7.Design an android app to fetch the JSON data from the internet and display the data using listView.
c. Employee data is stored in the internet. (use Async Task)
d. When app sends the request to the server, the server should provide data in json format.
e. The client app should fetch this data and display using listview.
https://raw.githubusercontent.com/wellingtoncosta/fake-contacts-api/master/db.json


```
{
 "contacts": [
  {
   "id": "5b2eee0a8fdd5b71c8148490",
   "age": 29,
   "name": "Campos York",
   "gender": "male",
   "company": "AVENETRO",
   "email": "camposyork@avenetro.com",
   "photo": "https://randomuser.me/api/portraits/men/3.jpg"
  },
```


activity_main.xml

```
<?xml version="1.0" encoding="utf-8"?>
c
<Button
android:layout_width="match_parent"
android:layout_height="wrap_content"
android:text="Fetch Data"
android:id="@+id/fetch"
/>
<TextView
android:layout_width="match_parent"
android:layout_height="wrap_content"
android:id="@+id/textView"
/>
<ListView
```

```
android:layout_width="match_parent"

android:layout_height="wrap_content"

android:id="@+id/list"

></ListView>

</LinearLayout>
```

Main_Activity.java

```
public class MainActivity extends AppCompatActivity {
 Button b;
 ListView lv;
 ArrayList<HashMap<String, String>>contactList;
@Override
protected void onCreate(Bundle savedInstanceState) {
super.onCreate(savedInstanceState);
 setContentView(R.layout.activity_main);
contactList = new ArrayList<>();
lv= (ListView) findViewById(R.id.list);
b= (Button) findViewById(R.id.fetch);
b.setOnClickListener(new View.OnClickListener() {
@Override
public void onClick(View v) {
String strUrl = "https://raw.githubusercontent.com/wellingtoncosta/fake-contacts-
api/master/db.json";

new UrlHandler().execute(strUrl);
 }
 });
 }
public class UrlHandler extends AsyncTask<String, Integer, String> {
@Override
protected void onPostExecute(String s) {
super.onPostExecute(s);
 ListAdapter adapter = new SimpleAdapter(MainActivity.this, contactList,
 R.layout.list_item, new String[]{ "id","name","email"},
new int[]{R.id.cid,R.id.cname, R.id.cemail});
lv.setAdapter(adapter);
 }
@Override
protected String doInBackground(String... params) {
 String json_response = null;
try {
 URL url = new URL(params[0]);
 HttpURLConnection connection = (HttpURLConnection) url.openConnection();
```

```
 connection.setRequestMethod("GET");
 connection.connect();
 InputStream in = new BufferedInputStream(connection.getInputStream());
 json_response = convertStreamToString(in);
if (json_response != null) {
try {
 JSONObject jsonObj = new JSONObject(json_response);
// Getting JSON Array node
JSONArray contacts = jsonObj.getJSONArray("contacts");
// looping through All Contacts
for (int i = 0; i < contacts.length(); i++) {
 JSONObject c = contacts.getJSONObject(i);
 String id = c.getString("id");
 String name = c.getString("name");
 String email = c.getString("email");
// tmp hash map for single contact
HashMap<String, String> contact = new HashMap<>();
// adding each child node to HashMap key => value
contact.put("id", id);
 contact.put("name", name);
 contact.put("email", email);
// adding contact to contact list
contactList.add(contact);
 }
 } catch (JSONException e) {
 Log.e("error", "Json parsing error: " + e.getMessage());
 }
 } else {
 Log.e("error", "Couldn't get json from server.");
 }
 } catch (MalformedURLException e) {
 e.printStackTrace();
 } catch (IOException e) {
 e.printStackTrace();
 }
return null;
 }
private String convertStreamToString(InputStream is) {
 BufferedReader reader = new BufferedReader(new InputStreamReader(is));
 StringBuilder sb = new StringBuilder();
 String line;
try {
while ((line = reader.readLine()) != null) {
 sb.append(line).append('\n');
 }
 } catch (IOException e) {
```

```
 e.printStackTrace();
 }
return sb.toString();
 }
 }
}
```

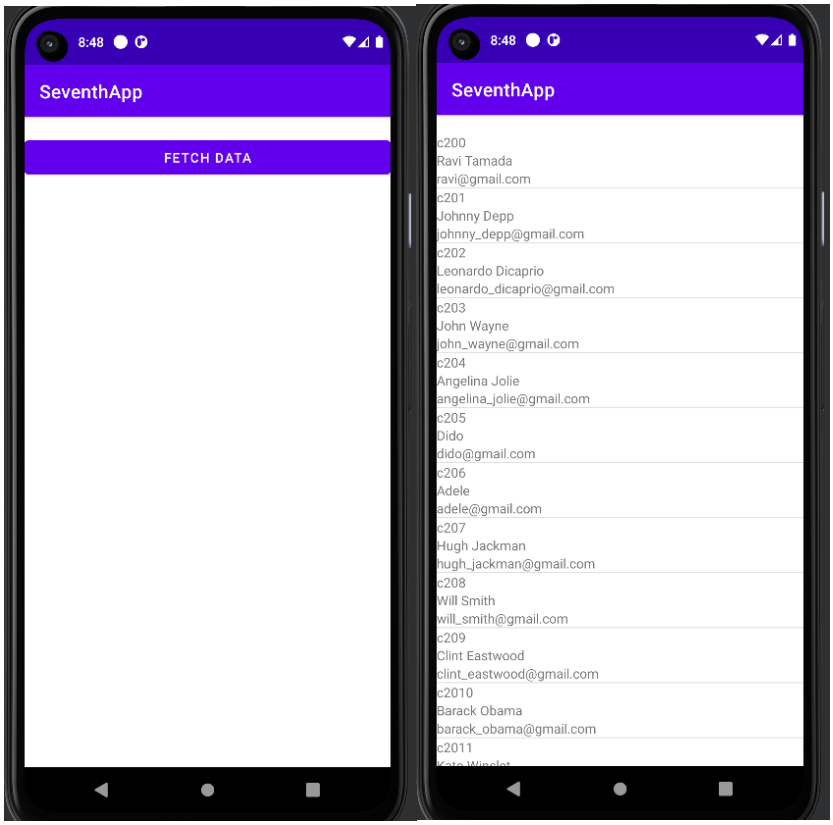
list_item.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
android:orientation="vertical" android:layout_width="match_parent"
android:layout_height="match_parent">
<TextView
android:layout_width="match_parent"
android:layout_height="wrap_content"
android:id="@+id/cid"
/>
<TextView
android:layout_width="match_parent"
android:layout_height="wrap_content"
android:id="@+id/cname"
/>
<TextView
android:layout_width="match_parent"
android:layout_height="wrap_content"
android:id="@+id/cemail"
/>
</LinearLayout>
```

Android_Mainfest.xml

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

Output:



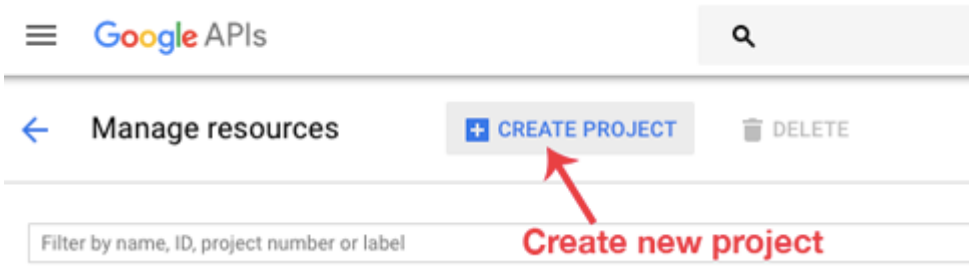
Q8.Develop an android app on Google Map, and should provide following functions.
a. How to incorporate Google Maps into an application.
b. How to register for and receive GPS location information.
c. How to create Google Maps Overlays.
d. Accept city name from user and marks it on map.
e. Explore features like Zoom and map types.

Steps For Getting The Google Maps Api Key:

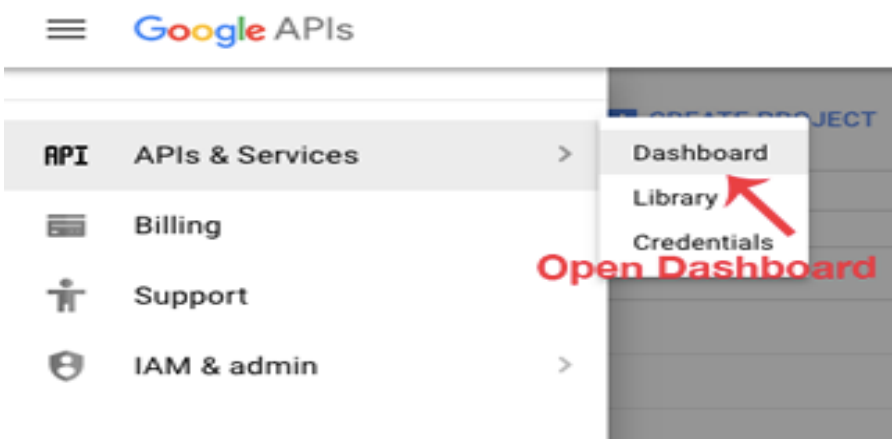
An API key is needed to access the Google Maps servers. This key is free and you can use it with any of your applications. If you haven't created project, you can follow the below steps to get started:

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

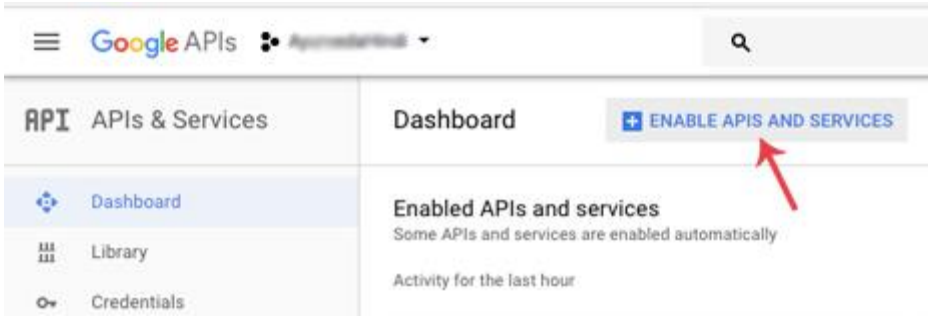
Step 2: Now create new project. You can create new project by clicking on the **Create Project** button and give name to your project.



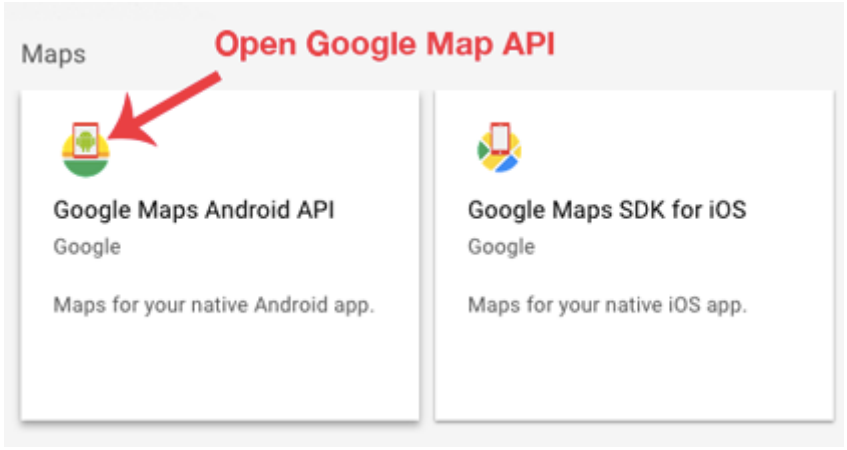
Step 3: Now click on APIs & Services and open **Dashboard** from it.



Step 4: In this open **Enable APIS AND SERICES**.

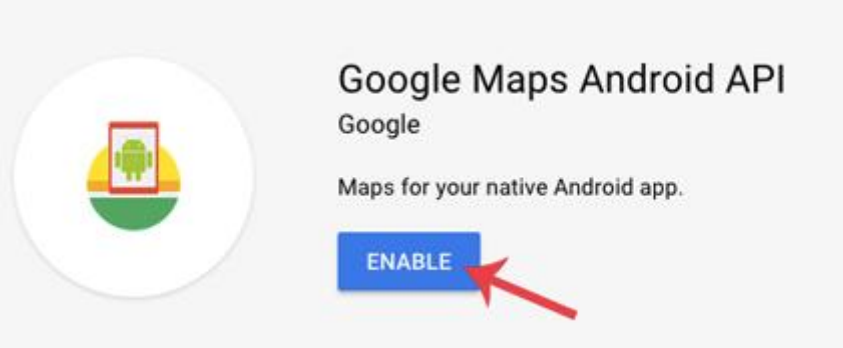


Step 5: Now open Google Map Android API.

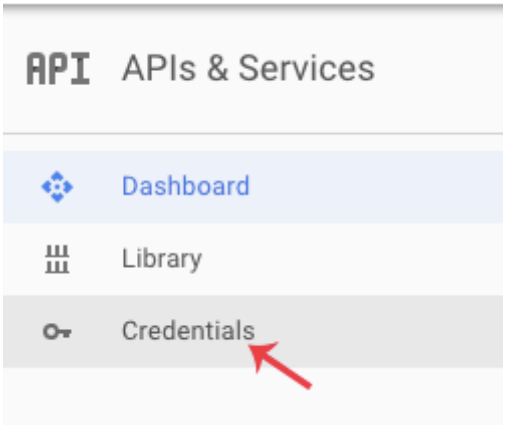


Step 6: Now enable the Google Maps Android API.

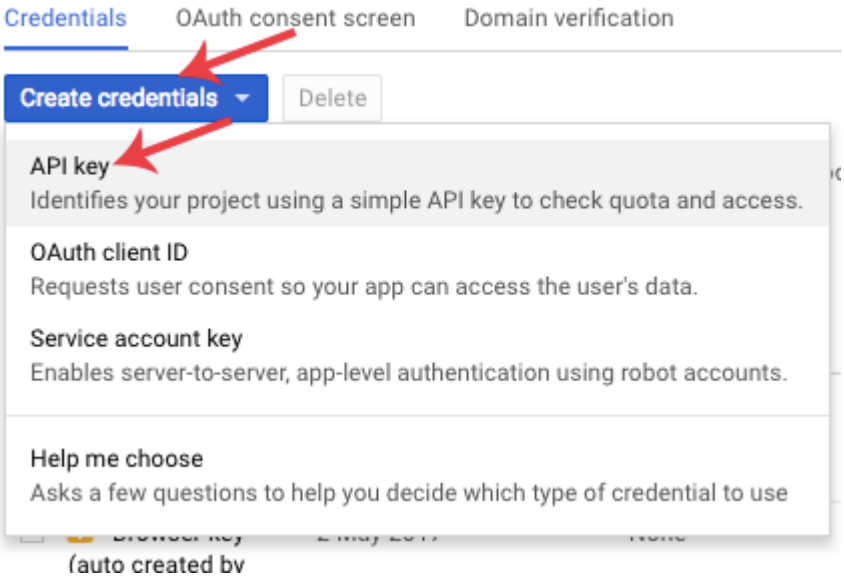
Step 6: Now enable the Google Maps Android API.



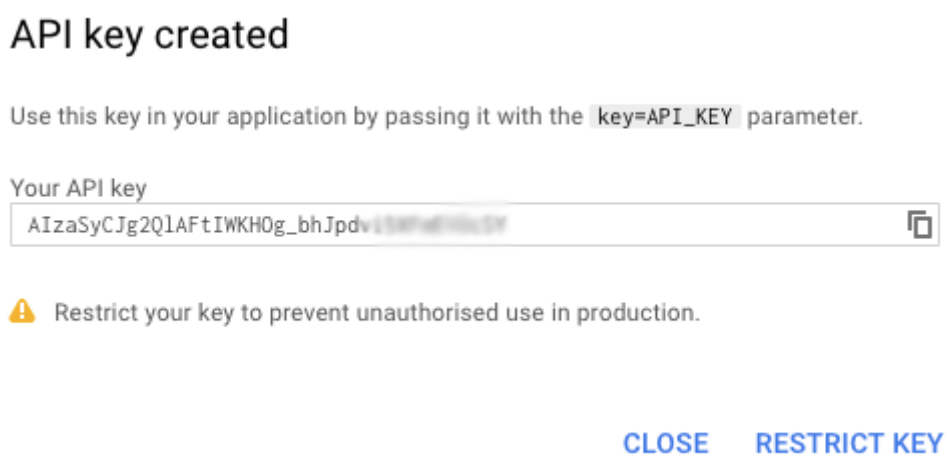
Step 6: Now go to **Credentials**



Step 7: Here click on Create credentials and choose API key



Step 8: Now API 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.



Google Maps Example To Access User Current Location In Android Studio:

In the below Google Map example we will show user current location in Map. We also example different map types, methods and lots more details required while implementing Map in Android.

Step 1: Create a New Android Project select Google Maps Activity and then click Next

MapsActivity.java

```
package com.example.program8;

import androidx.fragment.app.FragmentActivity;

import android.location.Address;
import android.location.Geocoder;
import android.os.Bundle;
import android.view.View;
import android.widget.EditText;
import android.widget.TextView;

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.program8.databinding.ActivityMapsBinding;

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

public class MapsActivity extends FragmentActivity implements OnMapReadyCallback {
```

```
    private GoogleMap mMap;
    private TextView tv;
    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 Bengaluru and move the camera
        LatLng Bengaluru = new LatLng(13, 78);
        mMap.addMarker(new MarkerOptions().position(Bengaluru).title("Marker in
Bengaluru"));
        mMap.moveCamera(CameraUpdateFactory.newLatLng(Bengaluru));
    }
    public void setmMap(GoogleMap mMap) {
        this.mMap = mMap;
    }

    public void onSearch(View view) {
        List<Address> addressList = null;
        EditText et_location = (EditText) findViewById(R.id.et1);
        String location = et_location.getText().toString();
```

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

            try {


                addressList = geocoder.getFromLocationName(location, 1);
            } catch (IOException e) {
                e.printStackTrace();
            }


            Address address = addressList.get(0);
            LatLng latLng = new LatLng(address.getLatitude(), address.getLongitude());
            mMap.addMarker(new MarkerOptions().position(latLng).title(location));
            mMap.animateCamera(CameraUpdateFactory.newLatLng(latLng));
        }
    }

    public void onType(View view) {
        if (mMap.getMapType() == GoogleMap.MAP_TYPE_NORMAL) {
            mMap.setMapType(GoogleMap.MAP_TYPE_SATELLITE);
        } else {
            mMap.setMapType(GoogleMap.MAP_TYPE_NORMAL);
        }
    }

    public void onZoom(View view) {
        if (view.getId() == R.id.zoomin) {
            mMap.animateCamera(CameraUpdateFactory.zoomIn());
        }
        if (view.getId() == R.id.zoomout) {
            mMap.animateCamera(CameraUpdateFactory.zoomOut());
        }

    }
}
```

Activity_maps.xml

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
android:layout_width="match_parent"
    android:orientation="vertical" android:layout_height="400dp">
    <LinearLayout
        android:layout_width="wrap_content"
```

```
    android:layout_height="wrap_content"
    android:orientation="horizontal"
    android:id="@+id/ll1">

    <EditText
        android:id="@+id/et1"
        android:layout_width="196dp"
        android:layout_height="wrap_content" />

    <Button
        android:id="@+id/searchbut"
        android:layout_width="98dp"
        android:layout_height="wrap_content"
        android:onClick="onSearch"
        android:text="Search" />

    <Button
        android:id="@+id/typebut"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:onClick="onType"
        android:text="Type" />

</LinearLayout>
<LinearLayout
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:orientation="horizontal"
    android:layout_below="@id/ll1"
    android:id="@+id/linearLayout"
    android:layout_alignParentBottom="true">

    <fragment xmlns:android="http://schemas.android.com/apk/res/android"
        xmlns:tools="http://schemas.android.com/tools"
        android:id="@+id/map"
        android:layout_below="@id/ll1"
        android:name="com.google.android.gms.maps.SupportMapFragment"
        android:layout_width="343dp"
        android:layout_height="match_parent"
        tools:context="com.example.mohan.demomaps.MapsActivity" />

    <LinearLayout
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:orientation="vertical">
```

```
            <Button
                android:id="@+id/zoomin"
                android:layout_width="wrap_content"
                android:layout_height="69dp"
                android:onClick="onZoom"
                android:text="+" />

            <Button
                android:id="@+id/zoomout"
                android:layout_width="wrap_content"
                android:layout_height="68dp"
                android:onClick="onZoom"
                android:text="-" />
        </LinearLayout>
    </LinearLayout>
</RelativeLayout>
```

AndroidMainfest.xml

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.example.program8">

    <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"/>
    <uses-permission android:name="android.permission.ACCESS_NETWORK_STATE"/>
    <uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE" />
    <uses-permission
android:name="com.google.android.providers.gsf.permission.READ_GSERVICES" />


    <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.Program8">



        <!--
            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}".
-->

```
<meta-data
    android:name="com.google.android.geo.API_KEY"
    android:value="AIzaSyAPTYkumdx_qL8KCypbGI5asIsNRXmt2p8" />

<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>
</application>

</manifest>
```

Output: