



# MOBILE APPLICATION DEVELOPMENT

## DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

### Vision of the Department

To facilitate quality education by focusing on assimilation, generation and dissemination of knowledge in the area of Computer Science & Engineering to transform students into socially responsible engineers.

### Mission of the Department

- Equip our graduates with the knowledge by student centric teaching-learning process and expertise to contribute significantly to the software industry and to continue to grow professionally.
- To train socially responsible, disciplined engineers who work with good leadership skills and can contribute for nation building.
- To make our graduates aware of cutting-edge technologies and make them industry-ready engineers.
- To shape the department into a Centre of academic and research excellence.

### Program Educational Objectives

PEO-1	To provide the graduates with <b><i>solid foundation in Computer Science and Engineering</i></b> along with the fundamentals of Mathematics and Sciences with a view to impart in them high quality technical skills like modeling, analyzing, designing, programming and implementation with global competence and helps the graduates for life-long learning.
PEO-2	To prepare and motivate graduates with <b><i>recent technological developments related to core subjects</i></b> like Programming, Databases, Design of Compilers and Network Security aspects and future technologies so as to contribute effectively for Research & Development by participating in professional activities like publishing and seeking copy rights.
PEO-3	To train graduates to choose a <b><i>decent career option either in high degree of employability/Entrepreneur or, in higher education</i></b> by empowering students with ethical administrative acumen, ability to handle critical situations and training to excel in competitive examinations.
PEO-4	To train the graduates to have <b><i>basic interpersonal skills and sense of social responsibility</i></b> that paves them a way to become good team members and leaders

**VASIREDDY VENKATADRI INSTITUTE OF TECHNOLOGY, NAMBUR**  
**DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING**

**Course** : B. Tech  
**Branch** : CSE  
**Regulation** : R20  
**Code** : 20CS4C01  
**Lab Name** : Mobile Application Development Lab(SOC)  
**Total Marks** 50  
**External Lab Marks** 50

**Evaluation:**

**EXTERNAL EXAMINATION**

**MAX MARKS: 50**

**SCHEME OF EVALUATION**

- |                         |       |
|-------------------------|-------|
| • DAY TO DAY EVALUATION | : 10M |
| • RECORD                | : 5M  |
| • DESCRIPTION           | : 10M |
| • EXECUTION             | : 10M |
| • VIVA-VOCE             | : 15M |

$$10+5+10+10+15=50$$

**MODEL POSTING OF MARKS ON THE SCRIPT:**

## Course Objectives & Course Outcomes:

Mobile Application Development course is designed to quickly get you up to write applications for Android devices. The student will learn the basics of Android platform, get to understand the application frontiers and able to design his own applications.

### Course Outcomes:

Students who complete this course will be able to:

At the end of the course the student should be able to:

**CO1** : Implement Basic Mobile applications. (K3)

**CO2** : Design data flow for the Mobile App.(K4)

**CO3** : Implement GPS tracking Applications as a case study. (K3)

**CO4** : Deploy Mobile Apps with Fire base. (K3)

### CO/ PO, PSO Mapping:

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2
CO1	-	-	3	3	3	-	-	-	2	2	2	1	2	-
CO2	-	-	3	3	3	-	-	-	2	2	2	1	2	-
CO3	-	-	3	3	3	-	-	-	2	2	2	1	2	-
CO4	-	-	3	3	3	-	-	-	2	2	2	1	2	-

### List of Experiments:

1. Procedure to install Android Studio.
2. Develop a sample "Hello World Application.
3. Create an application that takes the name from a text box and shows hello message along with the name entered in text box, when the user clicks the OK button.
4. Create a screen that has input boxes for User Name, Password, Address, Gender(radio buttons for male and female), Age (numeric), Date of Birth (Date Picket), State (Spinner) and a Submit button. On clicking the submit button, print all the data below the Submit Button (use any layout).
5. Design an android application to create page using Intent and one Button and pass the Values from one Activity to second Activity.
6. Design an android application Send SMS using Intent.
7. Create an android application using Fragments.
8. Design an android application for menu.
9. Design an android application to display list of fruits using Listview
10. . Design an android application to display list of users using Recyclerview from internet.

## Experiment 1:

AIM: Installing Android Studio in Windows / Mac

### Step 1 - System Requirements

The required tools to develop Android applications are open source and can be downloaded from the Web. Following is the list of software's you will need before you start your Android application programming.

Java JDK8 or later version

Java Runtime Environment (JRE) 8 or above

Android Studio

### Step 2 - Setup Android Studio

Android Studio is the official IDE for android application development. It works based on IntelliJ IDEA, You can download the latest version of android studio from [Android Studio Download](#), If you are new to installing Android Studio on windows, you will find a file, which is named as android-studio-2021-2.1.15-windows.exe. So just download and run on windows machine according to android studio wizard guideline.

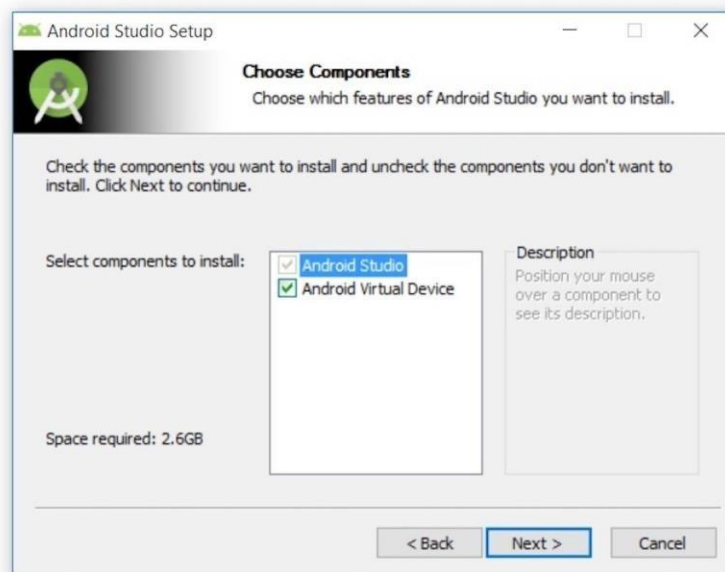
If you are installing Android Studio on Mac or Linux, You can download the latest version from [Android Studio Mac Download](#), or [Android Studio Linux Download](#), check the instructions provided along with the downloaded file for Mac OS and Linux. This tutorial will consider that you are going to setup your environment on Windows machine having Windows 8.1 operating system. Installation

So let's launch Android Studio.exe, Make sure before launch Android Studio, Our Machine should require installed Java JDK. To install Java JDK, take a references of [Android environment setup](#)



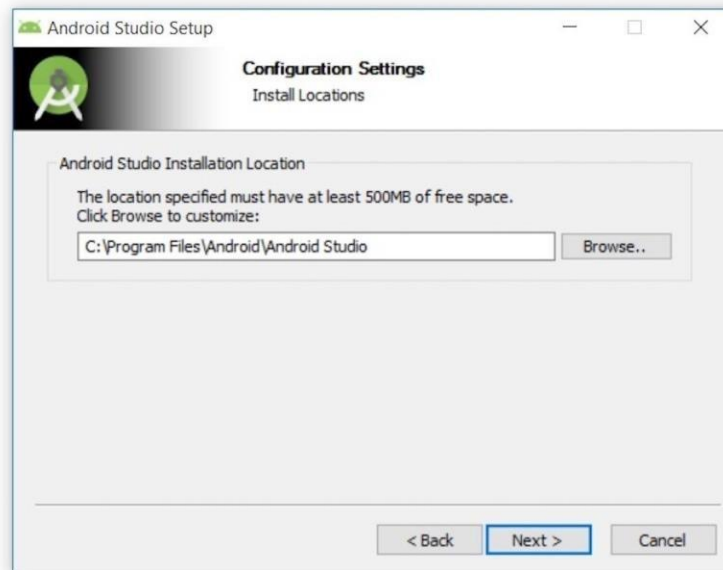
*Figure 1: Installing Android Studio*

Clicking Next took me to the following panel, which provides the option to decline installing an Android Virtual Device (AVD).



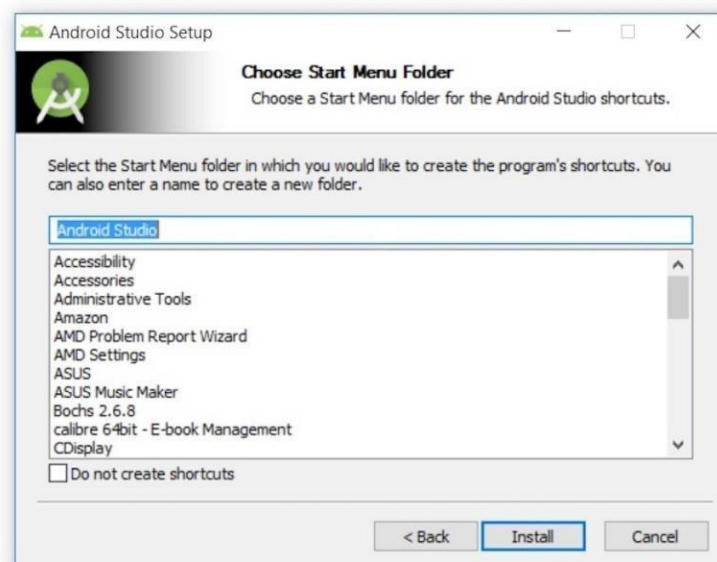
*Figure 2: Install AVD or not?*

Choose to keep the default settings. After clicking Next, It will be taken to the Configuration Settings panel, where choose where to install Android Studio.



*Figure 3: The installation location must have at least 500 MB free space*

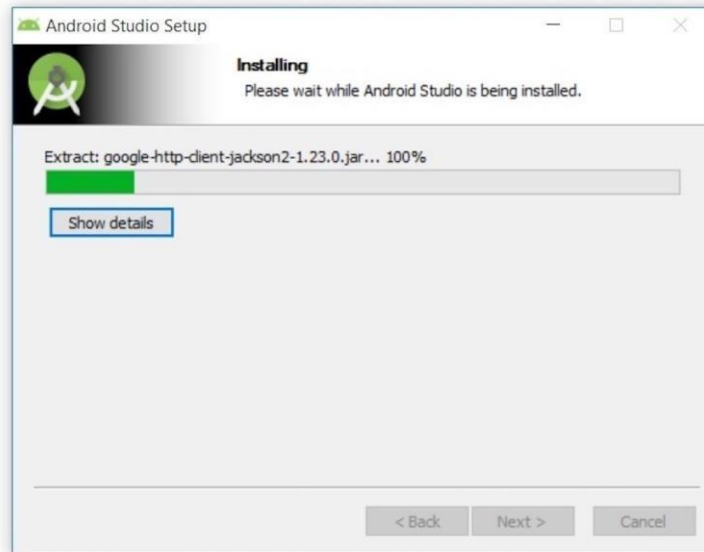
Keep the default installation location and clicked Next, and was greeted with the Choose Start Menu Folder panel.



*Figure 4: Select the folder in which to store Android Studio Shortcut*

Keeping the default setting and clicked Install. The following Installing panel appeared:





*Figure 5: This panel shows progress of installation*

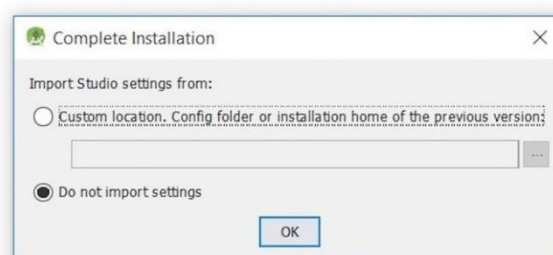
After clicking Next, the installer presented the Completing Android Studio Setup panel.



*Figure 6: Leave the Start Android Studio checkbox checked to run this software*

### Step 3 - Running Android Studio

The first time Android Studio runs, it presents a Complete Installation dialog box that offers the option of importing settings from a previous installation.



The next level of installation should contain selecting the activity to mobile, it specifies the default layout for Applications.

#### Step 4 - Your first Android Studio mobile app

We'll start with a variation on the "Hello, World" application: a little mobile app that displays a "Welcome to Android" message.

In the steps that follow, we'll start a new Android Studio project and get to know the main window. Welcome to Android Studio dialog box. From here, click Start a new Android Studio project. Android Studio will respond with the Create New Project dialog box.

Enter W2A (Welcome to Android) as the application name and vvitcse.com as the company domain name. On my desktop, I observed C:\Users \AndroidStudioProjects\W2A as the project location. Click Next to select your target devices.

Android Studio lets you select *form factors*, or categories of target devices, for every app you create. I kept the default setting.

Click **Next**, and you will be given the opportunity to choose a template for your app's main activity. For now we'll stick with **Empty Activity**. Select this template (if necessary) and click **Next**.

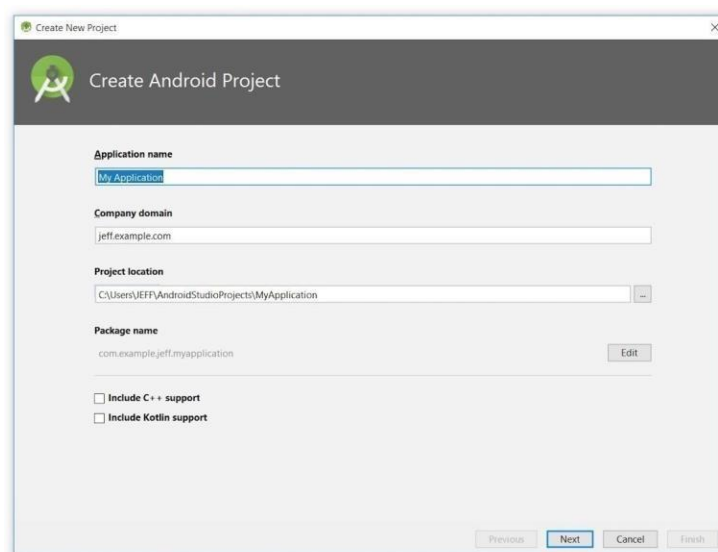
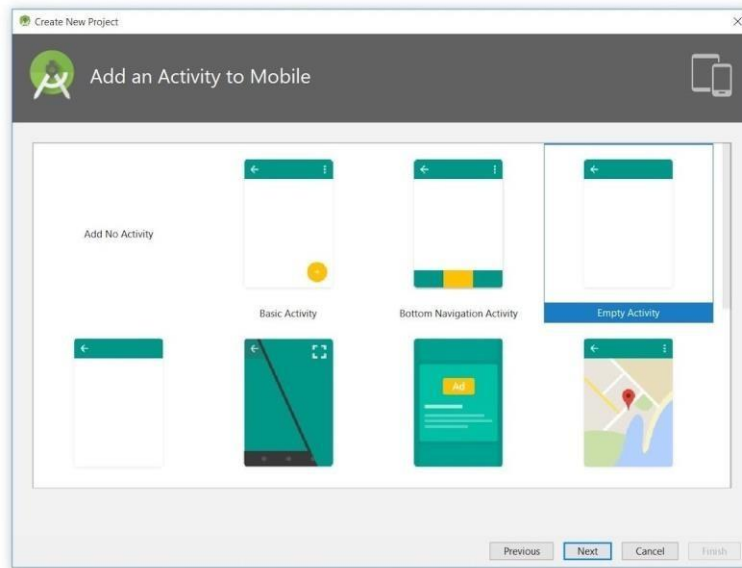


Figure 7: Create new android project



*Figure 8: Specify an activity template*

Next you'll customize the activity.

Enter W2A as the activity name and main as the layout name, and click Next to complete this step.

At the final stage it going to be open development tool to write the application code.

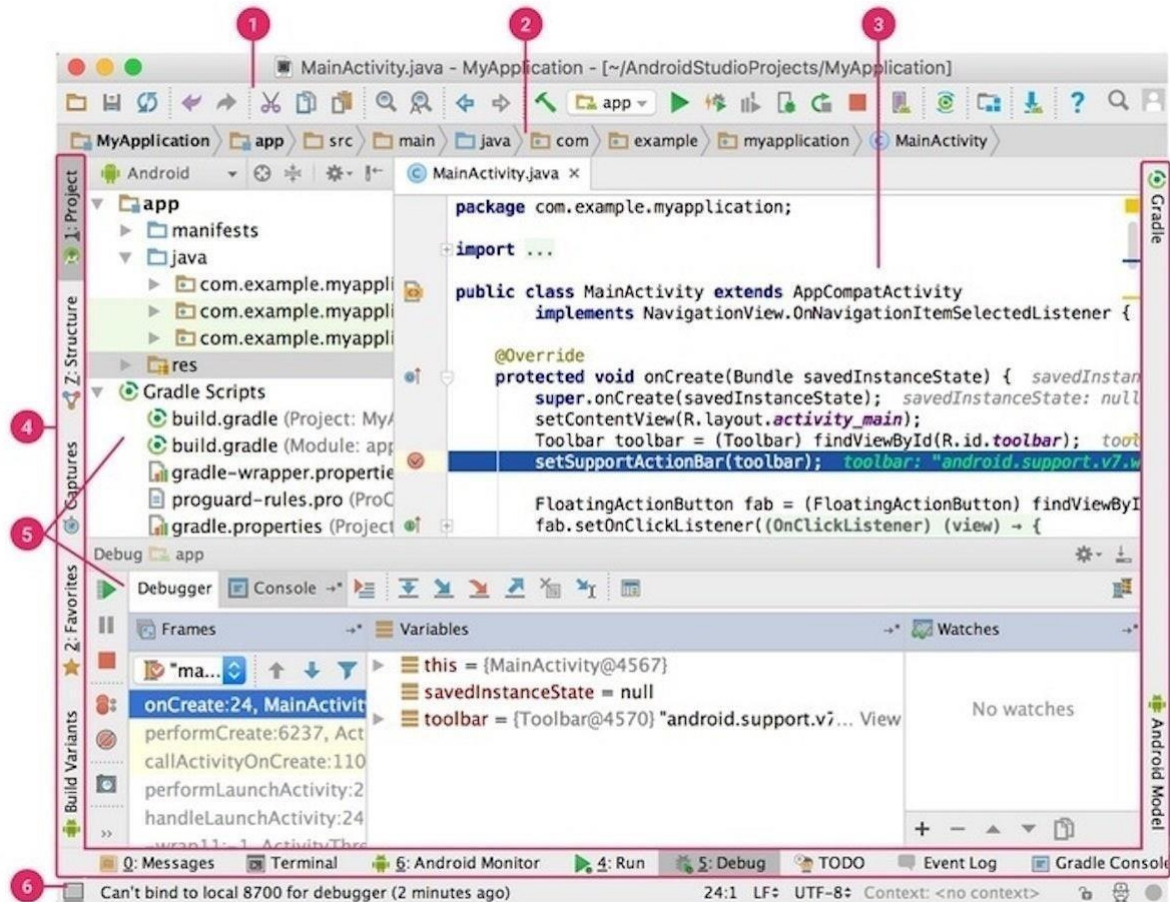
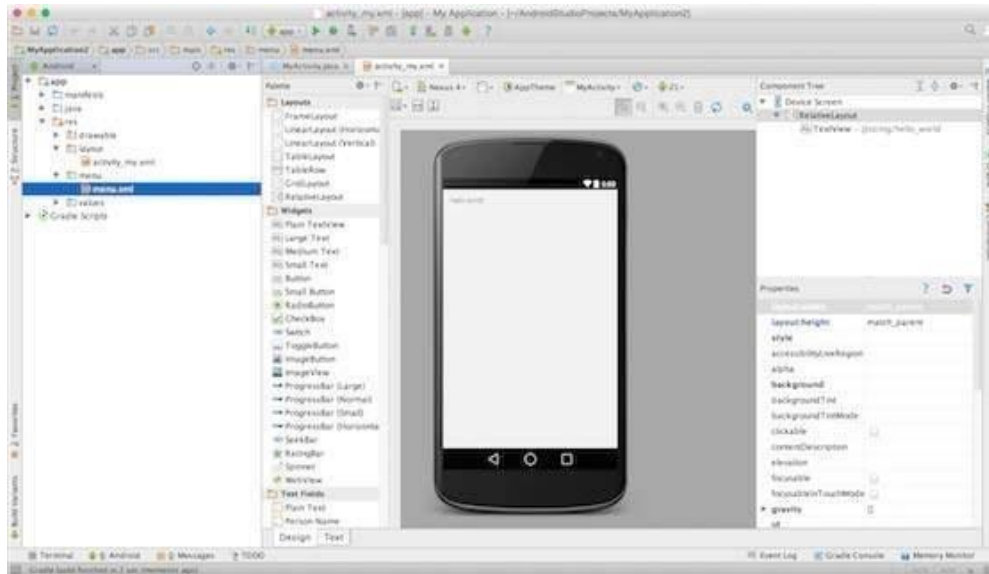


Figure 9: Different sections in Android Studio

- 1 The **toolbar** lets you carry out a wide range of actions, including running your app and launching Android tools.
- 2 The **navigation bar** helps you navigate through your project and open files for editing. It provides a more compact view of the structure visible in the **Project** window.
- 3 The **editor window** is where you create and modify code. Depending on the current file type, the editor can change. For example, when viewing a layout file, the editor displays the Layout Editor.
- 4 The **tool window bar** runs around the outside of the IDE window and contains the buttons that allow you to expand or collapse individual tool windows.
- 5 The **tool windows** give you access to specific tasks like project management, search, version control, and more. You can expand them and collapse them.
- 6 The **status bar** displays the status of your project and the IDE itself, as well as any warnings or messages.

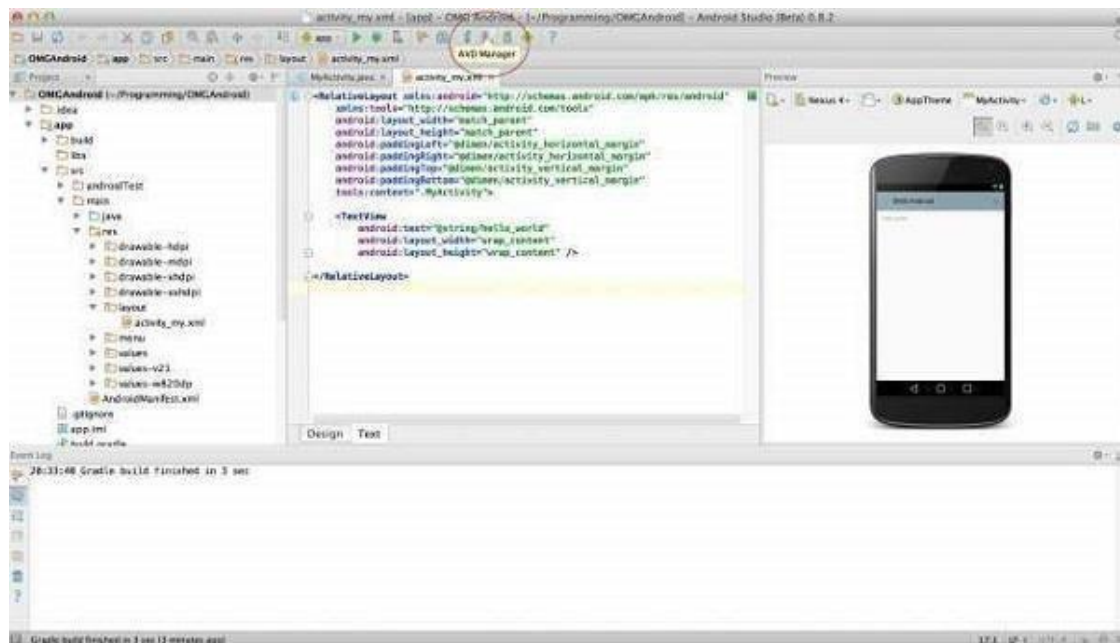
Figure 10: The main window presenting toolbar, editor window(etc.)

At the final stage it going to be open development tool to write the application code.

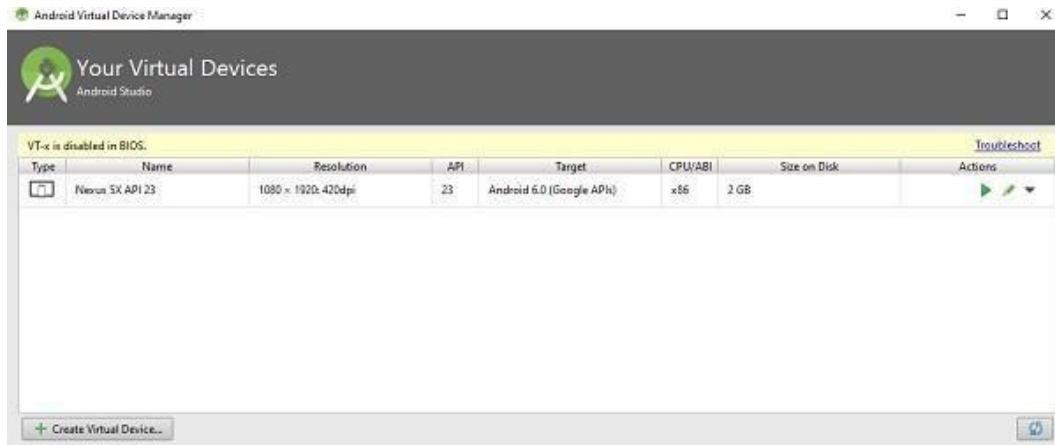


### Step 5 - Create Android Virtual Device

To test your Android applications, you will need a virtual Android device. So before we start writing our code, let us create an Android virtual device. Launch Android AVD Manager. Clicking AVD\_Manager icon as shown below



After Click on a virtual device icon, it going to be shown by default virtual devices which are present on your SDK, or else need to create a virtual device by clicking Create new Virtual device button



If your AVD is created successfully it means your environment is ready for Android application development. If you like, you can close this window using top-right cross button. Better you re- start your machine and once you are done with this last step, you are ready to proceed for your first Android example but before that we will see few more important concepts related to Android Application Development.

### Exercise Program

Write the steps in installing Android Studio in Windows and Linux.



## Experiment – 2

**AIM:** To design an android application to display Hello World

### Main Activity File

```
package com.vvit.myapplication;

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

public class MainActivity extends AppCompatActivity {

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

### The Layout File

The **activity\_main.xml** is a layout file available in res/layout directory, that is referenced by your application when building its interface. You will modify this file very frequently to change the layout of your application. For your "Hello World!" application, this file will have following content related to default layout –

```
<?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">

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:padding="16dp"
        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>
```

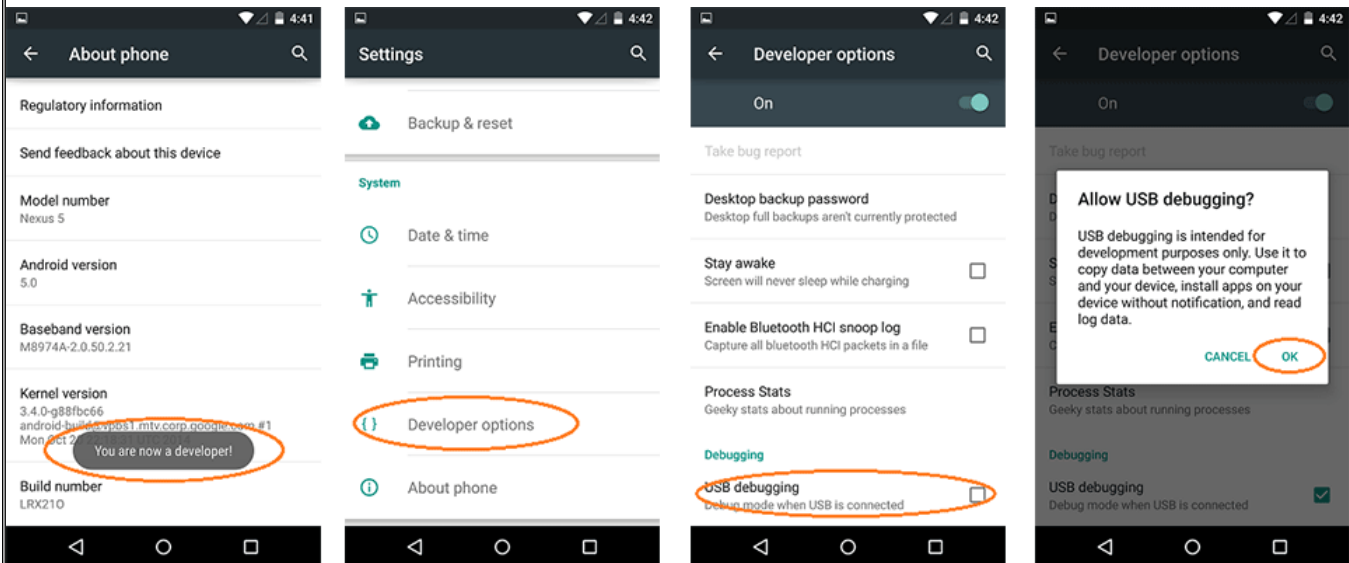
### I) Running app on Phone:

#### Connect your Phone to Computer

Plug in your device to your computer with a USB cable. If you're developing on Windows, you might need to install this [universal ADB USB driver](#) or find your [specific USB driver for your device](#).

## Enable USB Debugging

The next step is to enable USB debugging so your phone can interact with your computer in a developer mode.



The following steps are needed:

1. (Windows Only) Install [this ADB Driver](#)
2. Plug-in your Android Device to Computer via USB
3. Open the "Settings" App on the Device
4. Scroll down to bottom to find "About phone" item
5. Scroll down to bottom to find "Build number" section
6. Tap on "Build Number" 7 times in quick succession
7. You should see the message "You are now a developer!"
8. Go back to main "Settings" page
9. Scroll down bottom to find "Developer options" item
10. Turn on "USB Debugging" switch and hit "OK"
11. Unplug and re-plug the device
12. Dialog appears "Allow USB Debugging?"
13. Check "Always allow from this computer" and then hit "OK"

## Running your App


Now, we can launch apps from Android Studio onto our device:

1. Select one of your projects and click "Run" from the toolbar.



2. In the "Choose Device" window that appears, select the "Choose a running device" radio button, select the device, and click OK.

## II) Running app on Emulator(AVD)

To run the app from Android studio, open one of your project's activity files and click Run  icon from the tool bar. Android studio installs the app on your AVD and starts it and if everything is fine with your set-up and application, it will display following Emulator window –Once Gradle finishes building, Android Studio should install the app on your connected device and start it.



Hello World!



### Experiment – 3

**AIM:** Create an application that takes the name from a text box and shows hello message along with the name entered in text box, when the user clicks the OK button.

#### MainActivity.java

```
package com.vvit.myapplication;

import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.TextView;

import androidx.appcompat.app.AppCompatActivity;

public class MainActivity extends AppCompatActivity {
    // These are the global variables
    EditText editName, editPassword;
    Button buttonSubmit;
    TextView result;

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

        editName = findViewById(R.id.editName);
        editPassword = findViewById(R.id.editPassword);
        result = findViewById(R.id.tvResult);
        buttonSubmit = findViewById(R.id.buttonSubmit);
        buttonSubmit.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                String name = editName.getText().toString();
                String password = editPassword.getText().toString();
                result.setText("Name:\t" + name + "\nPassword:\t" +
password);
            }
        });
    }
}
```

#### 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">

    <EditText
```

```

        android:id="@+id/editName"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_marginStart="16dp"
        android:layout_marginTop="16dp"
        android:layout_marginEnd="16dp"
        android:layout_marginBottom="16dp"
        android:ems="10"
        android:hint="Enter Username"
        android:inputType="textPersonName"
        android:minHeight="48dp"
        app:layout_constraintBottom_toTopOf="@+id/editPassword"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent" />

```

<EditText

```

        android:id="@+id/editPassword"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_marginStart="16dp"
        android:layout_marginTop="16dp"
        android:layout_marginEnd="16dp"
        android:layout_marginBottom="8dp"
        android:ems="10"
        android:inputType="textPassword"
        android:hint="Enter Password"
        android:minHeight="48dp"
        app:layout_constraintBottom_toTopOf="@+id/buttonSubmit"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/editName" />

```

<Button

```

        android:id="@+id/buttonSubmit"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginStart="156dp"
        android:layout_marginTop="16dp"
        android:layout_marginEnd="161dp"
        android:text="Submit"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/editPassword" />

```

<TextView

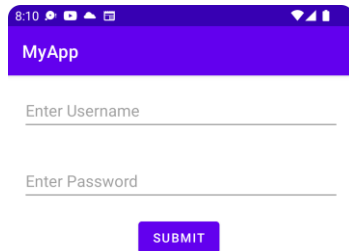
```

        android:id="@+id/tvResult"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginStart="172dp"
        android:layout_marginTop="250dp"
        android:layout_marginEnd="181dp"
        android:text="TextView"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/buttonSubmit" />

```

</androidx.constraintlayout.widget.ConstraintLayout>

**Result:**



8:10

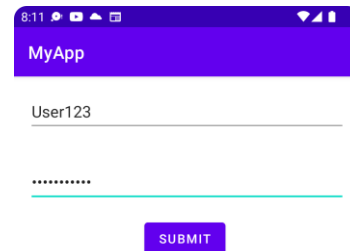
MyApp

Enter Username

Enter Password

SUBMIT

TextView



8:11

MyApp

User123

password123

SUBMIT

Name: User123  
Password: password123

## Experiment – 4

**AIM:** Create a screen that has input boxes for User Name, Password, Address, Gender (radio buttons for male and female), Age (numeric) and a Submit button. On clicking the submit button, print all the data below the Submit Button (use any layout)

### MainActivity.java

```
package com.vvit.myapplication;

import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.RadioButton;
import android.widget.RadioGroup;
import android.widget.TextView;

import androidx.appcompat.app.AppCompatActivity;

public class MainActivity extends AppCompatActivity {
    // These are the global variables
    EditText editName, editPassword, editAddress;
    TextView result;
    RadioGroup radioSexGender;
    private RadioButton radioGenderButton;
    Button buttonSubmit;

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

        editName = findViewById(R.id.editName);
        editPassword = findViewById(R.id.editPassword);
        editAddress = findViewById(R.id.editAddress);
        radioSexGender = findViewById(R.id.radioGender);
        result = findViewById(R.id.tvResult);
        buttonSubmit = findViewById(R.id.buttonSubmit);
        buttonSubmit.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                String name = editName.getText().toString();
                String password = editPassword.getText().toString();
                String address = editAddress.getText().toString();
                int selectedId = radioSexGender.getCheckedRadioButtonId();
                radioGenderButton = (RadioButton) findViewById(selectedId);
                result.setText("Name:\t" + name + "\nPassword:\t" +
                    password + "\nAddress:\t" + address + "\nGender:\t" +
                    radioGenderButton.getText().toString());
            }
        });
    }
}
```

### 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">

    <TextView
        android:id="@+id/textView"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginStart="175dp"
        android:layout_marginTop="30dp"
        android:layout_marginEnd="175dp"
        android:layout_marginBottom="28dp"
        android:text="TextView"
        app:layout_constraintBottom_toTopOf="@+id/editName"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent" />

    <EditText
        android:id="@+id/editName"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_marginStart="16dp"
        android:layout_marginTop="32dp"
        android:layout_marginEnd="16dp"
        android:layout_marginBottom="32dp"
        android:ems="10"
        android:inputType="textPersonName"
        android:hint="Enter Username"
        app:layout_constraintBottom_toTopOf="@+id/editPassword"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="1.0"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/textView" />

    <EditText
        android:id="@+id/editPassword"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_marginStart="16dp"
        android:layout_marginTop="32dp"
        android:layout_marginEnd="16dp"
        android:layout_marginBottom="56dp"
        android:ems="10"
        android:inputType="textPassword"
        app:layout_constraintBottom_toTopOf="@+id/radioGender"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/editName" />

    <EditText
        android:id="@+id/editAddress"
        android:layout_width="0dp"
        android:layout_height="wrap_content"
        android:layout_marginStart="16dp"
```

```

        android:layout_marginTop="32dp"
        android:layout_marginEnd="16dp"
        android:ems="10"
        android:inputType="textPostalAddress"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/radioGender" />

<RadioGroup
    android:id="@+id/radioGender"
    android:layout_width="0dp"
    android:orientation="horizontal"
    android:layout_height="wrap_content"
    android:layout_marginStart="16dp"
    android:layout_marginTop="26dp"
    android:layout_marginEnd="16dp"
    android:layout_marginBottom="32dp"
    app:layout_constraintBottom_toTopOf="@+id/editAddress"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    android:tooltipText="Gender"
    app:layout_constraintTop_toBottomOf="@+id/editPassword">

    <RadioButton
        android:id="@+id/radioButton3"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:padding="16dp"
        android:layout_marginLeft="32dp"
        android:text="Male" />

    <RadioButton
        android:id="@+id/radioButton4"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:padding="16dp"
        android:text="Female" />

</RadioGroup>

<Button
    android:id="@+id/buttonSubmit"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginStart="156dp"
    android:layout_marginTop="80dp"
    android:layout_marginEnd="161dp"
    android:text="Submit"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/editAddress" />

<TextView
    android:id="@+id/tvResult"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginStart="172dp"
    android:layout_marginTop="16dp"
    android:layout_marginEnd="181dp"
    android:layout_marginBottom="60dp"

```

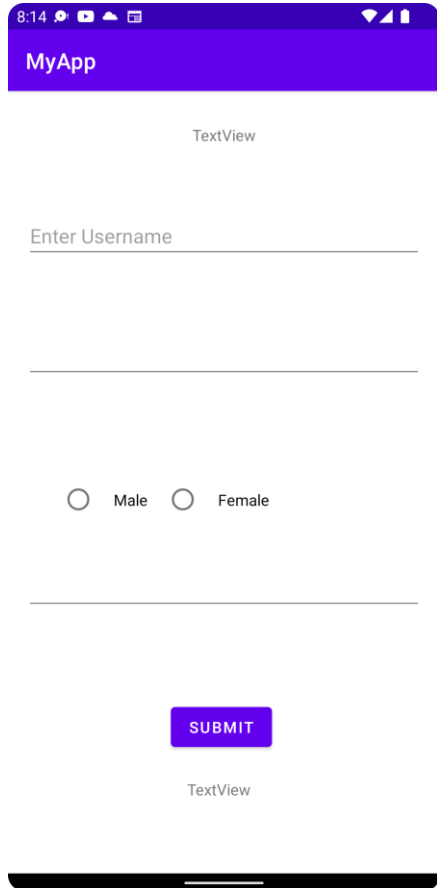
```

    android:text="TextView"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/button" />

```

</androidx.constraintlayout.widget.ConstraintLayout>

### Result:





## Experiment – 5

**AIM:** To design an android application to design a page using Intent and one Button and pass the Values from one Activity to second Activity.

### Main Activity.java

```
package com.vvit.myapplication;

import android.os.Bundle;
import android.app.Activity;
import android.content.Intent;
import android.view.View;
import android.view.View.OnClickListener;
import android.widget.Button;
import android.widget.EditText;

public class MainActivity extends Activity {
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        final EditText et = findViewById(R.id.editName);
        final EditText et1 = findViewById(R.id.editPlace);
        Button send = findViewById(R.id.buttonNext);
        send.setOnClickListener(new OnClickListener() {
            @Override
            public void onClick(View arg0) {
                String s = et.getText().toString();
                String s1 = et1.getText().toString();
                Intent it = new Intent(MainActivity.this,
SecondActivity.class);
                it.putExtra("name", s);

                it.putExtra("place", s1);
                startActivity(it);
            }
        });
    }
}
```

### SecondActivity.java

```
package com.vvit.myapplication;

import android.os.Bundle;
import android.app.Activity;
import android.widget.TextView;

public class SecondActivity extends Activity {
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_second);
        TextView tv = findViewById(R.id.textView2);
        TextView tv1 = findViewById(R.id.textView3);
        tv.setText(getIntent().getExtras().getString("name"));
        tv1.setText(getIntent().getExtras().getString("place"));
    }
}
```

## MainActivity.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"
android:padding="16dp"
tools:context=".MainActivity">

    <Button
        android:id="@+id/buttonNext"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginStart="145dp"
        android:layout_marginTop="84dp"
        android:layout_marginEnd="144dp"
        android:layout_marginBottom="365dp"
        android:text="Next Screen"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/editPlace" />

    <EditText
        android:id="@+id/editName"
        android:layout_width="0dp"
        android:layout_height="wrap_content"
        android:layout_marginStart="16dp"
        android:layout_marginTop="75dp"
        android:layout_marginEnd="16dp"
        android:layout_marginBottom="53dp"
        android:ems="10"
        android:inputType="textPersonName"
        android:minHeight="48dp"
        android:hint="Enter Name"
        app:layout_constraintBottom_toTopOf="@+id/editPlace"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent" />

    <EditText
        android:id="@+id/editPlace"
        android:layout_width="0dp"
        android:layout_height="wrap_content"
        android:layout_marginStart="16dp"
        android:layout_marginTop="53dp"
        android:layout_marginEnd="16dp"
        android:ems="10"
        android:inputType="textPersonName"
        android:minHeight="48dp"
        android:hint="Enter Place"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/editName" />
</androidx.constraintlayout.widget.ConstraintLayout>
```

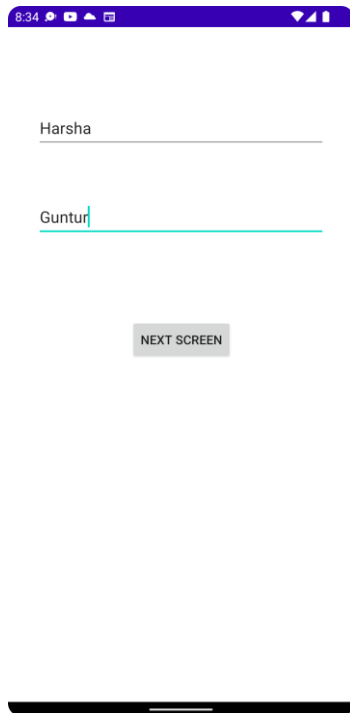
## SecondActivity.xml

```
<?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=".SecondActivity">

    <TextView
        android:id="@+id/textView2"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginStart="70dp"
        android:layout_marginTop="104dp"
        android:layout_marginEnd="283dp"
        android:layout_marginBottom="145dp"
        android:text="TextView"
        app:layout_constraintBottom_toTopOf="@+id/textView3"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent" />

    <TextView
        android:id="@+id/textView3"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginStart="74dp"
        android:layout_marginTop="145dp"
        android:layout_marginEnd="279dp"
        android:layout_marginBottom="443dp"
        android:text="TextView"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/textView2" />
</androidx.constraintlayout.widget.ConstraintLayout>
```

**Result:**



## Experiment – 6

**AIM:** To design an android application to Send SMS using Intent.

### MainActivity.java

```
package com.vvit.myapplication;

import android.app.Activity;
import android.os.Bundle;
import android.telephony.SmsManager;
import android.view.View;
import android.view.View.OnClickListener;
import android.widget.Button;
import android.widget.EditText;
import android.widget.Toast;

public class MainActivity extends Activity {
    private EditText phone, message;
    private Button send;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        phone = findViewById(R.id.editPhoneNumber);
        message = findViewById(R.id.editMessage);
        send = findViewById(R.id.buttonSend);

        send.setOnClickListener(new OnClickListener() {
            @Override
            public void onClick(View v) {
                String phoneNumber = phone.getText().toString();
                String msg = message.getText().toString();

                SmsManager sms = SmsManager.getDefault();
                sms.sendTextMessage(phoneNumber, null, msg, null, null);
                Toast.makeText(MainActivity.this, "Message Sent",
                    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"
    android:padding="16dp"
    tools:context=".MainActivity">

    <Button
        android:id="@+id/buttonSend"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginStart="151dp"
```

```

        android:layout_marginTop="96dp"
        android:layout_marginEnd="150dp"
        android:layout_marginBottom="303dp"
        android:text="Send"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/editMessage" />

<EditText
    android:id="@+id/editPhoneNumber"
    android:layout_width="0dp"
    android:layout_height="wrap_content"
    android:layout_marginStart="16dp"
    android:layout_marginTop="96dp"
    android:layout_marginEnd="16dp"
    android:ems="10"
    android:inputType="phone"
    android:hint="Enter Mobile Number"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent" />

<EditText
    android:id="@+id/editMessage"
    android:layout_width="0dp"
    android:layout_height="wrap_content"
    android:layout_marginStart="16dp"
    android:layout_marginTop="82dp"
    android:layout_marginEnd="16dp"
    android:layout_marginBottom="96dp"
    android:ems="10"
    android:inputType="textPersonName"
    android:hint="Enter Message"
    app:layout_constraintBottom_toTopOf="@+id/buttonSend"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/editPhoneNumber" />
</androidx.constraintlayout.widget.ConstraintLayout>

```

**Result:**

9:27   

9908055084

Hi, Hello

SEND

9:28   

 Search conversations 



(990) 805-5084  
You: Hi, Hello

4 min

 Start chat

## Experiment – 7

**AIM:** Create an android application using Fragments.

### MainActivity.java

```
package com.vvit.myapplication;

import android.app.Activity;
import android.app.Fragment;
import android.app.FragmentManager;
import android.app.FragmentTransaction;
import android.os.Bundle;
import android.view.View;

public class MainActivity extends Activity {

    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        final Fragment first = new FirstFragment();
        final Fragment second = new SecondFragment();
        findViewById(R.id.fragment1).setOnClickListener(new
View.OnClickListener() {
            @Override
            public void onClick(View v) {
                FragmentManager fm = getFragmentManager();
                FragmentTransaction fragmentTransaction =
fm.beginTransaction();
                fragmentTransaction.replace(R.id.layout, first);
                fragmentTransaction.commit();
            }
        });
        findViewById(R.id.fragment2).setOnClickListener(new
View.OnClickListener() {
            @Override
            public void onClick(View v) {
                FragmentManager fm = getFragmentManager();
                FragmentTransaction fragmentTransaction =
fm.beginTransaction();
                fragmentTransaction.replace(R.id.layout, second);
                fragmentTransaction.commit();
            }
        });
    }
}
```

### activity\_main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout 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:orientation="vertical"
    tools:context=".MainActivity"
    android:padding="8dp"
    android:layout_margin="8dp">

    <Button
        android:id="@+id/fragment1"
        android:layout_width="wrap_content"
```



```

        android:layout_height="wrap_content"
        android:layout_alignParentTop="true"
        android:layout_centerHorizontal="true"
        android:layout_gravity="center"
        android:layout_marginTop="27dp"
        android:background="@drawable/roundbutton"
        android:text="fragment1" />

<Button
    android:id="@+id/fragment2"
    android:layout_width="wrap_content"
    android:layout_gravity="center"
    android:layout_height="wrap_content"
    android:layout_alignParentTop="true"
    android:layout_centerHorizontal="true"
    android:layout_marginTop="27dp"
    android:background="@drawable/roundbutton"
    android:text="fragment2" />

<LinearLayout
    android:id="@+id/layout"
    android:layout_marginTop="20dp"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"></LinearLayout>
</LinearLayout>

```

## FirstFragment.java

```

package com.vvit.myapplication;

import android.app.Fragment;
import android.os.Bundle;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.TextView;

public class FirstFragment extends Fragment {

    private TextView textView;

    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
    }

    @Override
    public View onCreateView(LayoutInflater inflater, ViewGroup container,
                             Bundle savedInstanceState) {
        // Inflate the layout for this fragment
        View view = inflater.inflate(R.layout.fragment_first, container,
false);
        textView = view.findViewById(R.id.fragment_view1);
        textView.setText("First Fragment");
        return view;
    }
}

```

### fragment\_first.xml

```
<?xml version="1.0" encoding="utf-8"?>
<FrameLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="#dcdcdc"
    tools:context=".FirstFragment">

    <TextView
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:id="@+id/fragment_view1"
        android:layout_gravity="center"
        android:gravity="center"
        android:textColor="@color/purple_700"
        android:text="First Fragment" />

</FrameLayout>
```

### SecondFragment.java

```
package com.vvit.myapplication;

import android.app.Fragment;
import android.os.Bundle;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.TextView;

public class SecondFragment extends Fragment {

    private TextView textView;

    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
    }

    @Override
    public View onCreateView(LayoutInflater inflater, ViewGroup container,
        Bundle savedInstanceState) {
        // Inflate the layout for this fragment
        View view = inflater.inflate(R.layout.fragment_second, container,
false);

        textView = view.findViewById(R.id.fragment_view2);
        textView.setText("Second Fragment");
        return view;
    }
}
```

### fragment\_second.xml

```
<?xml version="1.0" encoding="utf-8"?>
<FrameLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="#FFFD04"
    tools:context=".SecondFragment">
```

```

<TextView
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:id="@+id/fragment_view2"
    android:textColor="@color/teal_700"
    android:layout_gravity="center"
    android:gravity="center"
    android:text="Second Fragment" />

</FrameLayout>

```

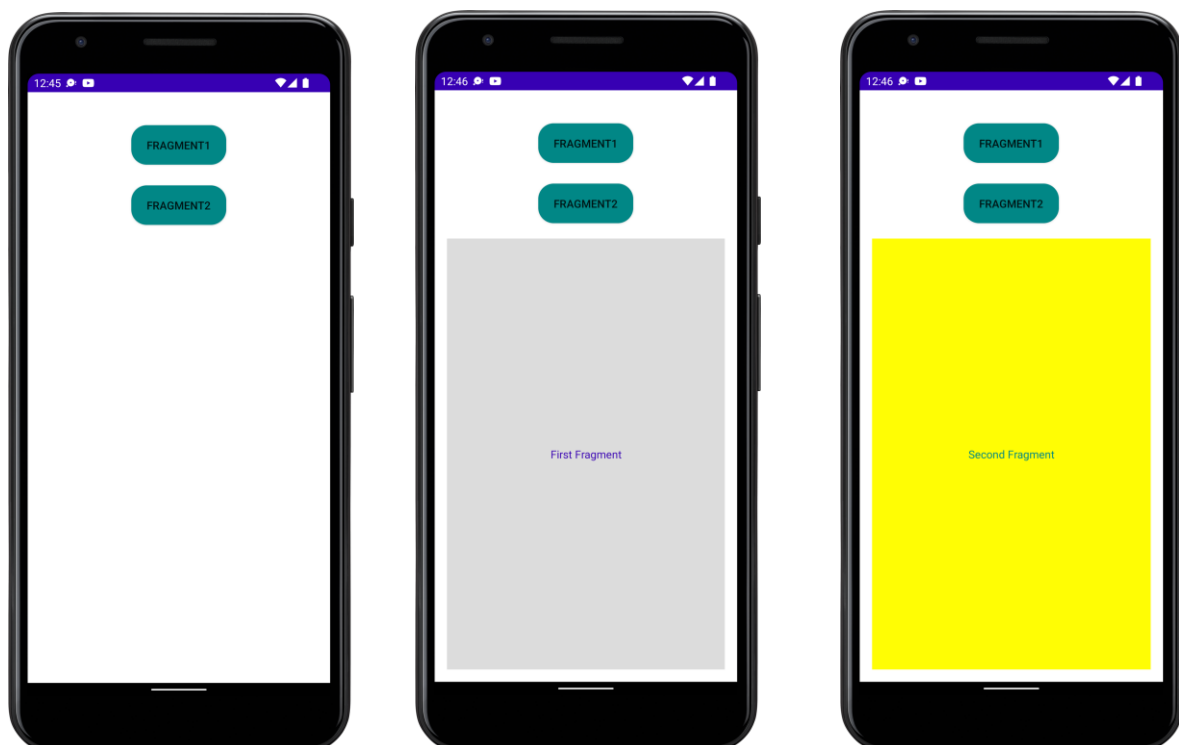
### roundbutton.xml

```

<?xml version="1.0" encoding="utf-8"?>
<shape xmlns:android="http://schemas.android.com/apk/res/android">
    <padding android:top="16dp" android:right="20dp" android:left="20dp"
    android:bottom="16dp"/>
    <solid android:color="@color/teal_700" />
    <corners android:radius="20dp"/>
</shape>

```

### Result



## Experiment – 8

**AIM:** Design an android application using menus.

### MainActivity.java

```
package com.vvit.myapplication;

import android.app.Activity;
import android.app.Fragment;
import android.app.FragmentManager;
import android.app.FragmentTransaction;
import android.os.Bundle;
import android.util.Log;
import android.view.Menu;
import android.view.MenuInflater;
import android.view.MenuItem;
import android.view.View;
import android.widget.Toast;
import androidx.appcompat.app.AppCompatActivity;
import androidx.appcompat.widget.Toolbar;

public class MainActivity extends AppCompatActivity {

    private static final String TAG = "mytag";

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

    @Override
    public boolean onCreateOptionsMenu(Menu menu) {
        MenuInflater inflater = getMenuInflater();
        inflater.inflate(R.menu.main_menu, menu);
        return true;
    }

    @Override
    public boolean onOptionsItemSelected(MenuItem item) {
        switch (item.getItemId()) {
            case R.id.aboutMenu:
                Toast.makeText(this, "Clicked on About!",
                    Toast.LENGTH_SHORT).show();
                Log.d(TAG, "Clicked on About!");
                // Code for About goes here
                return true;
            case R.id.helpMenu:
                Toast.makeText(this, "Clicked on Help!",
                    Toast.LENGTH_SHORT).show();
                Log.d(TAG, "Clicked on Help!");
                // Code for Help goes here
                return true;
            case R.id.signOutMenu:
                Toast.makeText(this, "Clicked on Sign out!",
                    Toast.LENGTH_SHORT).show();
                Log.d(TAG, "User signed out");
        }
    }
}
```

```

        // SignOut method call goes here
        return true;
    default:
        return super.onOptionsItemSelected(item);
    }
}
}

```

## activity\_main.xml

```

<?xml version="1.0" encoding="utf-8"?>
<androidx.coordinatorlayout.widget.CoordinatorLayout
    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">

    <com.google.android.material.appbar.AppBarLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:theme="@style/Widget.Material13.PopupMenu"/>

</androidx.coordinatorlayout.widget.CoordinatorLayout>

```

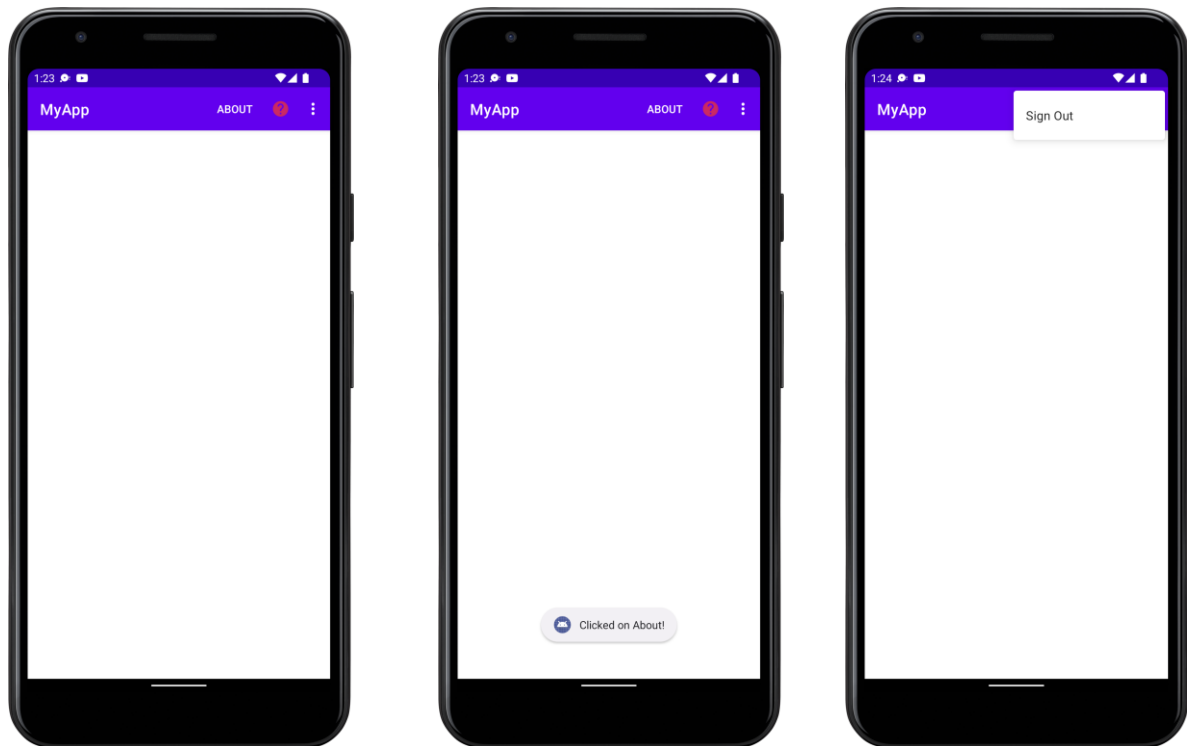
## main\_menu.xml

```

<?xml version="1.0" encoding="utf-8"?>
<menu xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto">
    <item
        android:id="@+id/aboutMenu"
        app:showAsAction="always"
        android:title="About"/>
    <item android:id="@+id/helpMenu"
        app:showAsAction="ifRoom"
        android:icon="@drawable/icon_help"
        android:title="Help" />
    <item android:id="@+id/signOutMenu"
        android:title="Sign Out"
        android:icon="@drawable/icon_logout"/>
</menu>

```

## Result



## Experiment – 9

**AIM:** Design an android application to display a list of fruits using list view.

### MainActivity.java

```
package com.vvit.myapplication;

import android.os.Bundle;
import android.view.View;
import android.widget.AdapterView;
import android.widget.AdapterView.OnItemClickListener;
import android.widget.ArrayAdapter;
import android.widget.ListView;
import android.widget.Toast;

import androidx.appcompat.app.AppCompatActivity;

public class MainActivity extends AppCompatActivity {
    private String myList[] = {"Apple", "Banana", "Cherry", "Dragon Fruit",
"Kiwi", "Grapes", "Jack Fruit", "Lemon", "Mango", "Orange", "Pomegranate",
"Papaya",
"Custard Apple", "Pine Apple", "Guava"};
    ListView myListView;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        myListView = findViewById(R.id.list_view);

        ArrayAdapter<String> myAdapter = new
ArrayAdapter<String>(this, R.layout.row_fruit, R.id._fruit_name, myList);
        myListView.setAdapter(myAdapter);

        myListView.setOnItemClickListener(new
AdapterView.OnItemClickListener() {
            @Override
            public void onItemClick(AdapterView<?> adapterView, View view,
int i, long l) {
                String value = myAdapter.getItem(i);
                Toast.makeText(MainActivity.this, "Clicked on "+ value ,
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">
```

```

<ListView
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:layout_marginStart="16dp"
    android:layout_marginTop="16dp"
    android:layout_marginEnd="16dp"
    android:layout_marginBottom="16dp"
    android:id="@+id/list_view"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent" />
</androidx.constraintlayout.widget.ConstraintLayout>

```

## row\_fruit.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="wrap_content">

    <TextView
        android:id="@+id/_fruit_name"
        android:textSize="24sp"
        android:layout_margin="6dp"
        android:padding="16dp"
        android:textColor="#1A4EB6"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="TextView"
        android:background="@drawable/borderview"
    />

</LinearLayout>

```

## borderview.xml

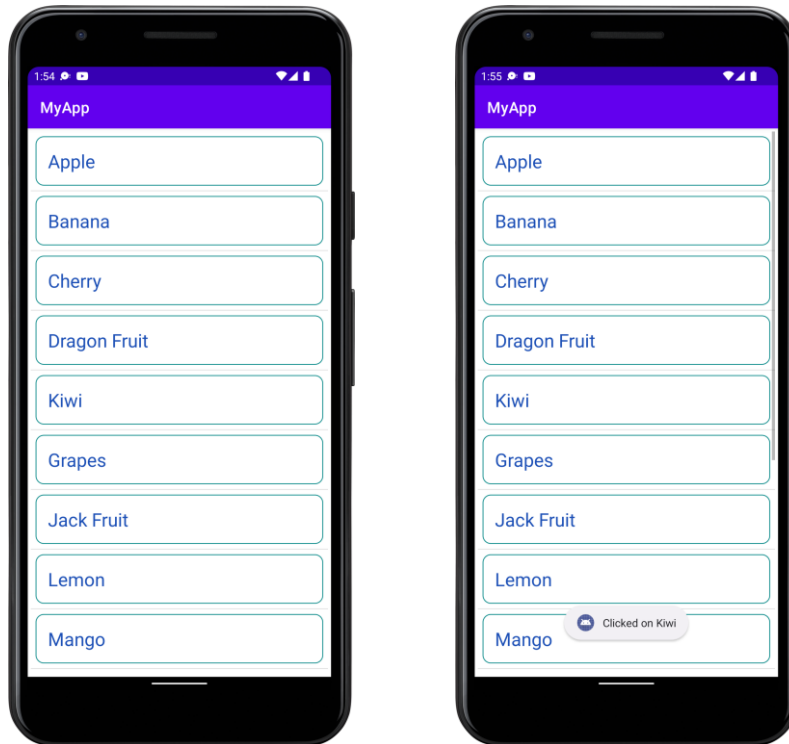
```

<?xml version="1.0" encoding="utf-8"?>
<shape xmlns:android="http://schemas.android.com/apk/res/android">
    <padding android:top="16dp" android:right="2dp" android:left="2dp"
        android:bottom="16dp"/>
    <stroke android:color="@color/teal_700" android:width="1dp" />
    <corners android:radius="10dp"/>
</shape>

```

## Result





## Experiment – 10

**AIM:** Design an android application to display a list of users using recycler view from internet.

### android.manifest.xml

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.vvit.myapplication">
    <uses-permission android:name="android.permission.INTERNET" />
    <uses-permission android:name="android.permission.ACCESS_NETWORK_STATE" />
    <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.MyApp">
        <activity
            android:name=".UserDetailActivity"
            android:exported="false"
            />
        <activity
            android:name=".UserActivity"
            android:exported="false"
            android:label="Home"/>
        <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>
        </activity>
    </application>
</manifest>
```

### MainActivity.java

```
package com.vvit.myapplication;
import android.os.Bundle;
import androidx.appcompat.app.AppCompatActivity;
public class MainActivity extends AppCompatActivity {
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        new UserAsyncAdapter(this).execute();
    }
}
```

### 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">
```

```

        <TextView
            android:layout_width="match_parent"
            android:layout_height="match_parent"
            android:text="Downloading Content"
            android:gravity="center"/>
    </androidx.constraintlayout.widget.ConstraintLayout>

```

#### row\_users.xml

```

<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:background="@drawable/borderview"
    android:orientation="vertical"
    android:layout_margin="8dp"
    android:id="@+id/_row_users_layout">
    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:orientation="vertical"
        android:padding="16dp">
        <LinearLayout
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:orientation="horizontal">
            <ImageView
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:src="@drawable/icon_user"
                android:paddingRight="16dp"/>
            <TextView
                android:id="@+id/_user_name"
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:textAllCaps="true"
                android:textStyle="bold"
                android:paddingBottom="8dp"/>
        </LinearLayout>
        <LinearLayout
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:orientation="horizontal">
            <ImageView
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:src="@drawable/icon_email"
                android:paddingRight="16dp"/>
            <TextView
                android:id="@+id/_user_email"
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                />
        </LinearLayout>
    </LinearLayout>
</LinearLayout>

```

#### UserAsyncAdapter.java

```

package com.vvit.myapplication;

import android.app.Activity;

```

```

import android.content.Intent;
import android.os.AsyncTask;

import org.json.JSONArray;
import org.json.JSONException;
import org.json.JSONObject;
import java.io.IOException;
import java.util.ArrayList;

public class UserAsyncAdapter extends AsyncTask<Void, Void, String> {
    private ArrayList<User> myUsers;
    private Activity activity;

    public UserAsyncAdapter(Activity activity) {
        this.activity = activity;
    }
    @Override
    protected String doInBackground(Void... voids) {
        String result = null;
        try {
            result = NetworkUtils.getResponseFromHttpUrl();
        } catch (IOException e) {
            e.printStackTrace();
        }
        return result;
    }
    @Override
    protected void onPostExecute(String s) {
        super.onPostExecute(s);
        myUsers = new ArrayList<>();
        try {
            JSONArray userArray = new JSONArray(s);
            for(int i=0;i<userArray.length(); i++){
                JSONObject user = userArray.getJSONObject(i);
                User myUser = new User(i+1);
                myUser.setName(user.getString("name"));
                myUser.setUsername(user.getString("username"));
                myUser.setEmail(user.getString("email"));
                myUser.setPhone(user.getString("phone"));
                myUsers.add(myUser);

                Intent intent = new Intent(activity, UserActivity.class);
                intent.putExtra("myList", myUsers);
                activity.startActivity(intent);
                activity.finish();
            }
        } catch (JSONException e) {
            e.printStackTrace();
        }
    }
}

```

### User.java

```

package com.vvit.myapplication;

import android.os.Parcel;
import android.os.Parcelable;

public class User implements Parcelable {
    private int id;
    private String name;
    private String username;

```

```

private String email;
private String phone;

public User(int id) {
    this.id = id;
}

protected User(Parcel in) {
    id = in.readInt();
    name = in.readString();
    username = in.readString();
    email = in.readString();
    phone = in.readString();
}

public static final Creator<User> CREATOR = new Creator<User>() {
    @Override
    public User createFromParcel(Parcel in) {
        return new User(in);
    }

    @Override
    public User[] newArray(int size) {
        return new User[size];
    }
};

public int getId() {
    return id;
}

public void setId(int id) {
    this.id = id;
}

public String getName() {
    return name;
}

public void setName(String name) {
    this.name = name;
}

public String getUsername() {
    return username;
}

public void setUsername(String username) {
    this.username = username;
}

public String getEmail() {
    return email;
}

public void setEmail(String email) {
    this.email = email;
}

public String getPhone() {
    return phone;
}

```

```

    }

    public void setPhone(String phone) {
        this.phone = phone;
    }

    @Override
    public String toString() {
        return "User{" +
            "id=" + id +
            ", name='" + name + '\'' +
            ", username='" + username + '\'' +
            ", email='" + email + '\'' +
            ", phone='" + phone + '\'' +
            '}';
    }

    @Override
    public int describeContents() {
        return 0;
    }

    @Override
    public void writeToParcel(Parcel parcel, int i) {
        parcel.writeInt(id);
        parcel.writeString(name);
        parcel.writeString(username);
        parcel.writeString(email);
        parcel.writeString(phone);
    }
}

```

### UserRecyclerAdapter.java

```

package com.vvit.myapp;

import android.app.Activity;
import android.content.Intent;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.LinearLayout;
import android.widget.TextView;
import androidx.annotation.NonNull;
import androidx.recyclerview.widget.RecyclerView;
import java.util.ArrayList;

public class UserRecyclerAdapter extends
RecyclerView.Adapter<UserRecyclerAdapter.UserViewHolder> {
    private Activity activity;
    private ArrayList<User> myUsers;

    public UserRecyclerAdapter(Activity activity, ArrayList<User> myUsers)
    {
        this.activity = activity;
        this.myUsers = myUsers;
    }

    @NonNull
    @Override
    public UserViewHolder onCreateViewHolder(@NonNull ViewGroup parent, int
viewType) {

```

```

        View view =
LayoutInflater.from(parent.getContext()).inflate(R.layout.row_users,
parent,false);
        ViewHolder holder = new ViewHolder(view);
        return holder;
    }

    @Override
    public void onBindViewHolder(@NonNull ViewHolder holder, int
position) {
        User user = myUsers.get(position);
        holder.user_name.setText(user.getName());
        holder.user_email.setText(user.getEmail());
        holder.layout.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                Intent intent = new Intent(activity,
UserDetailActivity.class);
                intent.putExtra("user",user);
                activity.startActivity(intent);
            }
        });
    }

    @Override
    public int getItemCount() {
        return myUsers.size();
    }

    public class ViewHolder extends RecyclerView.ViewHolder {
        TextView user_name, user_email;
        LinearLayout layout;
        public ViewHolder(@NonNull View itemView) {
            super(itemView);
            user_name = itemView.findViewById(R.id._user_name);
            user_email = itemView.findViewById(R.id._user_email);
            layout = itemView.findViewById(R.id._row_users_layout);
        }
    }
}

```

## UserActivity.java

```

package com.vvit.myapp;

import android.os.Bundle;

import androidx.appcompat.app.AppCompatActivity;
import androidx.recyclerview.widget.LinearLayoutManager;
import androidx.recyclerview.widget.RecyclerView;

import java.util.ArrayList;

public class UserActivity extends AppCompatActivity {
    private RecyclerView myUserListView;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_user);
    }
}

```

```

        myUserListView = findViewById(R.id.user_list_view);
        ArrayList<User> myUsers =
getIntent().getParcelableArrayListExtra("myList");

        myUserListView.setAdapter(new UserRecyclerViewAdapter(this, myUsers));
        myUserListView.setLayoutManager(new LinearLayoutManager(this));
    }
}

```

### activity\_user.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=".UserActivity">
    <androidx.recyclerview.widget.RecyclerView
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:layout_marginStart="16dp"
        android:layout_marginTop="16dp"
        android:layout_marginEnd="16dp"
        android:layout_marginBottom="16dp"
        android:id="@+id/user_list_view"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent" />

</androidx.constraintlayout.widget.ConstraintLayout>

```

### activity\_user\_detail.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=".UserDetailActivity"
android:padding="8dp">

    <androidx.cardview.widget.CardView
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:elevation="6dp"
        app:cardCornerRadius="6dp"
        android:background="@drawable/borderview"

        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent">

        <LinearLayout
            android:layout_width="match_parent"
            android:layout_height="match_parent"
            android:orientation="vertical">

            <ImageView

```



```

        android:layout_width="match_parent"
        android:layout_height="400dp"
        android:src="@drawable/icon_user"/>
    <TextView
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:id="@+id/_details_name"
        android:textSize="32sp"
    />
    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:orientation="horizontal">
        <ImageView
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:src="@drawable/icon_user"
            android:paddingRight="16dp"/>
        <TextView
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:id="@+id/_details_user_id"/>
    </LinearLayout>
    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:orientation="horizontal">
        <ImageView
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:src="@drawable/icon_email"
            android:paddingRight="16dp"/>
        <TextView
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:id="@+id/_details_email"/>
    </LinearLayout>
    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:orientation="horizontal">
        <ImageView
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:src="@drawable/icon_phone"
            android:paddingRight="16dp"/>
        <TextView
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:id="@+id/_details_phone"/>
    </LinearLayout>
</LinearLayout>
</androidx.cardview.widget.CardView>
</androidx.constraintlayout.widget.ConstraintLayout>

```

## UserDetailsActivity.java

```
package com.vvit.myapplication;

import androidx.appcompat.app.AppCompatActivity;
import android.os.Bundle;
import android.widget.TextView;

public class UserDetailsActivity extends AppCompatActivity {
    TextView userName, userId, userEmail, userPhone;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_user_detail);

        userName = findViewById(R.id._details_name);
        userId = findViewById(R.id._details_user_id);
        userEmail = findViewById(R.id._details_email);
        userPhone = findViewById(R.id._details_phone);

        User user = getIntent().getParcelableExtra("user");

        userName.setText(user.getUsername());
        userId.setText(user.getName());
        userPhone.setText(user.getPhone());
        userEmail.setText(user.getEmail());
        setTitle(new String(user.getUsername()));
    }
}
```

## Result:

