

## Department of Computer Science and Engineering

<b>Department:</b> Computer Science and Engineering	<b>Course Type:</b> Programme Core
<b>Course Title:</b> Mobile Application Development	<b>Course Code:</b> 18CSL67
<b>L-T-P:</b> 0-0-4	<b>Credits:</b> 2
<b>Total Contact Hours:</b> 36 Hours	<b>Duration of SEE:</b> 3 Hours
<b>SEE Marks:</b> 50	<b>CIE Marks:</b> 50

### COURSE DESCRIPTION

This laboratory introduces students with Android programming concepts and provides knowledge to create android apps.

### PREREQUISITES

- Students should have knowledge on any programming language.

### COURSE OBJECTIVES

- To introduce Android platform and its architecture.
- To learn activity creation and Android UI designing.
- To be familiarized with Intent, Broadcast receivers and Internet services.
- To work with SQLite Database and content providers.
- To integrate multimedia, camera and Location based services in Android Application. To explore Mobile security issues.

### LAB EXERCISES

#### PART – A

- In this lab we will be learning how to use and extend the Android user interface library.
  - Views, View Groups, Layouts, and Widgets are and how they relate to each other.
  - How to declare and reference resources in code.
  - How to navigate between multiple activities.
  - How to share the data between the activities.
  - Explore life-cycle methods of an activity.
  - How to use Events and Event Listeners.
  - How to create Toast Notifications.
- You will expand on your knowledge of the Android user interface library.
  - How to declare layouts statically as an xml resource.
  - How to create custom Views from scratch to suit a specific need.
  - How to create Options and Context Menus.

## Department of Computer Science and Engineering

- d. How to use List Adapter and Array Adapter to bind data source to a List View.
- e. How to create Alert Dialog and progress Dialog in your activity.
3. You will be persisting data using a SQLite Database and preserving the state of an application during its lifecycle.
  - a. How to save & restore data as Application Preferences (Shared Preference).
  - b. How to save & restore data as Instance State.
  - c. How to create and manage a SQLite Database in Android.
  - d. How to insert, update, remove, and retrieve data from a SQLite Database.
  - e. Display data using Recycler View.
4. Develop an app to capture a photo and store it into SDCard, extend this app to display all the photos capture in the grid view.
  - a. How to use the Camera.
  - b. How to write data to the SD card.
5. Create an application to demonstrate few key features of the Android framework. In particular, the application demonstrates how to send SMS text messages.
  - a. How to send SMS text messages.
  - b. How to dial using an in-built dialer
  - c. How to send email.
6. Develop an app that include broadcast Receiver to receive the miss calls from the Known number and display it to the user using notification services. This same app should also fetch phone number from the inbuilt contacts using the concept of content provider.
  - a. How to use broadcast receiver and notifications.
  - b. How to use content providers.
7. Design an android app to fetch the JSON data from the internet and display the data using listView.
  - a. Employee data is stored in the internet. (use Async Task)
  - b. When app sends the request to the server, the server should provide data in json format.
  - c. The client app should fetch this data and display using listview.
8. 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.

### PART – B

Student should develop a mini project in a group of three.

## Department of Computer Science and Engineering

### ASSESSMENT METHODS

Parameters	Marks
Midterm Test (Avg. of 2 Tests)	30
Course Project	20
<b>Total</b>	<b>50</b>
Final Exam will be conducted for 100 marks (SEE)	

### COURSE OUTCOMES

COs	Description	Bloom's level
CO 1	Describe Android platform, Architecture and features	L1
CO 2	Design User Interface and develop activity for Android App	L4
CO 3	Use Intent, Broadcast receivers and Internet services in Android App.	L3
CO 4	Design and implement Database Application and Content providers.	L4
CO 5	Use multimedia, camera and Location based services in Android App.	L3

### Mapping of Course outcomes (COs) to Program outcomes (POs\*)& PSO \*\*

Course Outcomes mapping to Program Outcomes													PSOs		
POs COs	1	2	3	4	5	6	7	8	9	10	11	12	PSO1	PSO2	PSO3
CO1	3	1	2		2								2	2	
CO2	3	3	2		2								2	2	
CO3	3	2	2		2								2	2	
CO4	3	2	2		2								2	2	
CO5	3	2	2		2								2	2	

3: Strong, 2: Medium, 1: Weak \*\* H: Highly related S: Supportive

# Mobile Application Development

Course Code:18CSL67

## Part – A

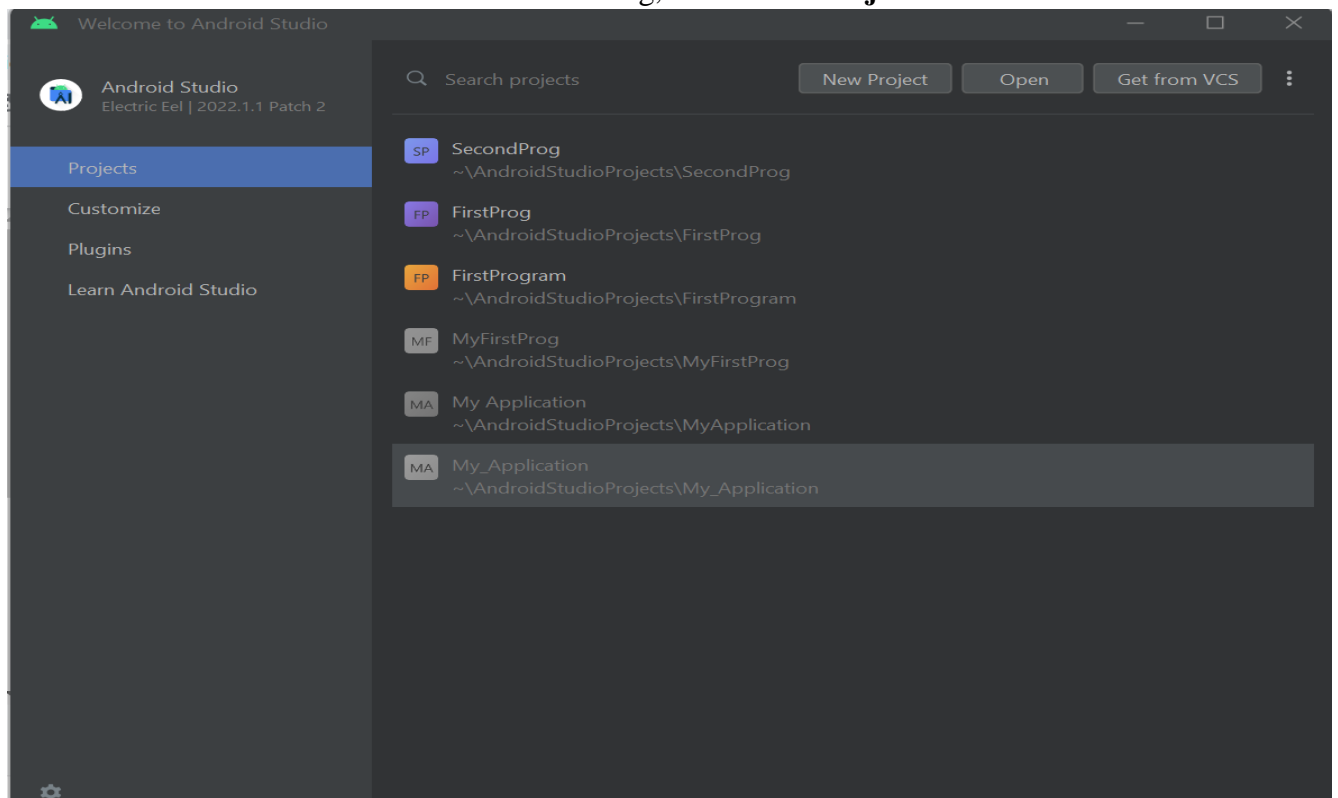
Q1. In this lab we will be learning how to use and extend the Android user interface library.

- Views, View Groups, Layouts, and Widgets are and how they relate to each other.
- How to declare and reference resources in code.
- How to navigate between multiple activities.
- How to share the data between the activities.
- Explore life-cycle methods of an activity.
- How to use Events and Event Listeners.
- How to create Toast Notifications.

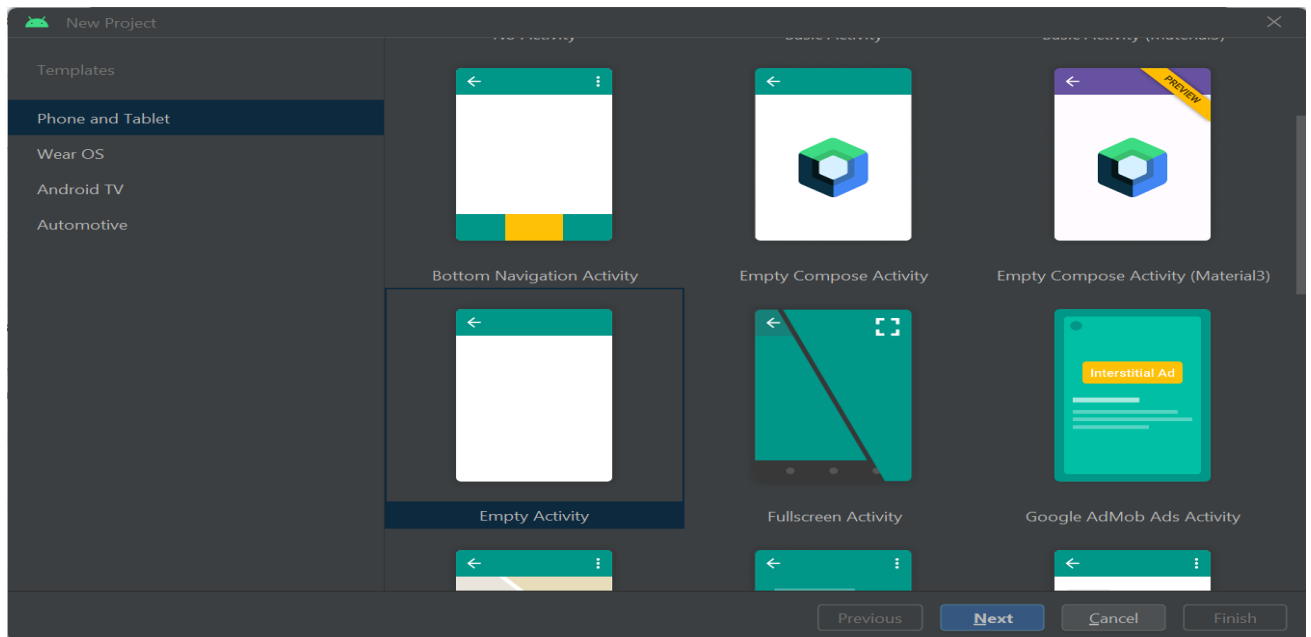
### Create your first project

#### **Step 1: Create a new project**

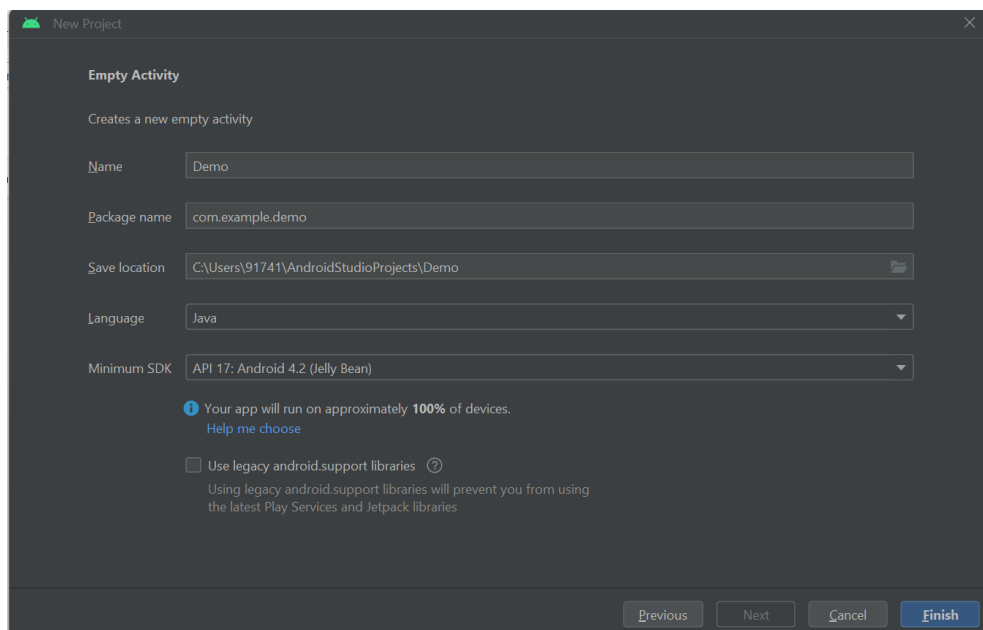
- Open Android Studio.
- In the **Welcome to Android Studio** dialog, click **New Project**.



3. Select **Empty Activity** (not the default). Click **Next**.



4. Give your application a name such as **Demo**



5. Make sure the **Language** is set to **Java** .

6. Leave the defaults for the other fields.

7. Click **Finish**.

**After these steps, Android Studio:**

- Creates a folder for your Android Studio project called **MyFirstApp**. This is usually in a folder called **AndroidStudioProjects** below your home directory.

- Builds your project (this may take a few moments). Android Studio uses Gradle as its build system. You can follow the build progress at the bottom of the Android Studio window.
- Opens the code editor showing your project.

### **Code:**

#### **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="com.example.firstprog.MainActivity">

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Hello World!"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintLeft_toLeftOf="parent"
        app:layout_constraintRight_toRightOf="parent"
        app:layout_constraintTop_toTopOf="parent" />
</androidx.constraintlayout.widget.ConstraintLayout>
```

#### **MainActivity.java**

```
package com.example.firstprog;

import androidx.appcompat.app.AppCompatActivity;

import android.os.Bundle;
import androidx.appcompat.app.AppCompatActivity;
import android.os.Bundle;
import android.util.Log;

public class MainActivity extends AppCompatActivity {

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

```

        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        Log.d("lifecycle", "onCreate invoked");
    }
    @Override
    protected void onStart() {
        super.onStart();
        Log.d("lifecycle", "onStart invoked");
    }
    @Override
    protected void onResume() {
        super.onResume();
        Log.d("lifecycle", "onResume invoked");
    }
    @Override
    protected void onPause() {
        super.onPause();
        Log.d("lifecycle", "onPause invoked");
    }
    @Override
    protected void onStop() {
        super.onStop();
        Log.d("lifecycle", "onStop invoked");
    }
    @Override
    protected void onRestart() {
        super.onRestart();
        Log.d("lifecycle", "onRestart invoked");
    }
    @Override
    protected void onDestroy() {
        super.onDestroy();
        Log.d("lifecycle", "onDestroy invoked");
    }
}

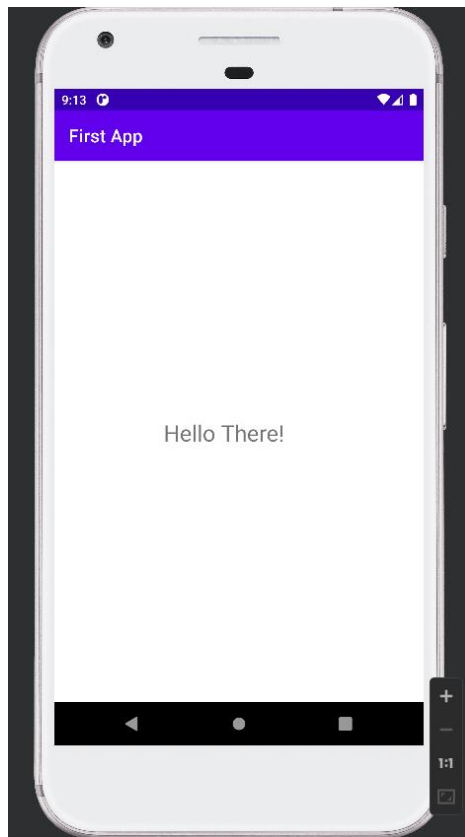
```

### **Output:**

```

2022-05-16 21:12:00.186 3095-3117/com.example.firstapp D/HostConnection: HostConnection::get() New Host Connect
2022-05-16 21:12:00.187 3095-3117/com.example.firstapp D/goldfish-address-space: allocate: Ask for block of siz
2022-05-16 21:12:00.206 3095-3117/com.example.firstapp D/goldfish-address-space: allocate: ioctl allocate retur
2022-05-16 21:12:00.256 3095-3117/com.example.firstapp D/HostConnection: HostComposition ext ANDROID_EMU_CHECKS
2022-05-16 21:12:01.063 3095-3111/com.example.firstapp W/System: A resource failed to call close.
2022-05-16 21:12:25.352 3095-3095/com.example.firstapp D/LifeCycle: onPause invoked
2022-05-16 21:12:25.876 3095-3095/com.example.firstapp D/LifeCycle: onStop invoked
2022-05-16 21:12:28.454 3095-3095/com.example.firstapp D/LifeCycle: onRestart invoked
2022-05-16 21:12:28.456 3095-3095/com.example.firstapp D/LifeCycle: onStart invoked
2022-05-16 21:12:28.457 3095-3095/com.example.firstapp D/LifeCycle: onResume invoked
2022-05-16 21:15:59.855 3095-3095/com.example.firstapp D/LifeCycle: onPause invoked
2022-05-16 21:16:02.426 3095-3095/com.example.firstapp D/LifeCycle: onStop invoked

```



## **Enable USB debugging**

To let Android Studio communicate with your Android device, you must enable USB debugging in the Developer options settings of the device.

To show developer options and enable USB debugging:

1. On your Android device, tap **Settings > About phone**.
2. Tap **Build number** seven times.
3. If prompted, enter your device password or pin. You know you succeeded when you see a **You are now a developer!** message.
4. Return to Settings and then tap **System > Developer options**.
5. If you don't see **Developer options**, tap **Advanced options**.


## **Run your app on the Android device with a cable**

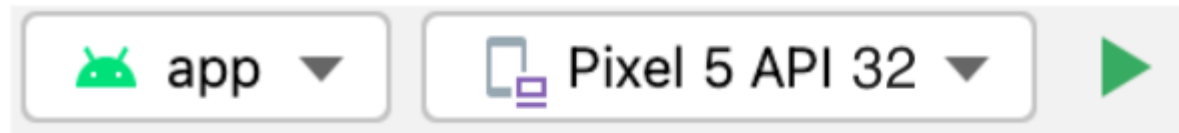
There are two ways to connect your device to Android Studio, through a cable or through Wi-Fi. You can choose whichever you like more.

To run your app from Android Studio on your Android device:

1. Connect your Android device to your computer with a USB cable. A dialog should appear on your device, which asks you to allow USB debugging.



2. Select the **Always allow from this computer** checkbox and then tap **OK**.
3. In Android Studio on your computer, make sure your device is selected in the dropdown. Click .



4. Select your device and then click **OK**. Android Studio installs the app on your device and runs it.
5. If your device runs an Android platform that isn't installed in Android Studio and you see a message that asks whether you want to install the needed platform, click **Install** > **Continue** > **Finish**. Android Studio installs the app on your device and runs it.

Q2. You will expand on your knowledge of the Android user interface library.

- a. How to declare layouts statically as an xml resource.
- b. How to create custom Views from scratch to suit a specific need.
- c. How to create Options and Context Menus.
- d. How to use ListAdapter and ArrayAdapter to bind data source to a List View.
- e. How to create AlertDialog and progress Dialog in your activity.

### **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"
tools:visibility="visible">

<Button
    android:id="@+id/button"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Long Click on Me"
    android:textSize="25sp"
    android:visibility="visible"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintLeft_toLeftOf="parent"
```

```
app:layout_constraintRight_toRightOf="parent"
app:layout_constraintTop_toTopOf="parent" />
```

```
</androidx.constraintlayout.widget.ConstraintLayout>
```

### **MainActivity.java**

```
package com.example.secondprog;

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

import android.os.Bundle;
import android.view.ContextMenu;
import android.view.MenuItem;
import android.view.View;
import android.widget.Button;
import android.widget.Toast;

public class MainActivity extends AppCompatActivity {
    private Button button;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        button = findViewById(R.id.button);
        registerForContextMenu(button);
    }

    @Override
    public void onCreateContextMenu(ContextMenu menu, View v,
ContextMenu.ContextMenuInfo menuInfo) {
        getMenuInflater().inflate(R.menu.menu, menu);
        super.onCreateContextMenu(menu, v, menuInfo);
    }

    @Override
    public boolean onContextItemSelected(@NonNull MenuItem item) {
        Toast.makeText(this, ""+item.getTitle(), Toast.LENGTH_SHORT).show();
        return super.onContextItemSelected(item);
    }
}
```

**Note:** on the *menu* right click and create a menu.xml

### menu.xml

```
<?xml version="1.0" encoding="utf-8"?>
<menu xmlns:android="http://schemas.android.com/apk/res/android">
    <item
        android:title="@string/one"
    />
    <item
        android:title="@string/two"
    />
    <item
        android:title="@string/three"
    />
    <item
        android:title="@string/four"
    />
    <item
        android:title="@string/five"
    />
</menu>
```

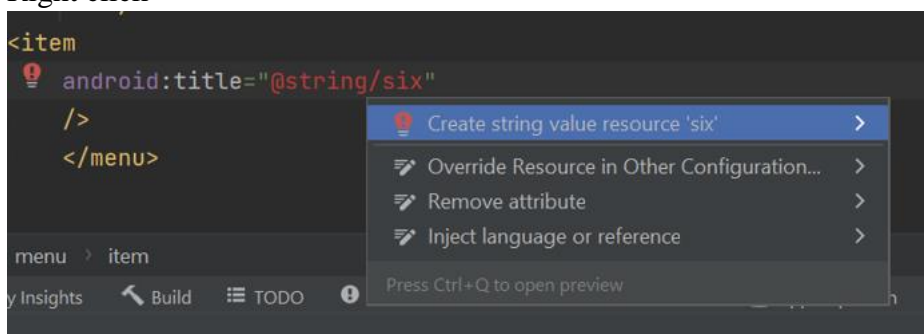
**Note:**

### Output:

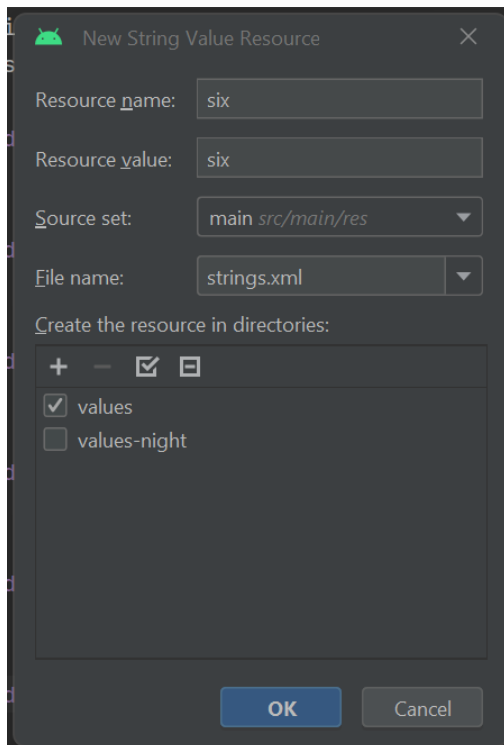


```
<item
    android:title="@string/six"
/>
</menu>
```

Right click

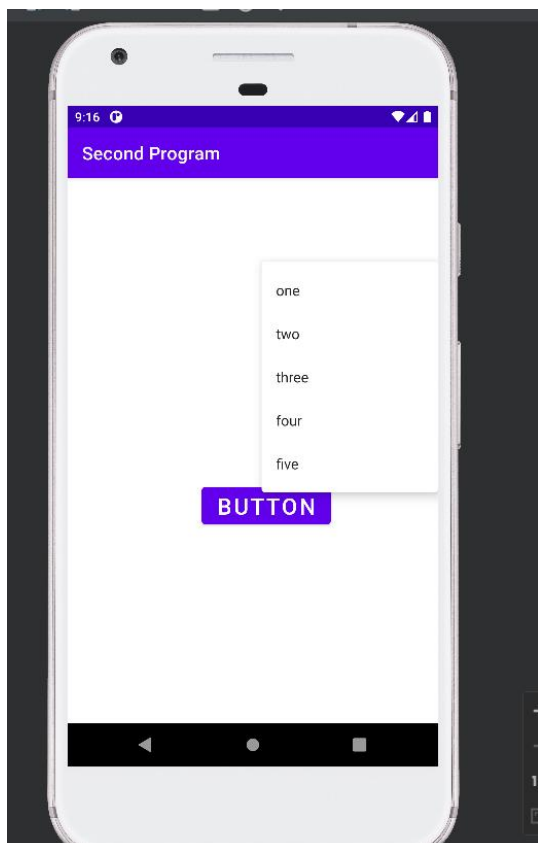


Click on create string value resource



Enter the Resource value and press **ok**

## Output



Q3. You will be persisting data using an SQLite Database and preserving the state of an application during its lifecycle.

- a. How to save & restore data as Application Preferences (Shared Preference).
- b. How to save & restore data as Instance State.
- c. How to create and manage an SQLiteDatabase in Android.
- d. How to insert, update, remove, and retrieve data from an SQLite Database.
- e. Display data using RecyclerView.

### **Activity\_main.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"
    android:padding="10dp"
    tools:context=".MainActivity">
    <TextView
        android:id="@+id/texttitle"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="Please enter the details below"
        android:textSize="24dp"
        android:layout_marginTop="20dp"/>

    <EditText
        android:id="@+id/name"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_below="@+id/texttitle"
        android:hint="Name"
        android:inputType="textPersonName"
        android:textSize="24dp" />

    <EditText
        android:id="@+id/contact"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_below="@+id/name"
        android:hint="Contact"
        android:inputType="number"
        android:textSize="24dp" />

    <EditText
        android:id="@+id/dob"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_below="@+id/contact"
        android:hint="DOB"
        android:inputType="text"
        android:textSize="24dp" />
</RelativeLayout>
```

```
android:layout_width="match_parent"
android:layout_height="wrap_content"
android:layout_below="@+id/contact"
android:hint="Date of Birth"
android:inputType="number"
android:textSize="24dp" />
```

```
<Button
```

```
    android:id="@+id/btnInsert"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_below="@id/dob"
    android:layout_marginTop="30dp"
    android:text="Insert New Data"
    android:textSize="24dp" />
```

```
<Button
```

```
    android:id="@+id/btnUpdate"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_below="@id/btnInsert"
    android:text="Update Data"
    android:textSize="24dp" />
```

```
<Button
```

```
    android:id="@+id/btnDelete"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_below="@id/btnUpdate"
    android:text="Delete Existing Data"
    android:textSize="24dp" />
```

```
<Button
```

```
    android:id="@+id/btnView"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_below="@id/btnDelete"
    android:text="View Data"
    android:textSize="24dp" />
```

```
</RelativeLayout>
```

## **MainActivity.java**

```
package com.example.fourprog;

import androidx.appcompat.app.AlertDialog;
import androidx.appcompat.app.AppCompatActivity;

import android.database.Cursor;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.Toast;

public class MainActivity extends AppCompatActivity {
    EditText name, contact, dob;
    Button insert, update, delete, view;
    DBHelper DB;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        name = findViewById(R.id.name);
        contact = findViewById(R.id.contact);
        dob = findViewById(R.id.dob);
        insert = findViewById(R.id.btnInsert);
        update = findViewById(R.id.btnUpdate);
        delete = findViewById(R.id.btnDelete);
        view = findViewById(R.id.btnView);
        DB = new DBHelper(this);
        insert.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                String nameTXT = name.getText().toString();
                String contactTXT = contact.getText().toString();
                String dobTXT = dob.getText().toString();

                Boolean checkinsertdata = DB.insertuserdata(nameTXT, contactTXT, dobTXT);
                if(checkinsertdata==true)
                    Toast.makeText(MainActivity.this, "New Entry Inserted",
Toast.LENGTH_SHORT).show();
                else
                    Toast.makeText(MainActivity.this, "New Entry Not Inserted",
Toast.LENGTH_SHORT).show();
            }
        });
    }
}
```

```

update.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        String nameTXT = name.getText().toString();
        String contactTXT = contact.getText().toString();
        String dobTXT = dob.getText().toString();

        Boolean checkupdatedata = DB.updateuserdata(nameTXT, contactTXT, dobTXT);
        if(checkupdatedata==true)
            Toast.makeText(MainActivity.this, "Entry Updated",
Toast.LENGTH_SHORT).show();
        else
            Toast.makeText(MainActivity.this, "New Entry Not Updated",
Toast.LENGTH_SHORT).show();
    }    });

delete.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        String nameTXT = name.getText().toString();
        Boolean checkdeletedata = DB.deletedata(nameTXT);
        if(checkdeletedata==true)
            Toast.makeText(MainActivity.this, "Entry Deleted",
Toast.LENGTH_SHORT).show();
        else
            Toast.makeText(MainActivity.this, "Entry Not Deleted",
Toast.LENGTH_SHORT).show();
    }    });

view.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        Cursor res = DB.getdata();
        if(res.getCount()==0){
            Toast.makeText(MainActivity.this, "No Entry Exists",
Toast.LENGTH_SHORT).show();
            return;
        }
        StringBuffer buffer = new StringBuffer();
        while(res.moveToNext()){
            buffer.append("Name :"+res.getString(0)+"\n");
            buffer.append("Contact :"+res.getString(1)+"\n");
            buffer.append("Date of Birth :"+res.getString(2)+"\n\n");
        }

        AlertDialog.Builder builder = new AlertDialog.Builder(MainActivity.this);

```



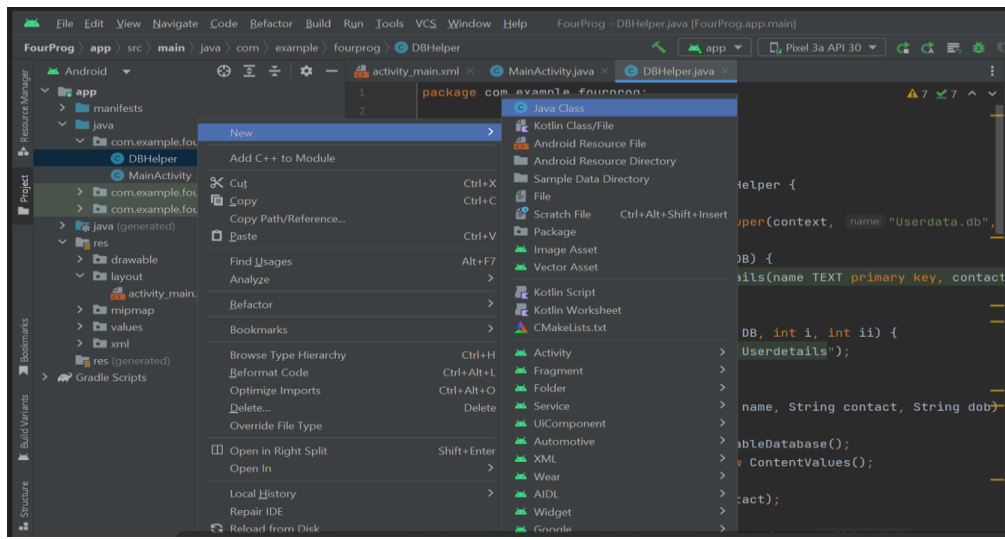
```

        builder.setCancelable(true);
        builder.setTitle("User Entries");
        builder.setMessage(buffer.toString());
        builder.show();
    }
}

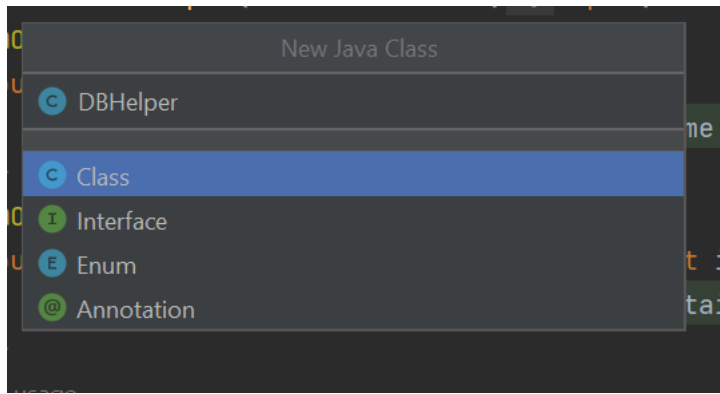
```

## DBHelper.java

Right-click ->New->Java Class



Click on Java Class



Give the name as DBHelper and create class

### **Code:**

```
package com.example.fourprog;

import android.content.ContentValues;
import android.content.Context;
import android.database.Cursor;
import android.database.sqlite.SQLiteDatabase;
import android.database.sqlite.SQLiteOpenHelper;
import androidx.annotation.Nullable;

public class DBHelper extends SQLiteOpenHelper {
    public DBHelper(Context context) {
        super(context, "Userdata.db", null, 1);
    }
    @Override
    public void onCreate(SQLiteDatabase DB) {
        DB.execSQL("create Table Userdetails(name TEXT primary key, contact TEXT, dob TEXT)");
    }
    @Override
    public void onUpgrade(SQLiteDatabase DB, int i, int ii) {
        DB.execSQL("drop Table if exists Userdetails");
    }
    public Boolean insertuserdata(String name, String contact, String dob)
    {
        SQLiteDatabase DB = this.getWritableDatabase();
        ContentValues contentValues = new ContentValues();
        contentValues.put("name", name);
        contentValues.put("contact", contact);
        contentValues.put("dob", dob);
        long result=DB.insert("Userdetails", null, contentValues);
        if(result==-1){
            return false;
        }else{
            return true;
        }
    }
    public Boolean updateuserdata(String name, String contact, String dob)
    {
        SQLiteDatabase DB = this.getWritableDatabase();
        ContentValues contentValues = new ContentValues();
        contentValues.put("contact", contact);
        contentValues.put("dob", dob);
        Cursor cursor = DB.rawQuery("Select * from Userdetails where name = ?", new
String[]{ name});
```

```

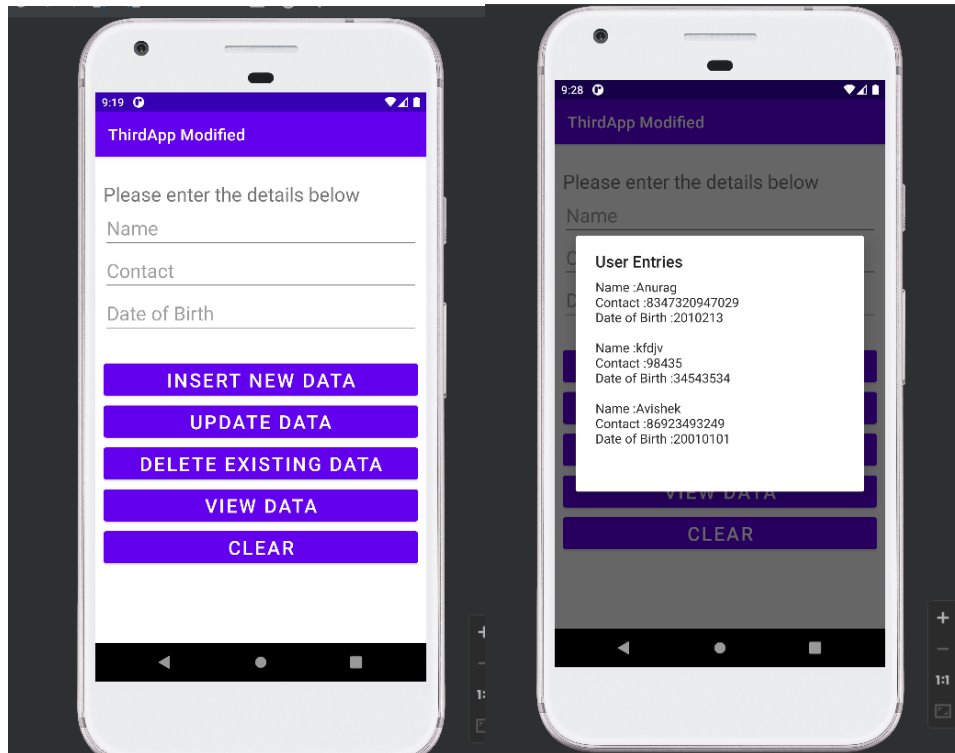
        if (cursor.getCount() > 0) {
            long result = DB.update("Userdetails", contentValues, "name=?", new
String[]{name});
            if (result == -1) {
                return false;
            } else {
                return true;
            }
        } else {
            return false;
        }
    }

    public Boolean deletedata (String name)
    {
        SQLiteDatabase DB = this.getWritableDatabase();
        Cursor cursor = DB.rawQuery("Select * from Userdetails where name = ?", new
String[]{name});
        if (cursor.getCount() > 0) {
            long result = DB.delete("Userdetails", "name=?", new String[]{name});
            if (result == -1) {
                return false;
            } else {
                return true;
            }
        } else {
            return false;
        }
    }

    public Cursor getdata ()
    {
        SQLiteDatabase DB = this.getWritableDatabase();
        Cursor cursor = DB.rawQuery("Select * from Userdetails", null);
        return cursor;
    }
}

```

### **Output:**



Prog 4 a & b

## **MainActivity.java**

```
package com.example.cam_app_4a_103;

import androidx.annotation.NonNull;
import androidx.annotation.Nullable;
import androidx.appcompat.app.AppCompatActivity;
import androidx.core.app.ActivityCompat;
import androidx.core.content.ContextCompat;

import android.Manifest;
import android.app.Activity;
import android.content.Intent;
import android.content.pm.PackageManager;
import android.graphics.Bitmap;
import android.graphics.BitmapFactory;
import android.net.Uri;
import android.os.Bundle;
import android.provider.MediaStore;
import android.view.View;
import android.widget.Button;
import android.widget.ImageView;
import android.widget.LinearLayout;
import android.widget.Toast;

import java.io.InputStream;
```

```
public class MainActivity extends AppCompatActivity {  
    private final int CAMERA_REQ_CODE = 100;  
    ImageView imgCamera;  
  
    boolean camImgfit;  
  
    Button cameraBtn, selectImgBtn;  
  
    private static final int REQUEST_CODE_STORAGE_PERMISSION = 1;  
    private static final int REQUEST_CODE_SELECT_IMAGE = 2;  
  
    @Override  
    protected void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
        setContentView(R.layout.activity_main);  
  
        imgCamera = findViewById(R.id.camImg);  
        cameraBtn = findViewById(R.id.cameraBtn);  
  
        cameraBtn.setOnClickListener(new View.OnClickListener() {  
            @Override  
            public void onClick(View view) {  
                Intent iCamera = new Intent(MediaStore.ACTION_IMAGE_CAPTURE);  
                startActivityForResult(iCamera, CAMERA_REQ_CODE);  
            }  
        });  
  
        selectImgBtn = findViewById(R.id.selectImgBtn);
```

```

selectImgBtn.setOnClickListener(new View.OnClickListener() {

    @Override

    public void onClick(View view) {

        if(ContextCompat.checkSelfPermission(

            getApplicationContext(), Manifest.permission.READ_EXTERNAL_STORAGE

        ) != PackageManager.PERMISSION_GRANTED) {

            ActivityCompat.requestPermissions(

                MainActivity.this,

                new String[] {

                    Manifest.permission.READ_EXTERNAL_STORAGE

                },

                REQUEST_CODE_STORAGE_PERMISSION

            );

        } else {

            selectImage();

        }

    }

});
}

```

```

@Override

protected void onActivityResult(int requestCode, int resultCode, @Nullable Intent data) {

    super.onActivityResult(requestCode, resultCode, data);

    if(requestCode == CAMERA_REQ_CODE && resultCode == RESULT_OK) {

        Bitmap img = (Bitmap) (data.getExtras().get("data"));

        imgCamera.setImageBitmap(img);

    }
}

```

```

if(requestCode == REQUEST_CODE_SELECT_IMAGE && resultCode == RESULT_OK) {
    if(data != null) {
        Uri selectedImageUri = data.getData();
        if(selectedImageUri != null) {
            try {
                InputStream inputStream = getContentResolver().openInputStream(selectedImageUri);
                Bitmap bitmap = BitmapFactory.decodeStream(inputStream);
                imgCamera.setImageBitmap(bitmap);
            } catch (Exception exception) {
                Toast.makeText(this, exception.getMessage(), Toast.LENGTH_SHORT).show();
            }
        }
    }
}

```

```

private void selectImage() {
    Intent intent = new Intent(Intent.ACTION_PICK, MediaStore.Images.Media.EXTERNAL_CONTENT_URI);

    if(intent.resolveActivity(getPackageManager()) != null) {
        startActivityForResult(intent, REQUEST_CODE_SELECT_IMAGE);
    }
}

```

@Override

```

public void onRequestPermissionsResult(int requestCode, @NonNull String[] permissions, @NonNull int[]
grantResults) {
    super.onRequestPermissionsResult(requestCode, permissions, grantResults);

    if(requestCode == REQUEST_CODE_STORAGE_PERMISSION && grantResults.length > 0) {
        if(grantResults[0] == PackageManager.PERMISSION_GRANTED) {
            selectImage();
        }
    }
}

```



```

    }
    else {
        Toast.makeText(this, "Permission Denied", Toast.LENGTH_SHORT).show();
    }
}
}
}
}
}

```

## 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">

    <ImageView
        android:id="@+id/camImg"
        android:layout_width="0dp"
        android:layout_height="0dp"
        android:layout_marginTop="20dp"
        android:layout_marginBottom="100dp"
        android:adjustViewBounds="false"
        android:scaleType="fitCenter"
        app:layout_constraintBottom_toTopOf="@+id/cameraBtn"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.0"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        app:srcCompat="@color/black" />

```

<Button

android:id="@+id/cameraBtn"

android:layout\_width="0dp"

android:layout\_height="wrap\_content"

android:layout\_marginBottom="10dp"

android:text="Open Camera"

app:layout\_constraintBottom\_toTopOf="@+id/selectImgBtn"

app:layout\_constraintEnd\_toEndOf="parent"

app:layout\_constraintHorizontal\_bias="0.0"

app:layout\_constraintStart\_toStartOf="parent" />

<Button

android:id="@+id/selectImgBtn"

android:layout\_width="0dp"

android:layout\_height="wrap\_content"

android:layout\_marginBottom="50dp"

android:text="Select Image"

app:layout\_constraintBottom\_toBottomOf="parent"

app:layout\_constraintEnd\_toEndOf="parent"

app:layout\_constraintHorizontal\_bias="0.0"

app:layout\_constraintStart\_toStartOf="parent" />

</androidx.constraintlayout.widget.ConstraintLayout>

AndroidManifest.xml (add the below line)

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

Q5. Create an application to demonstrate few key features of the Android framework. In particular, the application demonstrates how to send SMS text messages.

- a. How to send SMS text messages.
- b. How to dial using an in-built dialer
- c. How to send email.

**activitymain.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="com.example.smsemailphone.MainActivity">
    <TextView
        android:id="@+id/textView1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Sending SMS Example"
        android:layout_alignParentTop="true"
        android:layout_centerHorizontal="true"
        android:textSize="30dp" />
    <Button
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Send Sms"
        android:id="@+id/btnSendSMS"
        android:layout_centerHorizontal="true"
        android:layout_marginTop="48dp" />
    <Button
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Send Email"
        android:id="@+id/btnSendEmail"
        android:layout_below="@+id/btnSendSMS"
        android:layout_centerHorizontal="true"
        android:layout_marginTop="48dp" />
    <Button
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Phone"
        android:id="@+id/btnDialPhone"
        android:layout_marginTop="54dp" />
</RelativeLayout>
```

### **MainActivity:**

```
package com.example.smsprog;

import androidx.appcompat.app.AppCompatActivity;

import android.content.Intent;
import android.net.Uri;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.Toast;

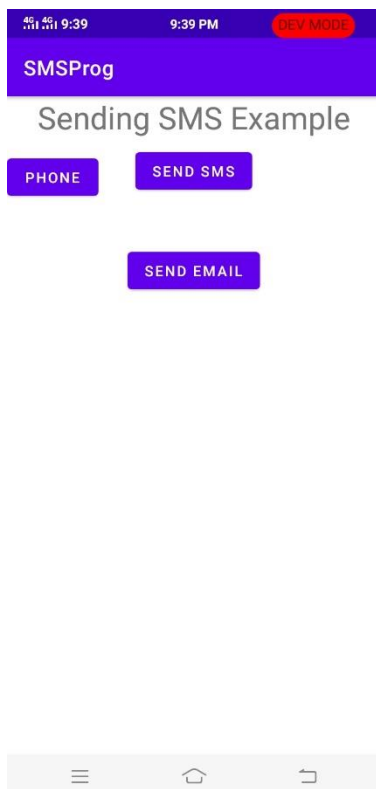
public class MainActivity extends AppCompatActivity {
    private static final int MY_PERMISSIONS_REQUEST_SEND_SMS = 0;
    Button sendBtn;
    Button btnSendEmail;
    Button btnPhone;
    String phoneNo;
    String message;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        sendBtn = (Button) findViewById(R.id.btnSendSMS);
        btnSendEmail = (Button) findViewById(R.id.btnSendEmail);
        btnPhone = (Button) findViewById(R.id.btnDialPhone);
        sendBtn.setOnClickListener(new View.OnClickListener() {
            public void onClick(View view) {
                sendSMSMessage();
            }
        });
        btnSendEmail.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                sendEmail();
            }
        });
        btnPhone.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                PhoneDial();
            }
        });
    }
    protected void sendEmail() {
```

```

Intent emailIntent = new Intent(Intent.ACTION_SEND);
emailIntent.setData(Uri.parse("mailto:"));
emailIntent.setType("text/plain");
emailIntent.putExtra(Intent.EXTRA_EMAIL, new String[]{"ba.mohan@gmail.com"});
emailIntent.putExtra(Intent.EXTRA_SUBJECT, "subject Test");
emailIntent.putExtra(Intent.EXTRA_TEXT, "Message Body Test");
startActivity(emailIntent);
}
protected void sendSMSMessage() {
    Intent sendIntent = new Intent(Intent.ACTION_VIEW);
    sendIntent.putExtra("sms_body", "default content");
    sendIntent.setType("vnd.android-dir/mms-sms");
    startActivity(sendIntent);
    Toast.makeText(getApplicationContext(), "SMS sent.",
        Toast.LENGTH_LONG).show();
} protected void PhoneDial() {
    Intent intent = new Intent(Intent.ACTION_DIAL);
    startActivity(intent); }
}

```

## Output:



Q6. Develop an app that include broadcast Receiver to receive the miss calls from the Known number and display it to the user using notification services. This same app should also fetch phone number from the inbuilt contacts using the concept of content provider.

- a. How to use broadcast receiver and notifications.
- b. How to use content providers.

6a:

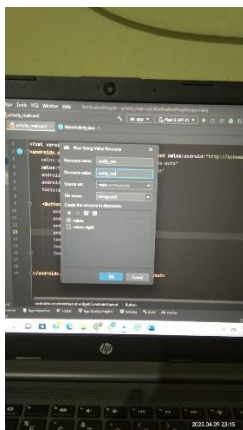
### Activity main.xml

```
<?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">

    <Button
        android:id="@+id/buttonNotify"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="@string/notify_me"
        tools:layout_editor_absoluteX="33dp"
        tools:layout_editor_absoluteY="40dp"
        tools:ignore="MissingConstraints" />

</androidx.constraintlayout.widget.ConstraintLayout>
```

Note: add a resource value



## Main\_activity.java

```
package com.example.notificationprog6a;

import android.app.Notification;
import android.app.NotificationChannel;
import android.app.NotificationManager;
import android.app.PendingIntent;
import android.app.TaskStackBuilder;
import android.content.Context;
import android.content.Intent;
import android.os.Bundle;
import android.widget.Button;

import androidx.appcompat.app.AppCompatActivity;
import androidx.core.app.NotificationCompat;

@SuppressWarnings("ALL")
public class MainActivity extends AppCompatActivity {
    Button but;
    Context context=MainActivity.this;
    String CHANNEL_ID="my_channel_01";
    CharSequence name = "my_channel";
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        but= findViewById(R.id.buttonNotify);
        NotificationManager notificationManager = (NotificationManager)
context.getSystemService(Context.NOTIFICATION_SERVICE);

        but.setOnClickListener(v -> {
            NotificationChannel mChannel = new NotificationChannel(CHANNEL_ID, name,
NotificationManager.IMPORTANCE_HIGH);
            notificationManager.createNotificationChannel(mChannel);

            Intent notificationIntent = new Intent(context, MainActivity.class);

            TaskStackBuilder stackBuilder = TaskStackBuilder.create(context);
            stackBuilder.addParentStack(MainActivity.class);
            stackBuilder.addNextIntent(notificationIntent);

            PendingIntent pendingIntent = stackBuilder.getPendingIntent(0,
PendingIntent.FLAG_IMMUTABLE);
```

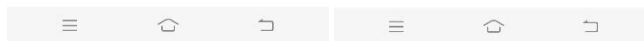
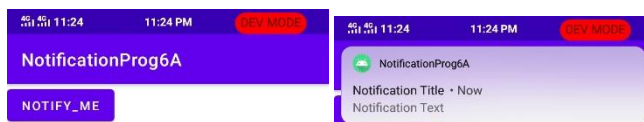


```
NotificationCompat.Builder builder = new  
NotificationCompat.Builder(context, CHANNEL_ID);
```

```
Notification notification = builder.setContentTitle("Notification Title")  
    .setContentText("Notification Text")  
    .setTicker("Test!")  
    .setSmallIcon(R.mipmap.ic_launcher)  
    .setAutoCancel(true)  
    .setContentIntent(pendingIntent).build();
```

```
notificationManager.notify(0, notification);  
});  
}  
}
```

### **Output:**



Program 6b:

**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">

    <Button
        android:id="@+id/button"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="@string/button"
        tools:layout_editor_absoluteX="33dp"
        tools:layout_editor_absoluteY="40dp"
        tools:ignore="MissingConstraints" />

</androidx.constraintlayout.widget.ConstraintLayout>
```

**Main activity.java**

```
package com.example.contentprovider6b;

import android.Manifest;
import android.annotation.SuppressLint;
import android.app.Activity;
import android.content.Intent;
import android.database.Cursor;
import android.net.Uri;
import android.os.Bundle;
import android.provider.ContactsContract;
import android.widget.Button;
import android.widget.Toast;

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

@SuppressWarnings("ALL")
public class MainActivity extends AppCompatActivity {

    public static final int RequestPermissionCode = 1 ;
    @Override
```

```

protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);
    Button button = findViewById(R.id.button);
    EnableRuntimePermission();
    button.setOnClickListener(v -> {
        Intent intent = new Intent(Intent.ACTION_PICK,
ContactsContract.Contacts.CONTENT_URI);
        startActivityForResult(intent, 1);
    });
}

public void EnableRuntimePermission(){

    if (ActivityCompat.shouldShowRequestPermissionRationale(MainActivity.this,
Manifest.permission.READ_CONTACTS))
    {
        Toast.makeText(MainActivity.this,"CONTACTS permission allows us to Access
CONTACTS app", Toast.LENGTH_LONG).show();
    } else {
        ActivityCompat.requestPermissions(MainActivity.this,new String[]{
Manifest.permission.READ_CONTACTS}, RequestPermissionCode);
    }
}

@Override
public void onActivityResult(int requestCode, int resultCode, Intent data){
    super.onActivityResult(requestCode, resultCode, data);
    if (requestCode == 1) {
        if (resultCode == Activity.RESULT_OK) {
            Uri contactData = data.getData();
            try (Cursor c = managedQuery(contactData, null, null, null, null)) {
                if (c.moveToFirst()) {
                    String id =
c.getString(c.getColumnIndexOrThrow(ContactsContract.Contacts._ID));
                    @SuppressWarnings("Range")
                    String hasPhone =
c.getString(c.getColumnIndex(ContactsContract.Contacts.HAS_PHONE_NUMBER));
                    if (hasPhone.equalsIgnoreCase("1")) {
                        Cursor phones =
getContentResolver().query(ContactsContract.CommonDataKinds.Phone.CONTENT_URI,
null, ContactsContract.CommonDataKinds.Phone.CONTACT_ID + " = " + id, null, null);
                        phones.moveToFirst();
                        @SuppressWarnings("Range")
                        String cNumber =
phones.getString(phones.getColumnIndex(ContactsContract.CommonDataKinds.Phone.NU

```

```
MBER));
        Toast.makeText(getApplicationContext(), cNumber,
Toast.LENGTH_SHORT).show();
        phones.close();
    }
}
catch(Exception e)
{
    e.printStackTrace();
}
}
}
}
```

## AndriodManifest.xml

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

**Output:**



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 Manifest.xml**

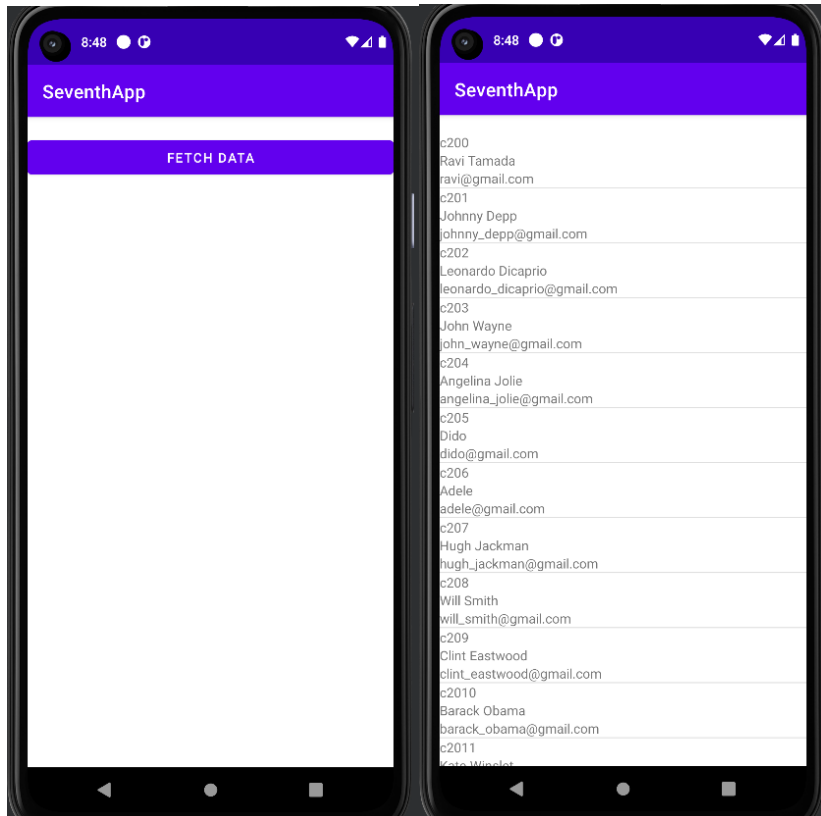
```

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

```

### **Output:**





Q8. Develop an android app on Google Map, and should provide following functions.

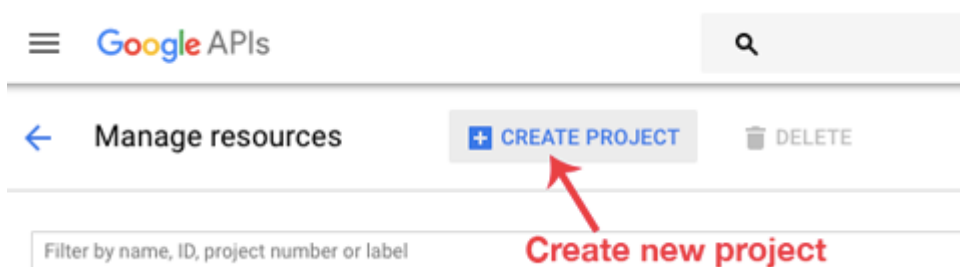
- a. How to incorporate Google Maps into an application.
- b. How to register for and receive GPS location information.
- c. How to create Google Maps Overlays.
- d. Accept city name from user and marks it on map.
- e. Explore features like Zoom and map types.

#### Steps For Getting The Google Maps Api Key:

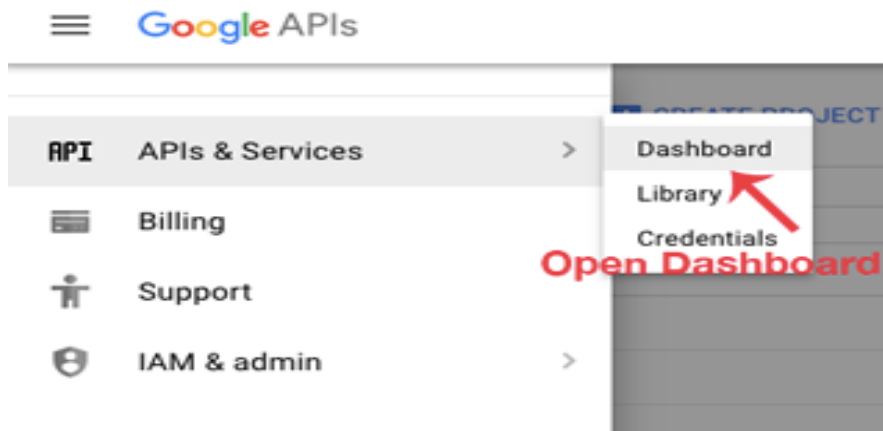
An API key is needed to access the [Google Maps](https://developers.google.com/maps/) servers. This key is free and you can use it with any of your applications. If you haven't created project, you can follow the below steps to get started:

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

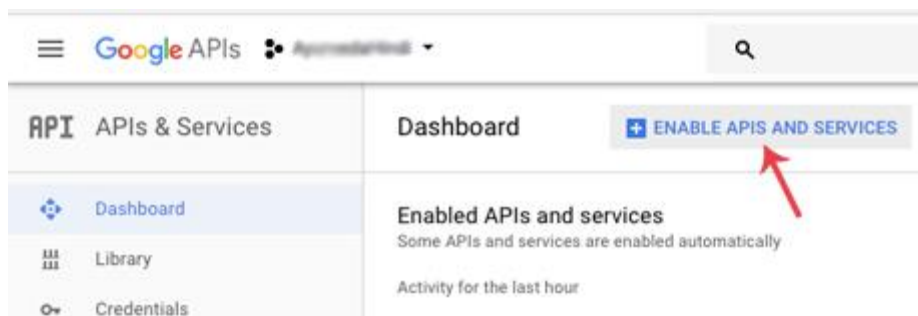
**Step 2:** Now create new project. You can create new project by clicking on the **Create Project** button and give name to your project.



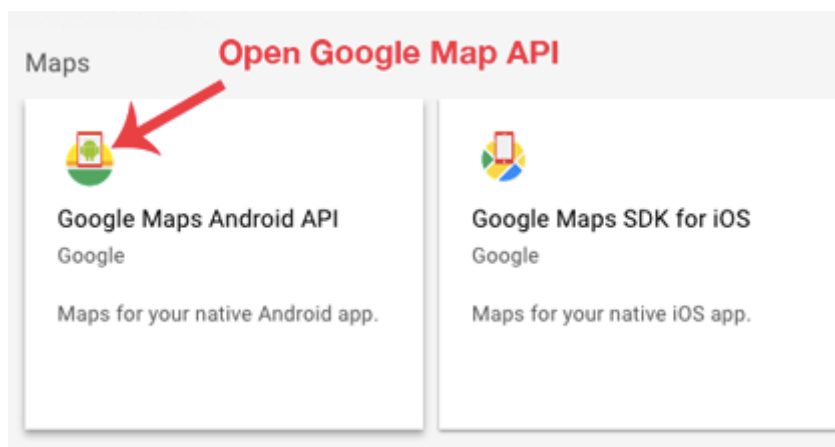
**Step 3:** Now click on APIs & Services and open **Dashboard** from it.



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

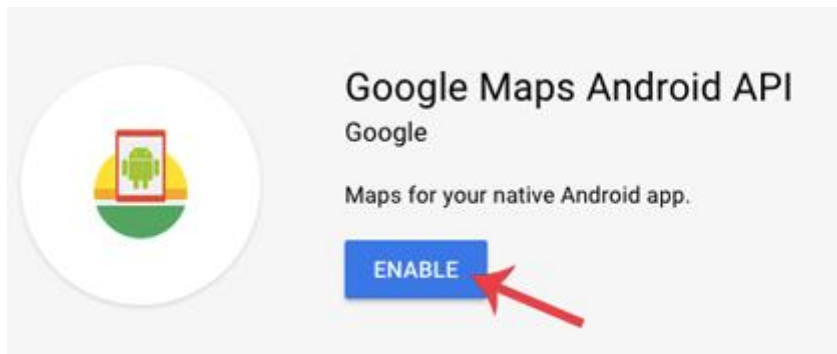


**Step 5:** Now open Google Map Android API.

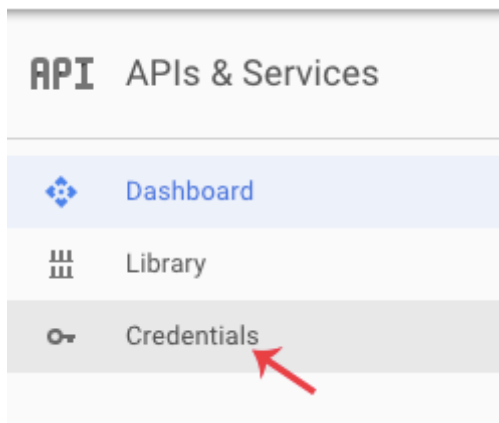


**Step 6:** Now enable the Google Maps Android API.

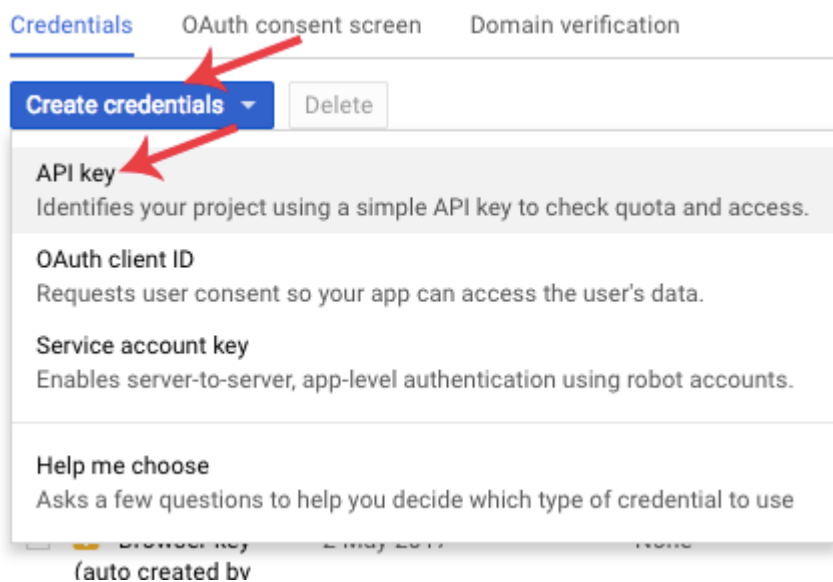
**Step 6:** Now enable the Google Maps Android API.



**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.

## API key created

Use this key in your application by passing it with the `key=API_KEY` parameter.

Your API key

AIzaSyCJg2Q1AFtIWKH0g\_bhJpdv1S8Hnd7HhCSY



 Restrict your key to prevent unauthorised use in production.

[CLOSE](#) [RESTRICT KEY](#)

### 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.MainActivity" />
```

```
<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>

```

### **AndroidManifest.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:supportRtl="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:

