Location>() {
      @Override
      public void onSuccess(Location location) {
        if(location != null){
          currentLocation = location;
          Toast.makeText(getApplicationContext(),currentLocation.getLatitude()
+""+currentLocation.getLongitude(), Toast.LENGTH_SHORT).show();
          SupportMapFragment supportMapFragment = (SupportMapFragment)
getSupportFragmentManager().findFragmentById(R.id.google_map);
          supportMapFragment.getMapAsync(MainActivity.this);
        }
      }
    });
  }
  @Override
  public void onMapReady(@NonNull GoogleMap googleMap) {
    double srcLat = currentLocation.getLatitude();
    double srcLong = currentLocation.getLongitude();
    LatLng latLng = new
LatLng(currentLocation.getLatitude(),currentLocation.getLongitude());
    MarkerOptions markerOptions = new MarkerOptions().position(latLng).title(srcLat + ":" +
srcLong);
```

```
    googleMap.animateCamera(CameraUpdateFactory.newLatLng(latLng));
    googleMap.animateCamera(CameraUpdateFactory.newLatLngZoom(latLng,10));
    googleMap.addMarker(markerOptions);
    gMap = googleMap;
    gMap.setOnMapClickListener(new GoogleMap.OnMapClickListener() {
      int count = -1;
      @Override
      public void onMapClick(@NonNull LatLng latLng) {
        MarkerOptions markerOptions1 = new MarkerOptions();
        markerOptions1.position(latLng);
        markerOptions1.title(latLng.latitude+ ":" + latLng.longitude);
        count++;
        if(count % 2 != 0){
          gMap.clear();
        }
        gMap.animateCamera(CameraUpdateFactory.newLatLng(latLng));
        gMap.animateCamera(CameraUpdateFactory.newLatLngZoom(latLng,10));
        gMap.addMarker(markerOptions1);
        double res = distance(srcLat,latLng.latitude,srcLong,latLng.longitude);
        Toast.makeText(getApplicationContext(), "Distance = "+ res,
Toast.LENGTH_SHORT).show();
      }
    });
  }
  public static double distance(double lat1, double lat2, double
      lon1,double lon2)
  {
    lon1 = Math.toRadians(lon1);
    lon2 = Math.toRadians(lon2);
    lat1 = Math.toRadians(lat1);
    lat2 = Math.toRadians(lat2);
    double dlon = lon2 - lon1;
    double dlat = lat2 - lat1;
    double a = Math.pow(Math.sin(dlat / 2), 2) + Math.cos(lat1) * Math.cos(lat2) *
Math.pow(Math.sin(dlon / 2),2);
    double c = 2 * Math.asin(Math.sqrt(a));
    double r = 6371;

    return(c * r);
  }
```

```
  @SuppressLint("MissingSuperCall")
  @Override
  public void onRequestPermissionsResult(int requestCode, @NonNull String[] permissions,
@NonNull int[] grantResults) {
      switch (requestCode) {
         case REQUEST_CODE:
            if (grantResults.length > 0 && grantResults[0] ==
PackageManager.PERMISSION_GRANTED);
            fetchlastLocation();
            break;
      }
  }
}
```

Output:

Conclusion:I have successfully implemented an android program to work with google maps and location .