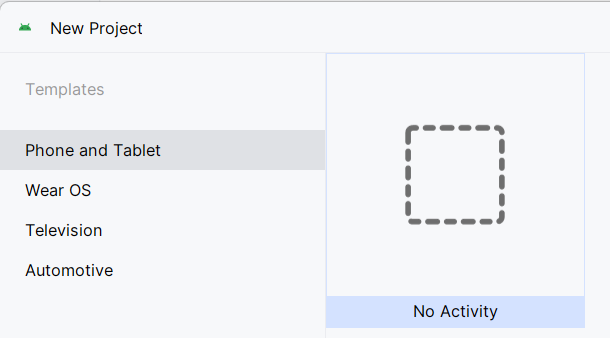
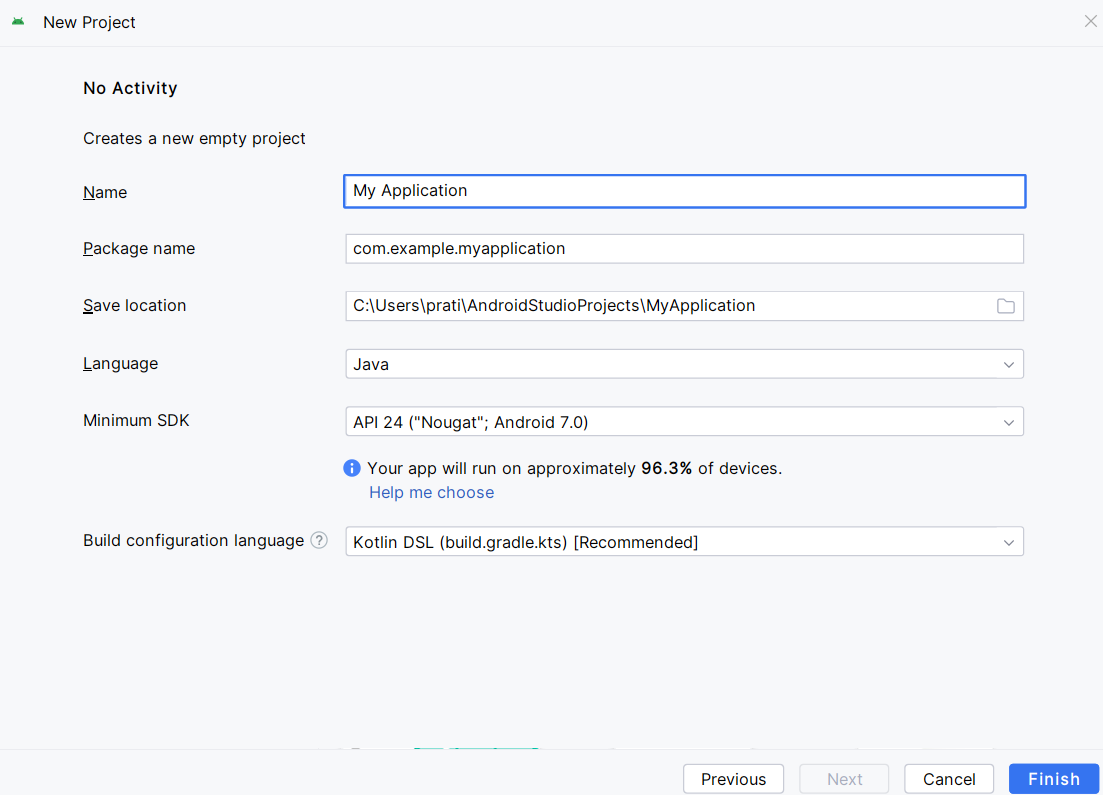
PRACTICAL 7

**AIM:** Android App with Google Map Service.

**Scenario:** Create an Android App which implements Google Map Service as full screen with a marker set using static Latitude and Longitude values.

This practical is using [Android Studio Hedgehog](https://developer.android.com/studio/install.html) and the [Android Gradle plugin](https://developer.android.com/build/releases/gradle-plugin) version 8.2.

1. Create a new android project => Phone and Tablet => No Activity

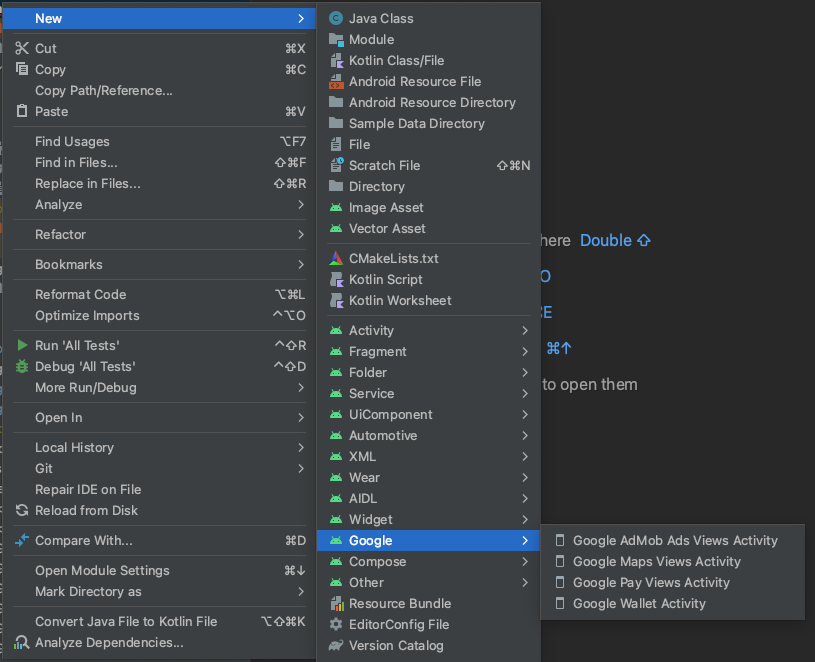
1. Android Studio starts Gradle and builds the project. This may take some time.

Add the **Google Maps Views Activity**:

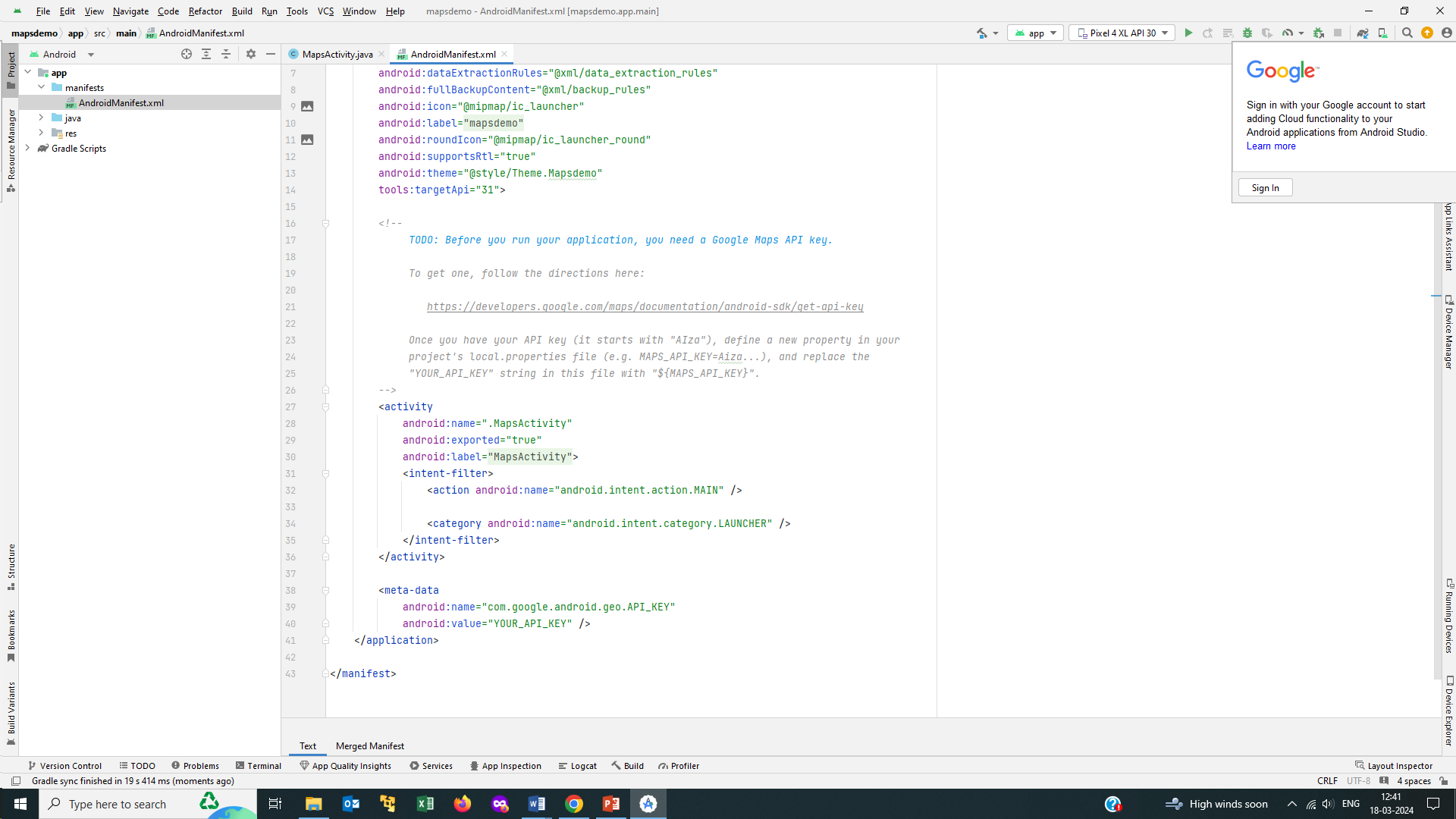
* 1. Right-click on the app folder in your project.

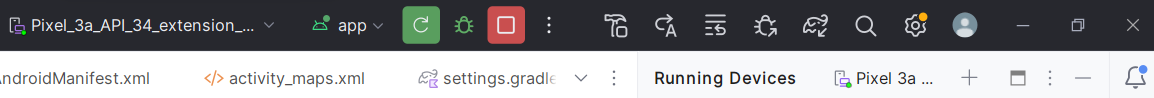
Select **New > Google > Google Maps Views Activity**.

* 1. In the **New Android Activity** dialog box, select the **Launcher Activity** checkbox. Select **Finish**.

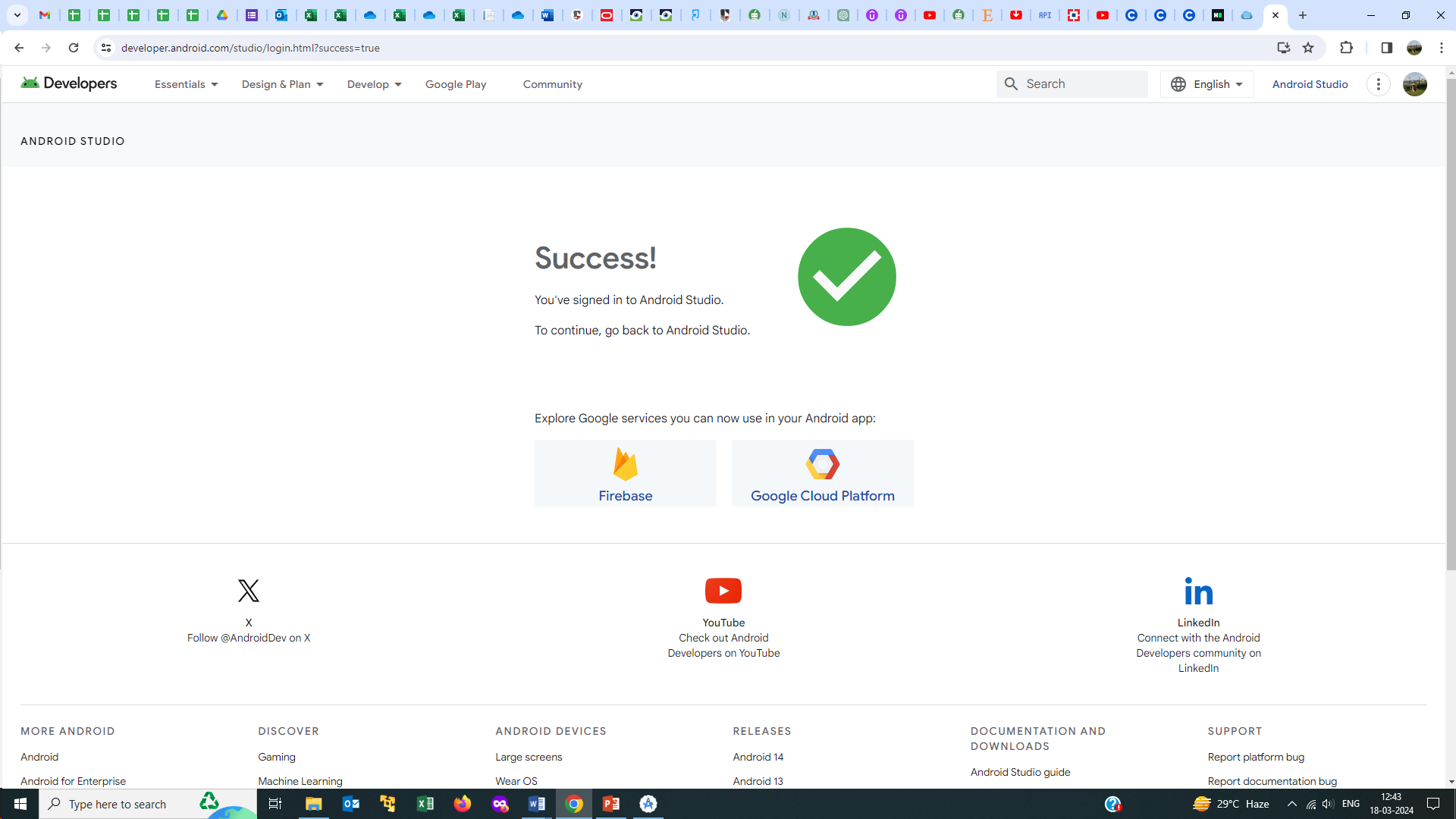


1. In Android Studio Project Login to Gmail Account

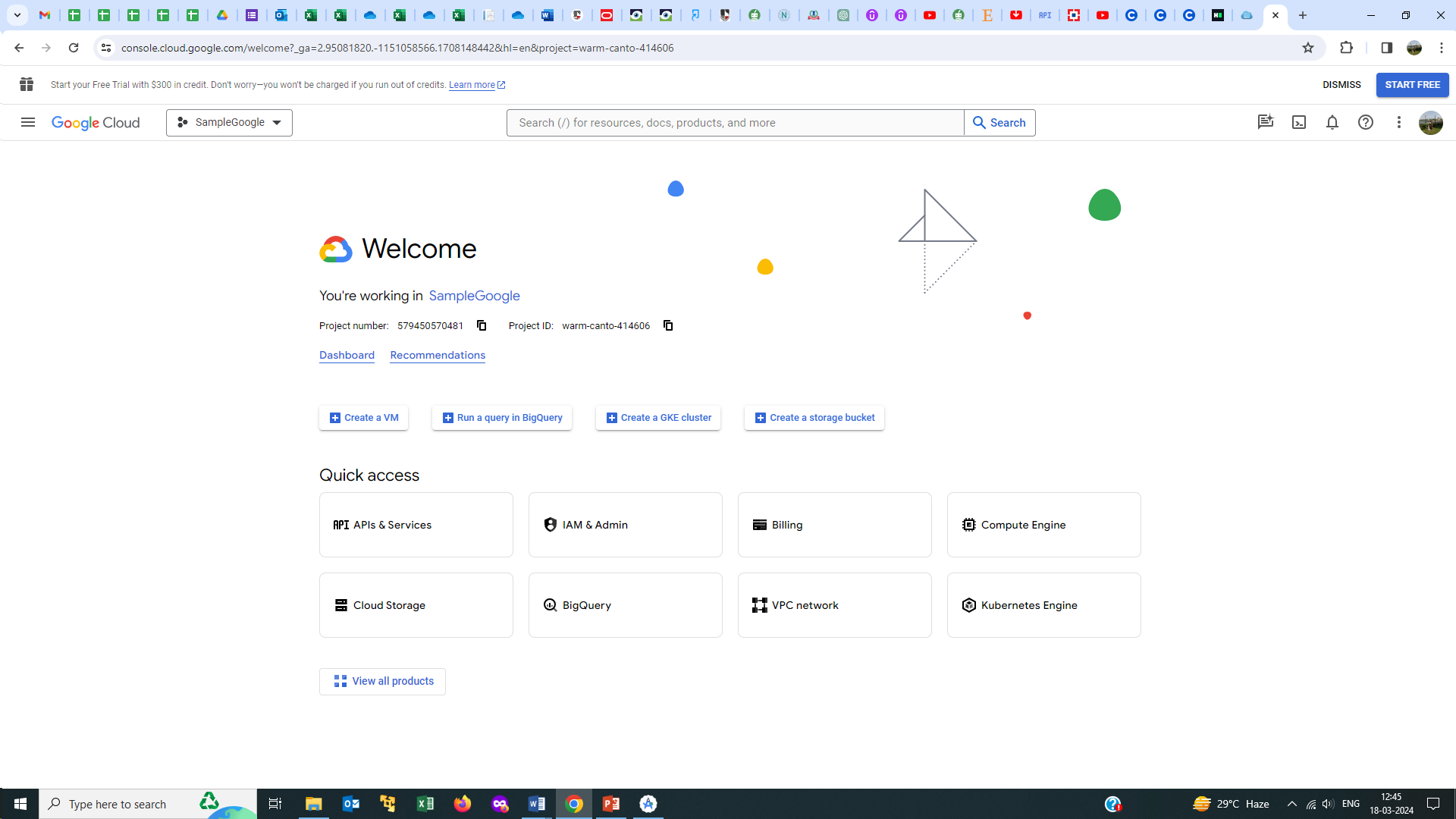


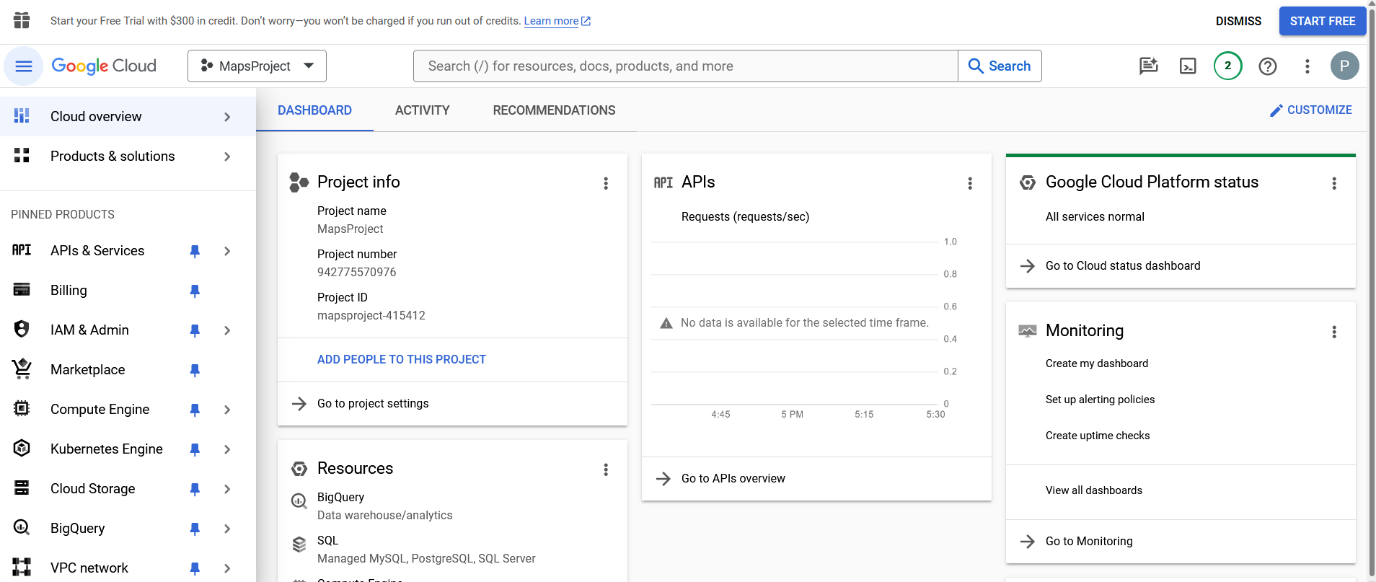


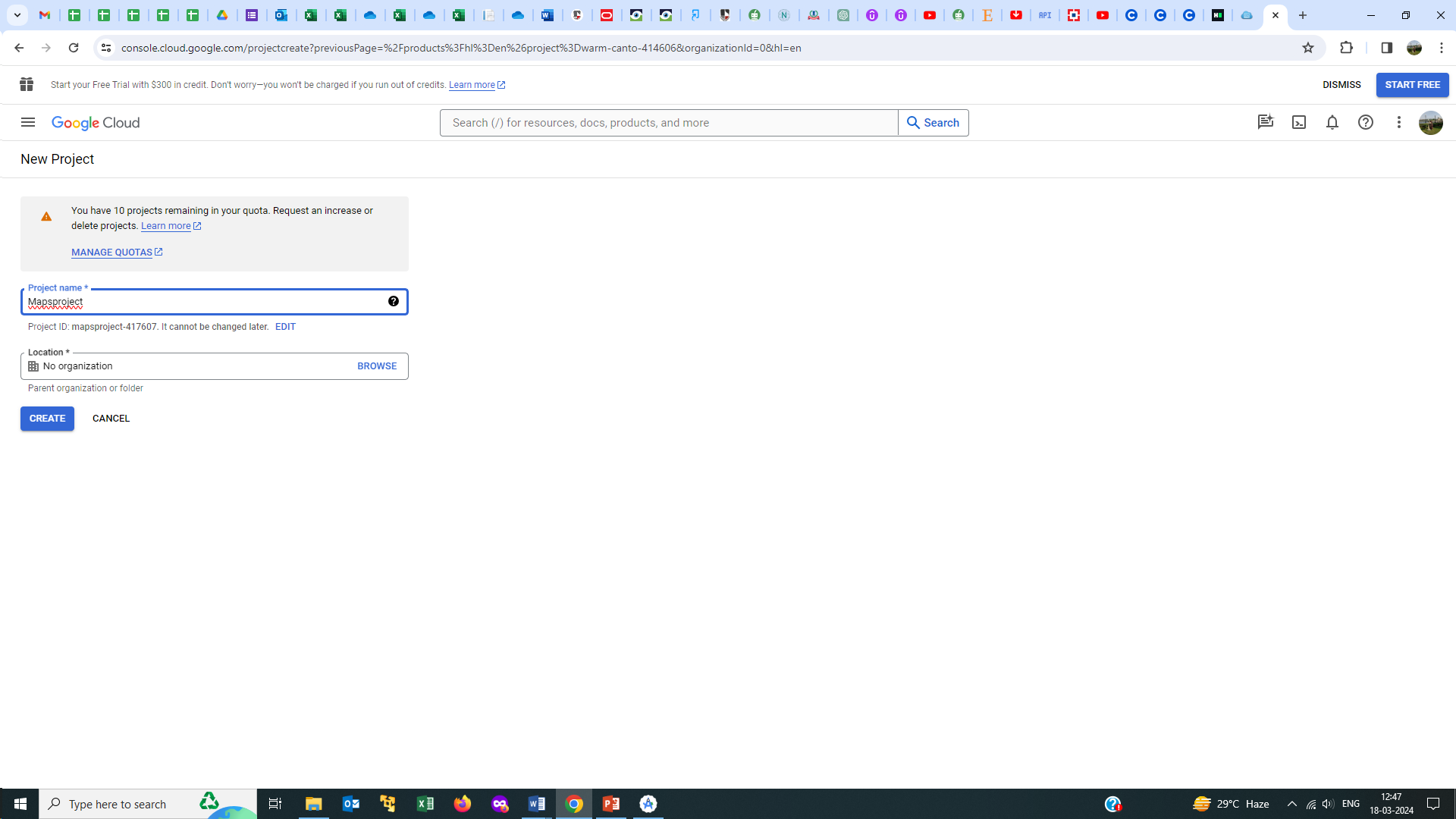
1. Create a project MapsProject on Google Cloud Platform



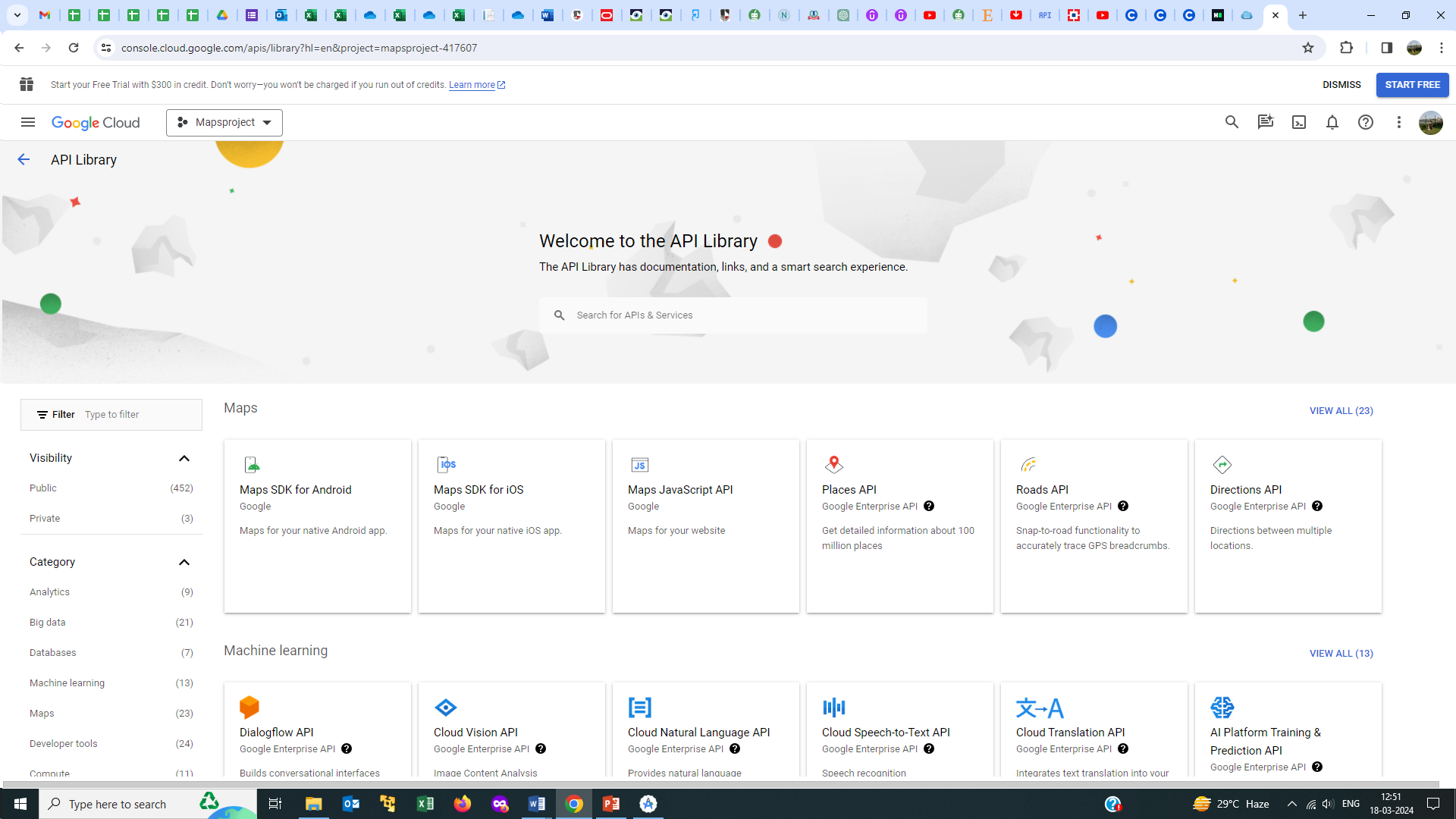
Go to Console

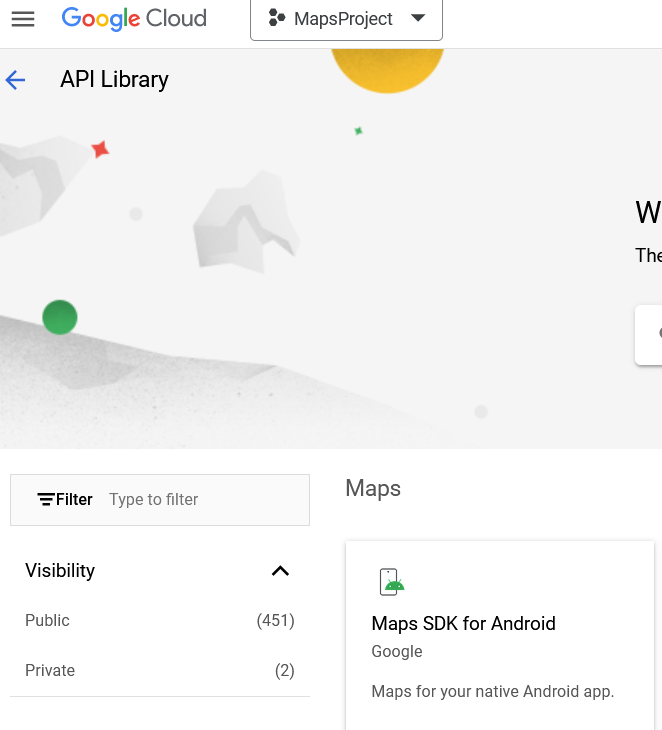


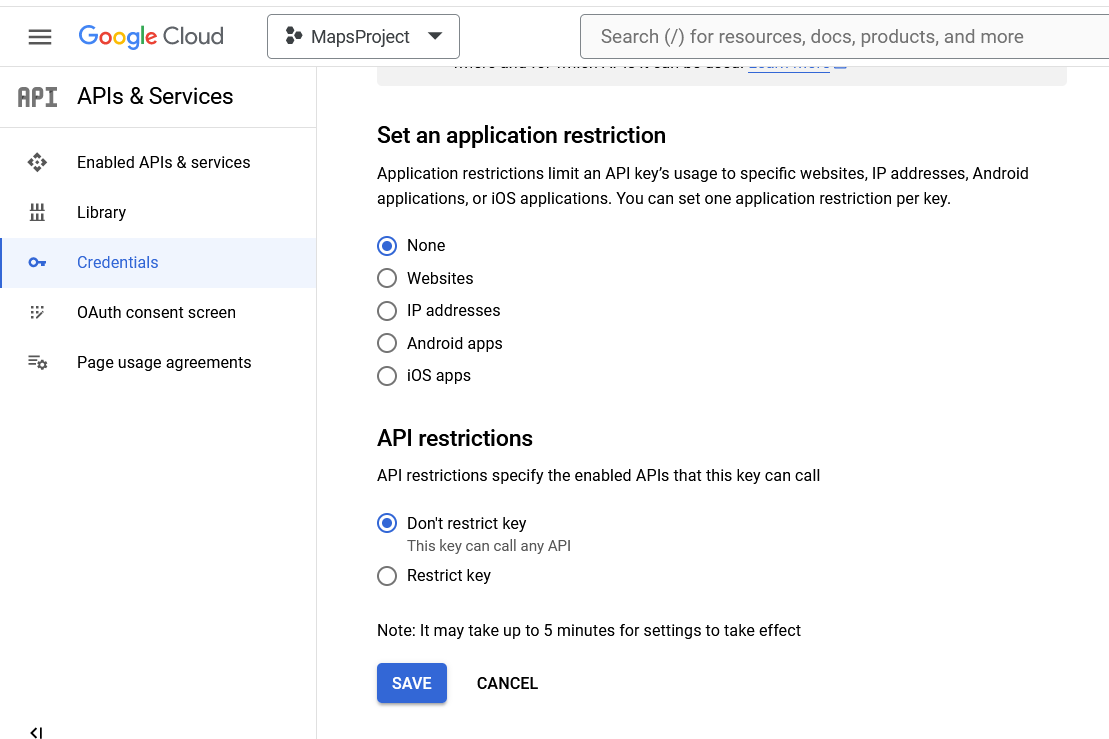


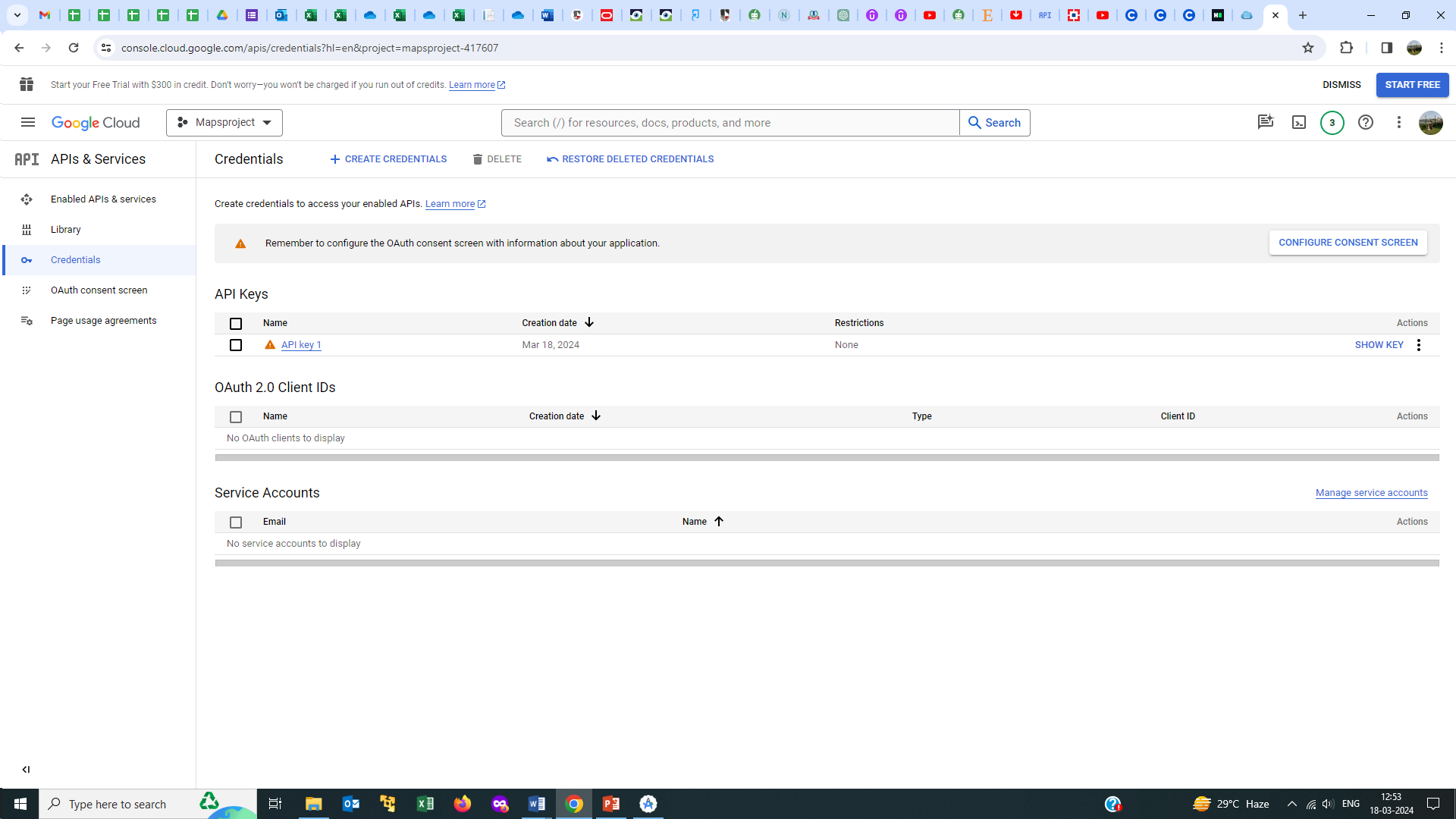


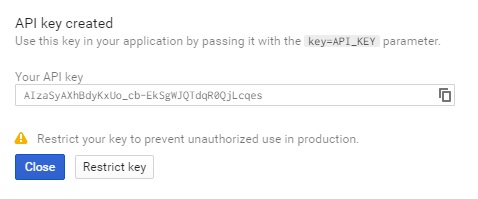
1. Select APIs and Services => Library => Maps SDK for Android and enable it.



1. 
2. Select APIs and Services => Credentials => Create Credentials







1. Copy the API Key from credentials and add it in AndroidManifest.xml.

|  |
| --- |
| Output of the app: |

**Part B (to be completed by students)**

**(Students must submit the soft copy as per the following segments. The soft copy must be uploaded on the Blackboard. The filename should be Batch\_RollNo\_Exp\_No)**

|  |  |
| --- | --- |
| **Roll No.: K005** | **Name: Jal Bafana** |
| **Prog/Yr/Sem: Cyber Sec/2nd/4th** | **Batch: K1** |
| **Date of Experiment: 12.03.2025** | **Date of Submission: 12.03.2025** |

1. **Program Scenario and Program code:** (Write Scenario and Paste your program code (Java, xml resource and layout)).

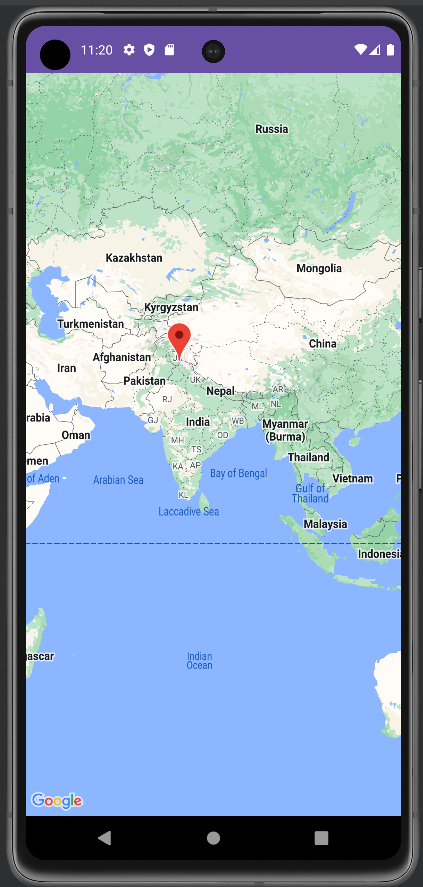
*package* com.example.k005\_lab7;  
  
*import* androidx.fragment.app.FragmentActivity;  
  
*import* android.os.Bundle;  
  
*import* com.google.android.gms.maps.CameraUpdateFactory;  
*import* com.google.android.gms.maps.GoogleMap;  
*import* com.google.android.gms.maps.OnMapReadyCallback;  
*import* 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.k005\_lab7.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 Sydney and move the camera* LatLng sydney = *new* LatLng(33, 76);  
 mMap.addMarker(*new* MarkerOptions().position(sydney).title("Marker in Sydney"));  
 mMap.moveCamera(CameraUpdateFactory.newLatLng(sydney));  
 }  
}

<?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" />

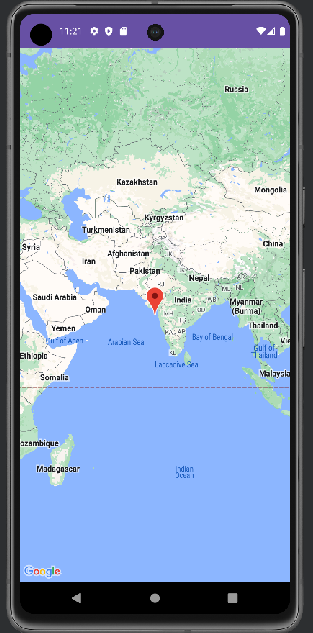
<?xml version="1.0" encoding="utf-8"?>  
<manifest xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:tools="http://schemas.android.com/tools">  
  
 <application  
 android:allowBackup="true"  
 android:dataExtractionRules="@xml/data\_extraction\_rules"  
 android:fullBackupContent="@xml/backup\_rules"  
 android:icon="@mipmap/ic\_launcher"  
 android:label="@string/app\_name"  
 android:roundIcon="@mipmap/ic\_launcher\_round"  
 android:supportsRtl="true"  
 android:theme="@style/Theme.K005\_Lab7"  
 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}".  
 -->* <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>  
  
 <meta-data  
 android:name="com.google.android.geo.API\_KEY"  
 android:value="Api was here(removed for safety reasons" />  
 </application>  
  
</manifest>

1. **Output:** (Paste your program input and output screen shots).

**Kashmir**

****

**Mumbai**

****

1. **Observations:** A brief description of the design aspects and working of the code in your own words.

In this experiment, we designed an Android app to implement Google Map services. The main objective was to use static latitude and longitude values to set markers on the map.

**Google Maps Integration**: The integration of Google Maps in the app was successful. The map loaded correctly and displayed the location based on the latitude and longitude.

1. **Questions:** Draw & Explain with respect to layouts the Scene Graph of the experiment.

**ActivityMapsBinding (Root Node)**

**|**

**|--- SupportMapFragment (Google Map Fragment)**

**|**

**|--- Other UI Elements (if any, such as Buttons, TextView, etc.)**

**Scene Graph Explanation:**

* The root of the graph is the **ActivityMapsBinding**. This is where all UI elements for this activity are defined and bound.
* Inside the ActivityMapsBinding, there’s a **SupportMapFragment** that will be responsible for displaying the Google Map. This is the most significant part of the layout since it occupies the entire screen for displaying the map.
* The scene graph may contain other elements if additional UI components are added, like buttons, text fields, or custom controls, but in this case, we have just the map fragment.

1. **Conclusion (Learning Outcomes):** How were the outcomes defined for the experiment in Part A fulfilled through the scenarios?

In this experiment, we successfully integrated Google Maps into an Android app, using static latitude and longitude to add markers on the map. We learned how to configure API keys, handle permissions, and display the map in full-screen mode.