



DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

Discover. Learn. Empower.

Experiment-1.1

Student Name: Aafreen Khan
Branch: BE-CSE
Semester: 6th
Subject Name: MAD Lab

UID: 21BCS1397
Section/Group: 606-A
Date of Performance: 16/01/2024
Subject Code: 21 CSH-355

1. **Aim of the practical:** Installation and configuration of Android Studio.(CO1)
2. **Objective:-**-The objective of installing and configuring Android Studio is to set up a development environment for creating Android applications. The process involves installing the necessary tools and components, configuring the development environment, and preparing the IDE for efficient Android app development.

3. Input/Apparatus Used:

Input:

Computer: Android Studio is compatible with Windows, macOS, and Linux. Ensure that your computer meets the minimum system requirements for the chosen operating system.

Internet Connection: A reliable internet connection is required to download Android Studio and the necessary SDK components during the installation process.

Apparatus:

Android Studio Installer: Download the Android Studio installer from the official Android Studio website (<https://developer.android.com/studio>). Choose the appropriate version for your operating system.

Computer Mouse,Keyboard and Monitor/Display: Use a mouse and keyboard to interact with the installation process, configure settings, and



DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

Discover. Learn. Empower.

navigate through Android Studio. Android Studio requires a monitor or display to visualize the installation process and subsequently to develop and test Android applications.

Storage Space: Ensure sufficient free storage space on your computer to accommodate the Android Studio installation and any additional SDK components you may download.

4. Procedure:

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 JDK5 or later version

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 2.2 Download](#), If you are new to installing Android Studio on windows,you will find a file, which is named as android-studio-bundle-143.3101438-windows.exe. So just download and run on windows machine according to android studio wizard guideline.

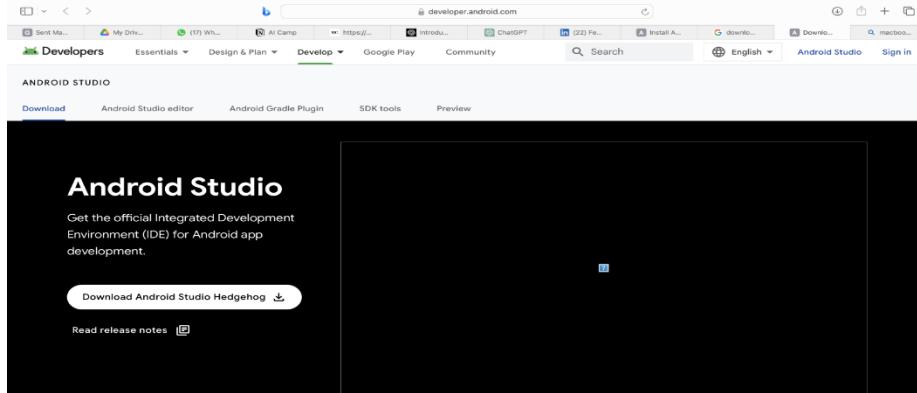
If you are installing Android Studio on Mac or Linux, You can download the latest version from [Android Studio Mac Download](#) or [Android Studio Linux Download](#), check the instructions provided along with the



DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

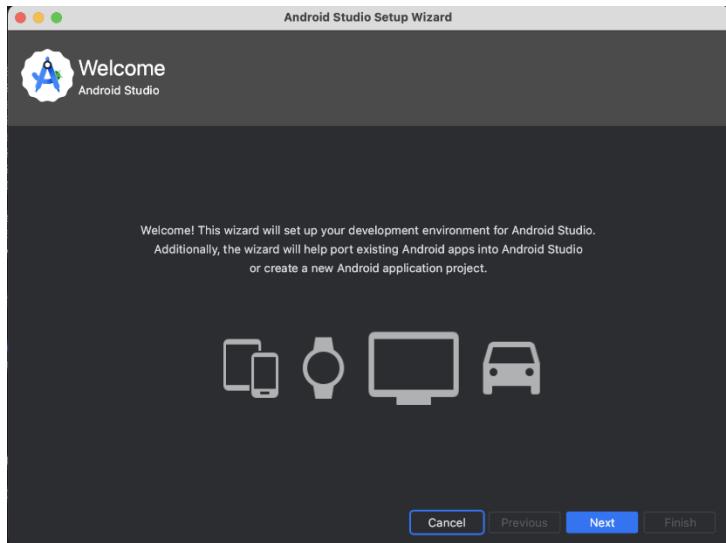
Discover. Learn. Empower.

downloaded file for Mac OS and Linux.



New features

- **Step 3:** It would extract SDK packages into our local machine, it would take a while to finish the task and would take 2626 MB of Hard disk space.

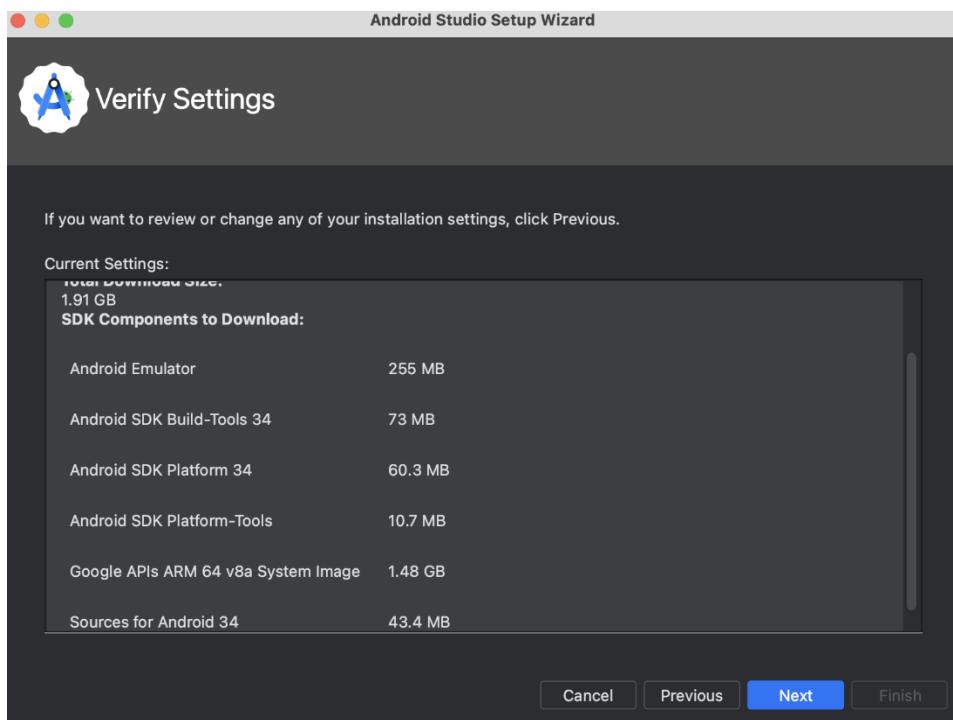
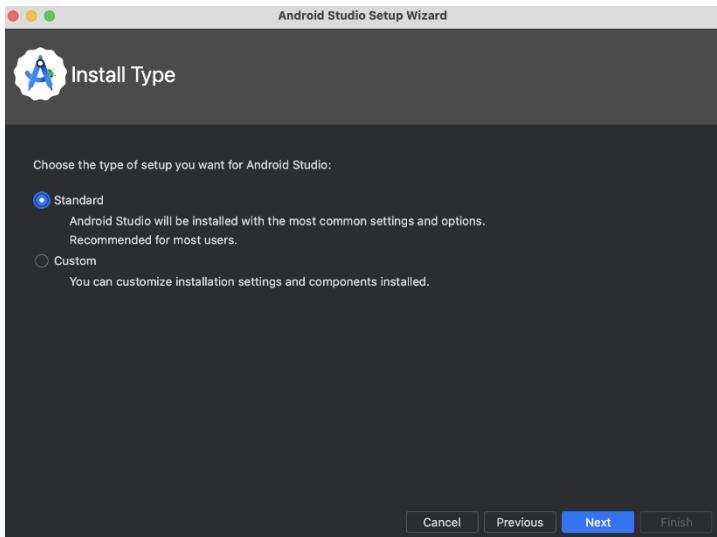




DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

Discover. Learn. Empower.

- **Step 4:** After done all above steps perfectly, you must get finish button and it can be open android studio project with Welcome to android studio message as shown below.

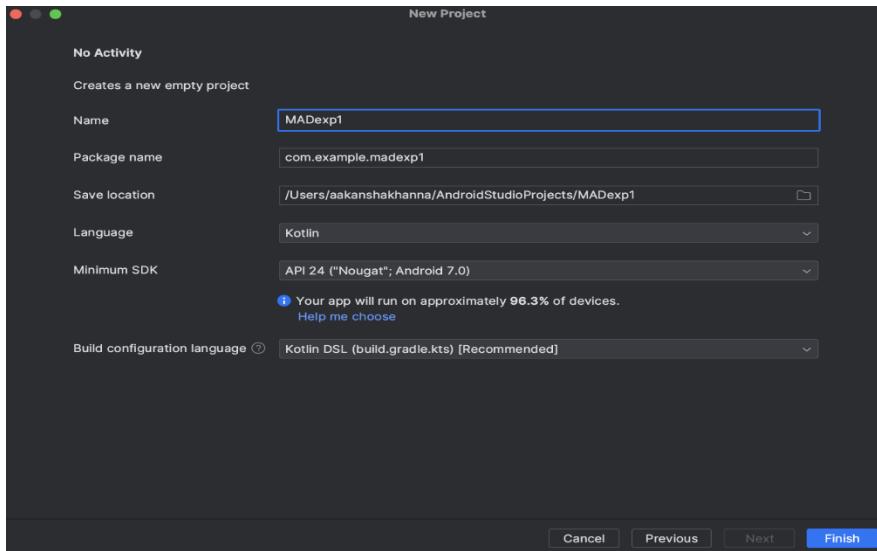




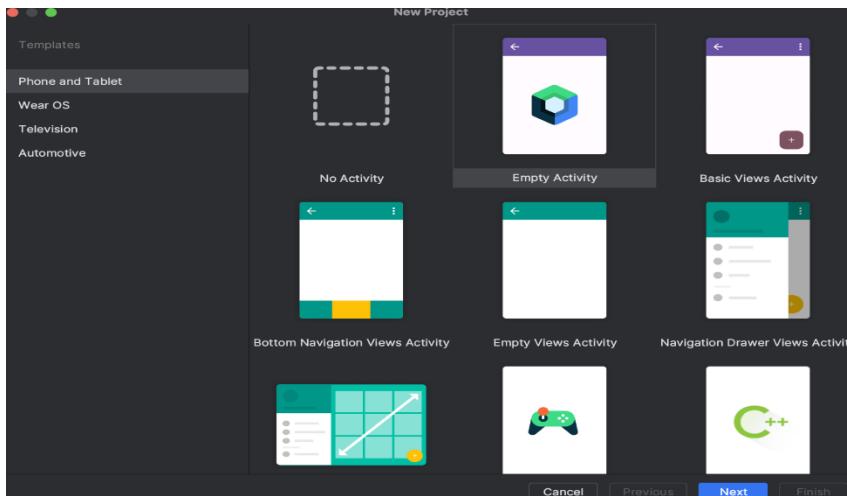
DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

Discover. Learn. Empower.

- **Step 5:** You can start your application development by calling start a new android studio project. in a new installation frame should ask Application name, package information and location of the project.



- **Step 6:** The next level of installation should contain selecting the activity to mobile, it specifies the default layout for Applications



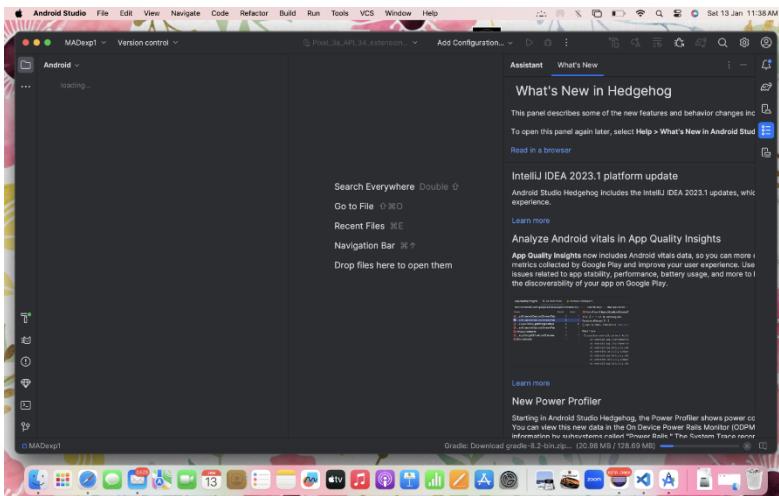
- **Step 7:** At the final stage it is going to be an open development tool to write the application code. If your AVD is created successfully it means your environment is ready for Android application development. If you like, you can close this window using the top-right cross button. Better you re-start



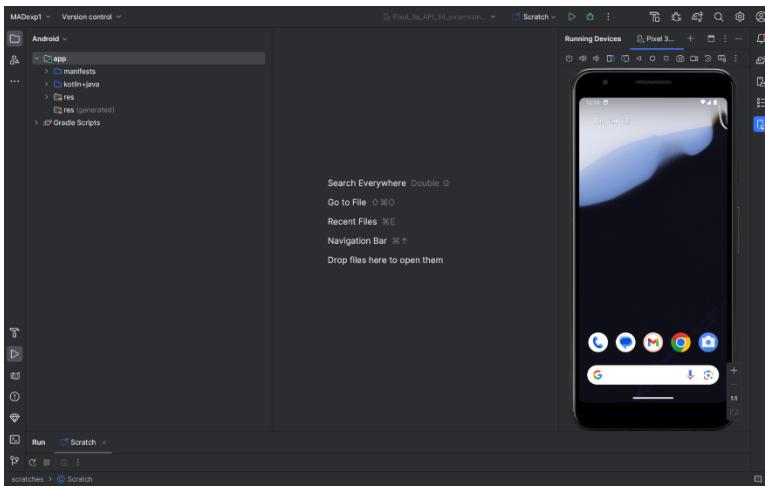
DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

Discover. Learn. Empower.

your machine and once you are done with this last step, you are ready to proceed for your first Android example



5. Result: Android Studio is successfully installed.



6. Learning Outcomes:

- Successful setup of your Android development environment.
- Project Workspace.
- Configuration Wizard Completion.
- SDK Components Downloaded.
- Android Studio IDE Launch.
- Error-Free Launch.



DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

Discover. Learn. Empower.

Experiment No: 1.2

Student Name: Aafreen Khan
Branch: CSE
Semester: 6th
Subject Name: MADLAB
Subject Code: 21CSH-355

UID: 21BCS1397
Section/Group: CC-606 A
Date of Performance: 23/1/24

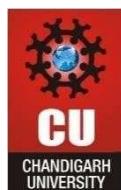
- **Aim:** To design an android application to display Hello World.
- **Objective:** Setting up Android Studio and configuring a basic Android project. Compiling and running a simple Android application on an emulator or a physical device.
- **Input/Apparatus Used:**

Input:-

- Computer: Android Studio is compatible with Windows, macOS, and Linux. Ensure that your computer meets the minimum system requirements for the chosen operating system.
- Internet Connection: A reliable internet connection is required to download Android Studio and the necessary SDK components during the installation process.

Apparatus:-

- Android Studio Installer: Download the Android Studio installer from the official Android Studio website (<https://developer.android.com/studio>). Choose the appropriate version for your operating system.
- Computer Mouse, Keyboard and Monitor/Display: Use a mouse and keyboard to interact with the installation process, configure settings, and navigate through Android Studio. Android Studio requires a monitor or display to visualize the installation process and subsequently to develop and test Android applications.



DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

Discover. Learn. Empower.

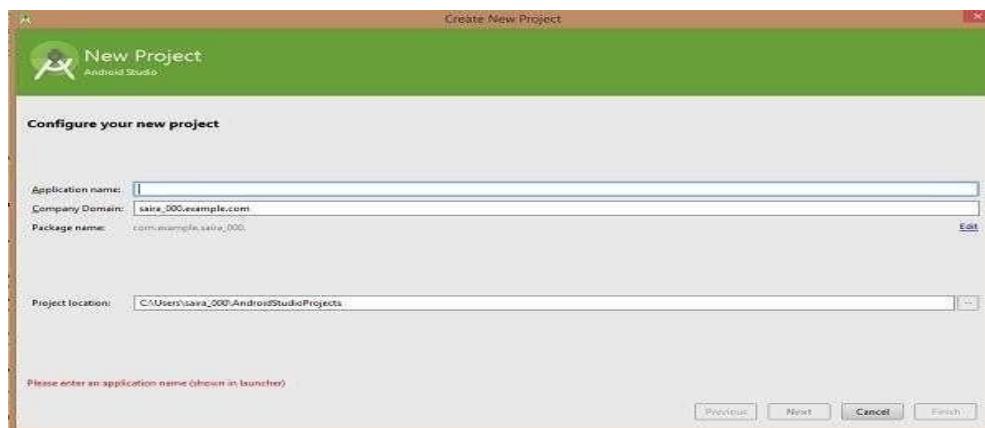
- Storage Space: Ensure sufficient free storage space on your computer to accommodate the Android Studio installation and any additional SDK components you may download.

- **Procedure/Algorithm/Pseudocode:**

- First step is to create a simple Android Application using Android studio. When you click on Android studio icon, it will show screen as shown below



- You can start your application development by calling start a new android studio project. in a new installation frame should ask Application name, package information and location of the project.—

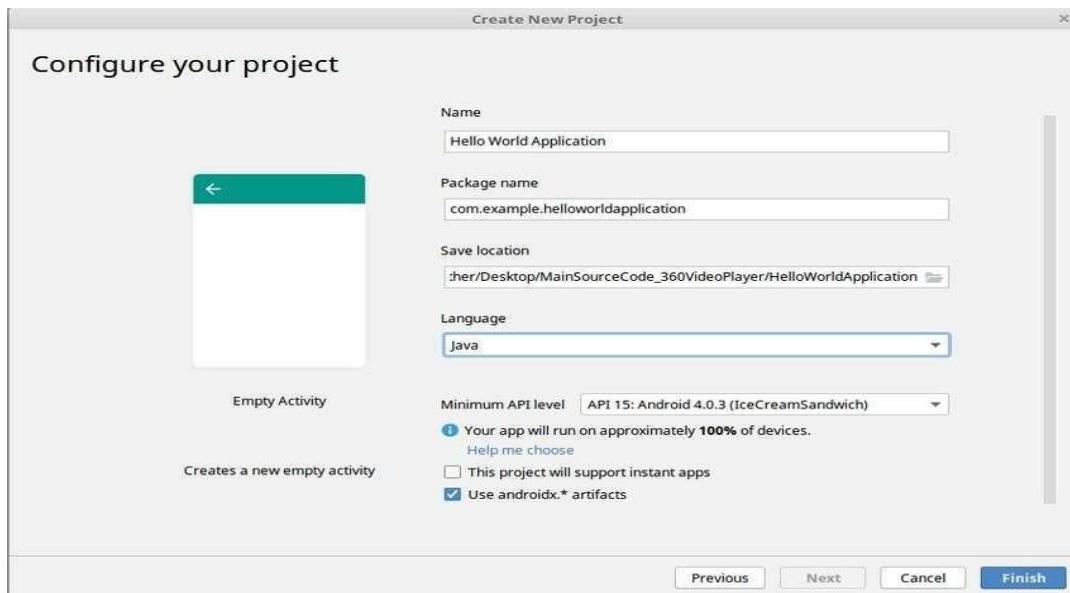




DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

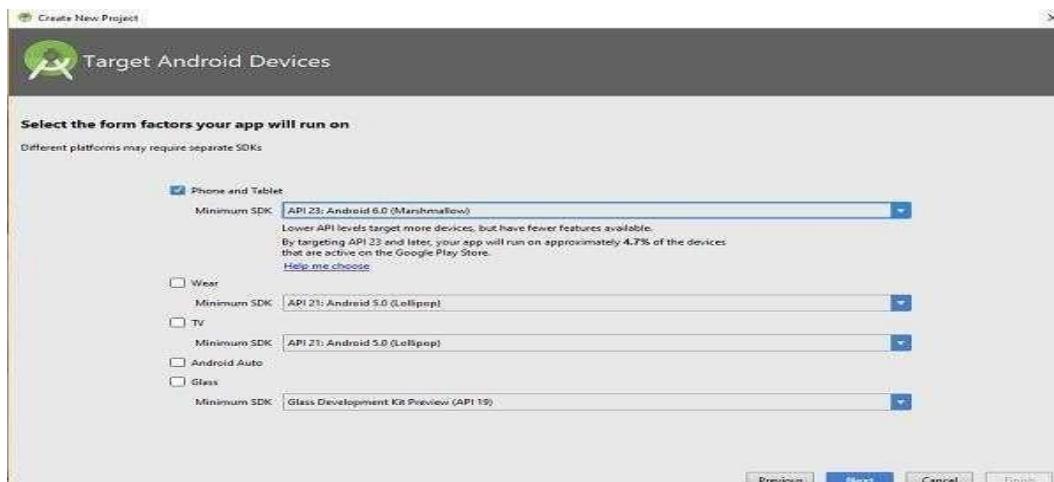
Discover. Learn. Empower.

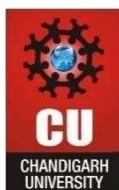
Configure the Hello World Project Details We'll finish creating the project by configuring some details about its name, location, and the API version it



Change the name of the application. Change the default Project location to your preferred directory or just leave it as the default location.

On the **minimum API level**, ensure that **API 15: Android 4.0.3 IceCreamSandwich** is set as the Minimum SDK. This ensures that your application runs on almost all devices.





DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

Discover. Learn. Empower.

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



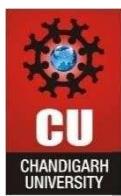
SOURCE CODE :

The Main Activity File

```
package com.example.helloworldapplication;
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);
    }
}
```



DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

Discover. Learn. Empower.

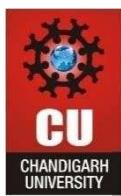
The Layout File

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 –

```
<RelativeLayout
    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" >

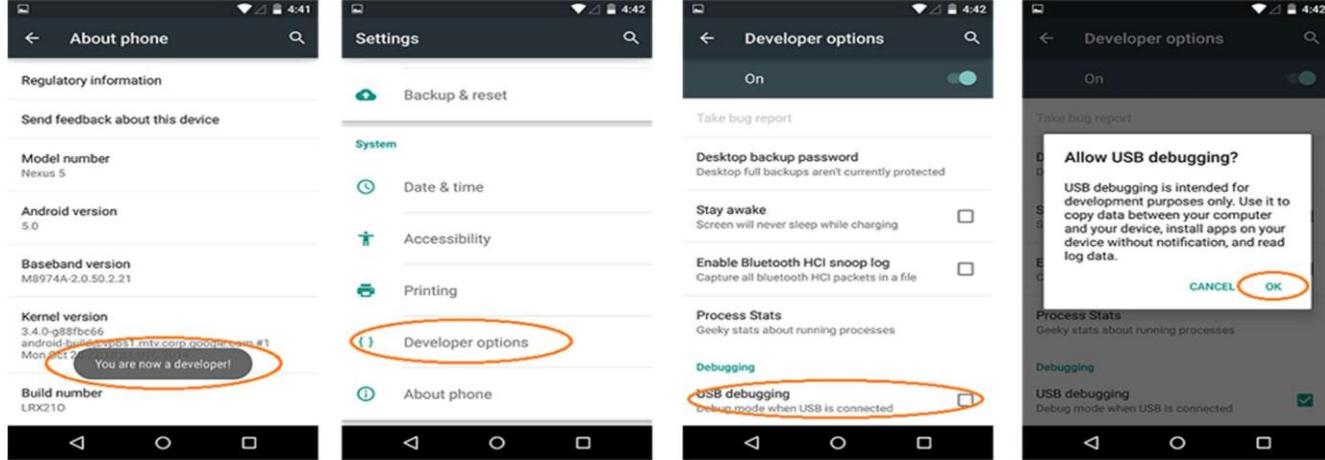
    <TextView
        android:layout_widt
        h="wrap_content"
        android:layout_heigh
        t="wrap_content"
        android:layout_cente
        rHorizontal="true"
        android:layout_cente
        rVertical="true"
        android:padding="@
        dimen/padding_medi
        um"
        android:text="@strin
        g/hello_world"
        tools:context=".Main
        Activity" />
```

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



DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

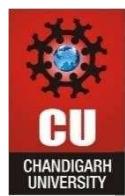
Discover. Learn. Empower.



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"

Output:-



DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

Discover. Learn. Empower.



Learning Outcomes:

- Understanding Android Architecture.
- Building own project environment
- Learning Java in practical manner



DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

Discover. Learn. Empower.

Experiment 1.3

Student Name: Aafreen Khan

UID: 21BCS1397

Branch: BE-CSE

Section/Group: CC-606-A

Semester: 6th

Date of Performance: 06-02-2024

Subject Name: Mobile Application Development

Subject Code: 21CSP-355

1. Aim:

Create Application by Using Widgets.

2. Objective:

To understand the concept of widgets and to implement the Widgets in Android Studio.

3. Procedure:

Step 1: Create a New Project

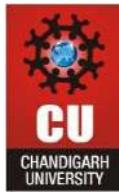
To create a new project in Android Studio please refer to How to Create/Start a New Project in Android Studio. We are implementing it for both Java and Kotlin languages.

Step 2: Add the App Widget to the Project

Right-Click on the app, move the cursor to new, find the “Widget” option at the end, select it.

Step 3: Install and Run the Code

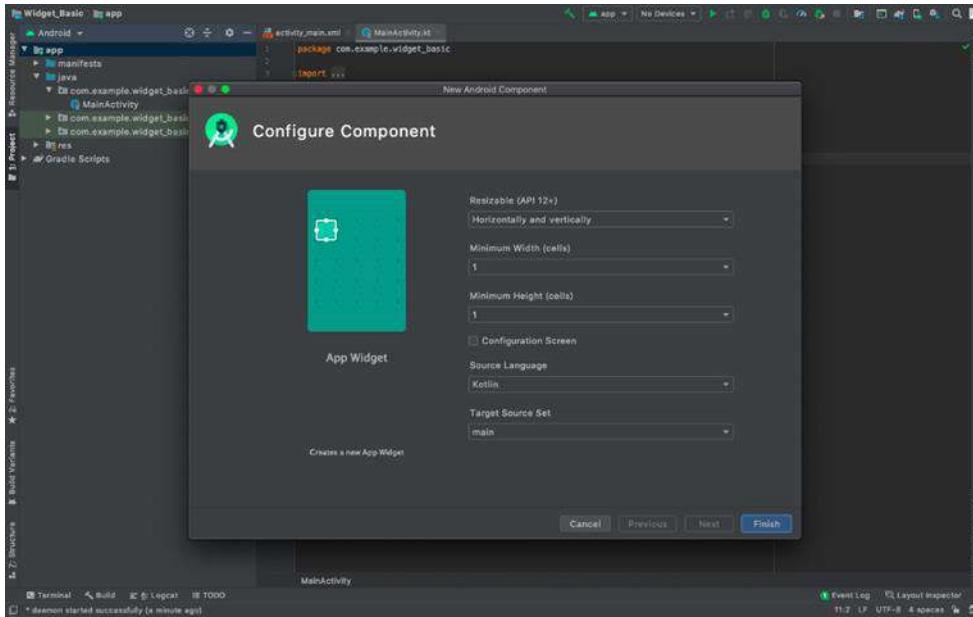
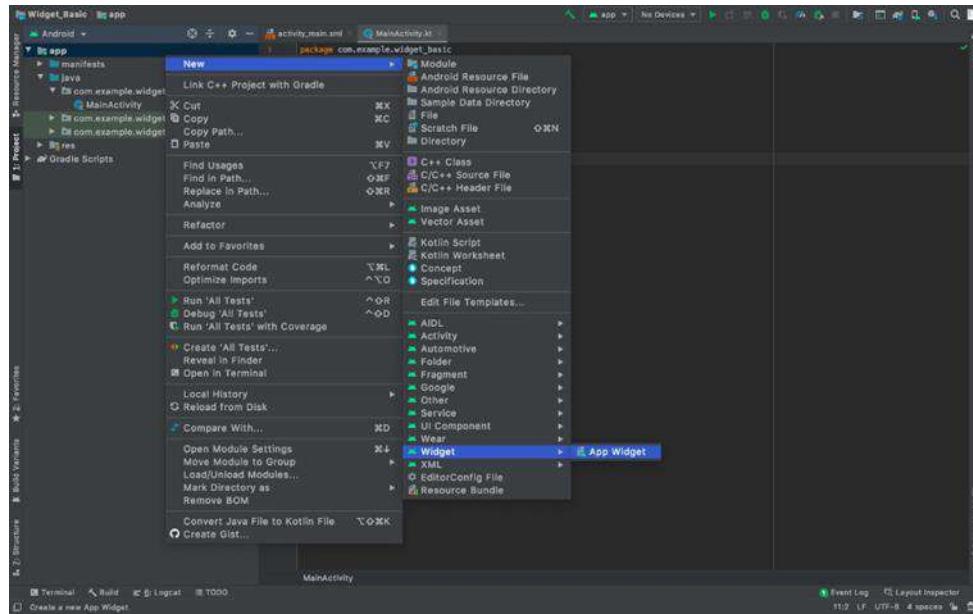
Install and run the code on Android Virtual Device (AVD) or a personal device. Open the widget section of the phone, lookup for a widget with the Application name, select it, bring it to the home screen. Try changing the dimensions and we are done! During this selecting and deploying process, a few extra files are generated and minor changes are made to existing files as well. No programming



DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

Discover. Learn. Empower.

is required for generating a basic widget and is only required if an application is to be embedded inside the widget, as discussed in the later parts of the article. Let us now explain the newly generated files the changes make to the existing ones, one by one.



Step 4: Write code in file



DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

Discover. Learn. Empower.

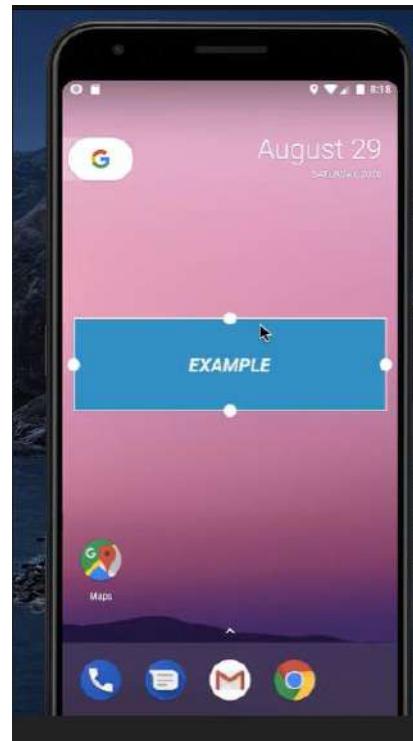
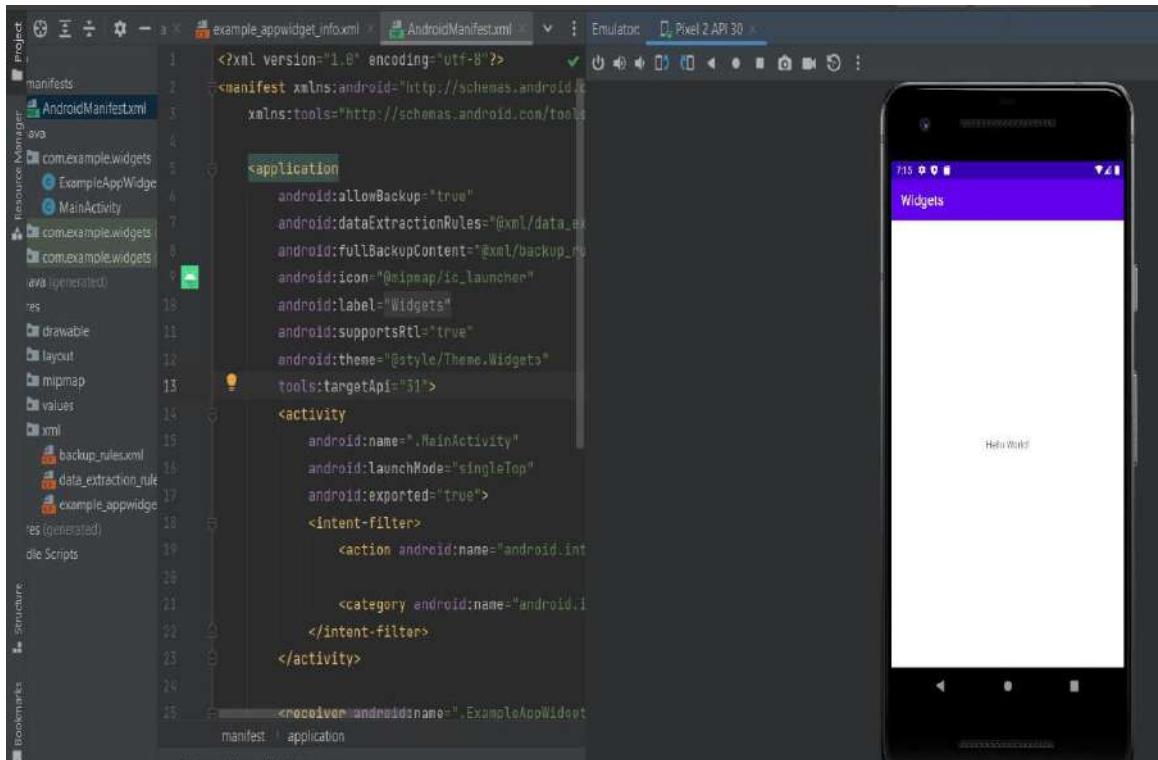
Make changes in newappwidget.java, new_app_widget.xml, dimens.xml and new_app_widget_info.xml.

```
for (int appWidgetId : appWidgetIds) updateAppWidget(context, appWidgetManager, appWidgetId);  
  
// Enter relevant functionality for  
// when the first widget is created  
// Usages:  
@Override public void onEnabled(Context context) { super.onEnabled(context); }  
  
// Enter relevant functionality for  
// when the last widget is disabled  
// Usages:  
@Override public void onDisabled(Context context) { super.onDisabled(context); }  
  
private void updateAppWidget(Context context, AppWidgetManager appWidgetManager, int appWidgetId)  
{  
    String widgetText = "EXAMPLE";  
  
    // Construct the RemoteViews object.  
    RemoteViews views = new RemoteViews(context.getPackageName(), R.layout.new_app_widget);  
    views.setTextViewText(R.id.appwidget_text, widgetText);  
  
    // Instruct the widget manager to update the widget.  
    appWidgetManager.updateAppWidget(appWidgetId, views);  
}
```

```
<RelativeLayout  
    xmlns:android="http://schemas.android.com/apk/res/android"  
    android:layout_width="match_parent"  
    android:layout_height="match_parent"  
    android:background="#80C"  
    android:padding="10px">  
  
    <TextView  
        android:id="@+id/appwidget_text"  
        android:layout_width="wrap_content"  
        android:layout_height="wrap_content"  
        android:layout_centerHorizontal="true"  
        android:layout_centerVertical="true"  
        android:margin="8dp"  
        android:background="#80C"  
        android:contentDescription="EXAMPLE"  
        android:text="EXAMPLE"  
        android:textColor="#fffff"  
        android:textSize="24sp"  
        android:textStyle="bolditalic" />  
  
</RelativeLayout>
```

```
<?xml version="1.0" encoding="utf-8"?>  
<resources>  
  
    <!--  
    Refer to App Widget Documentation for margin information  
    http://developer.android.com/guide/topics/appwidgets/index.html#CreatingLayout  
    -->  
    <dimen name="widget_margin">8dp</dimen>  
  
</resources>
```

4. Output:





DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

Discover. Learn. Empower.

Experiment 2.1

Name: Aafreen Khan

UID: 21BCS1397

Branch: BE-CSE

Section: 606-A

Semester: 6th

Date of Performance: 20-02-2024

Subject Name: MAD LAB

Subject Code: 21CSH-355

1. Aim:

Create an Android app that uses Intent with button to create a page and passes values from one activity to another.

2. Objective:

The objective of an Android app that uses Intent with a button to create a page and passes values from one activity to another could be to demonstrate and implement a simple data communication flow between different activities within an Android application. This type of app is commonly used to understand and showcase the concept of passing data between different screens or pages in Android.

3. Procedure:

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 we will need before start our Android application programming. Android Studio Java JDK 21.0.1 version

4. Code:

- **MainActivity.kt**

```
package com.example.intent
```

```
import android.content.Intent
import android.os.Bundle
```



DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

Discover. Learn. Empower.

```
import android.widget.Button
import androidx.activity.ComponentActivity
import androidx.activity.compose.setContent
import androidx.compose.ui.tooling.preview.Preview
import com.example.intent.ui.theme.IntentTheme

class MainActivity : ComponentActivity() {
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_main)

    }
}
```

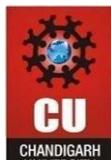
- **SecondActivity.kt**

```
package com.example.intent

import android.content.Intent
import androidx.appcompat.app.AppCompatActivity
class SecondActivity : AppCompatActivity() {
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_second)
        // Start the MainActivity
        startActivity(intent)
    }
}
```

- **activity_main.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">
    <Button
        android:id="@+id	btnNavigate"
        android:text="Navigate to Second Activity"
```



DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

CHANDIGARH
UNIVERSITY

Discover. Learn. Empower.

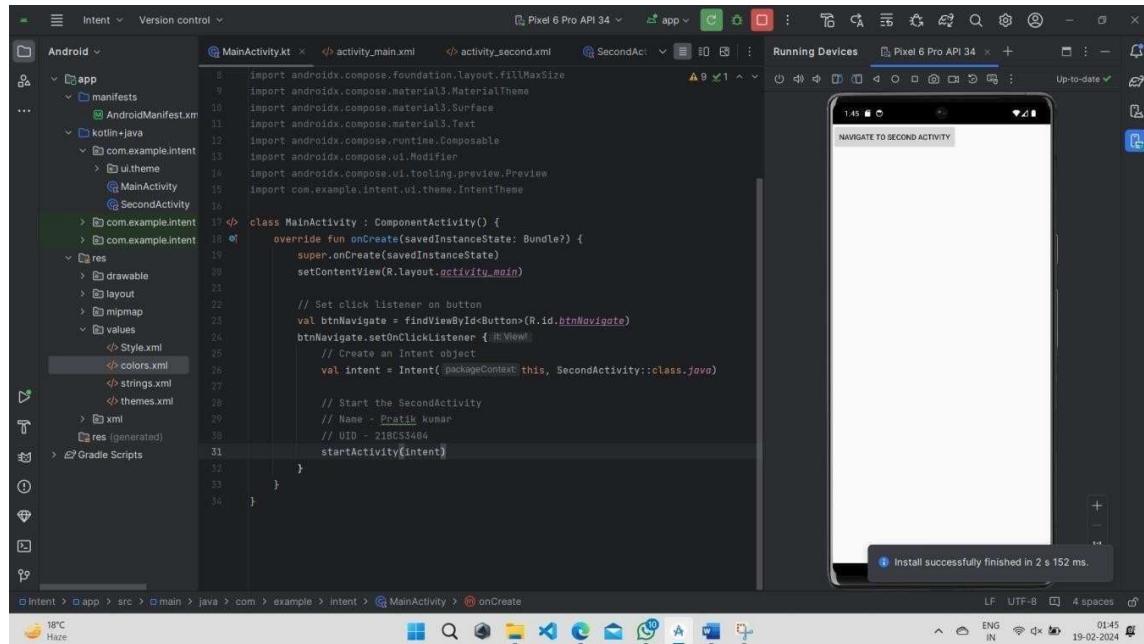
• activity_second.xml

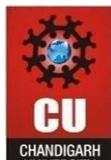
```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    tools:context=".SecondActivity">
    <Button
        android:id="@+id	btnBackToMain"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Back to Main Activity"
    />
</androidx.constraintlayout.widget.ConstraintLayout>
```

5. Output:

• Main Screen (MainActivity):

- When you open the app, you'll see the main screen.
- On the main screen, there is a button labeled "Navigate to Second Activity".
- Tap on this button to navigate to the second screen (SecondActivity).





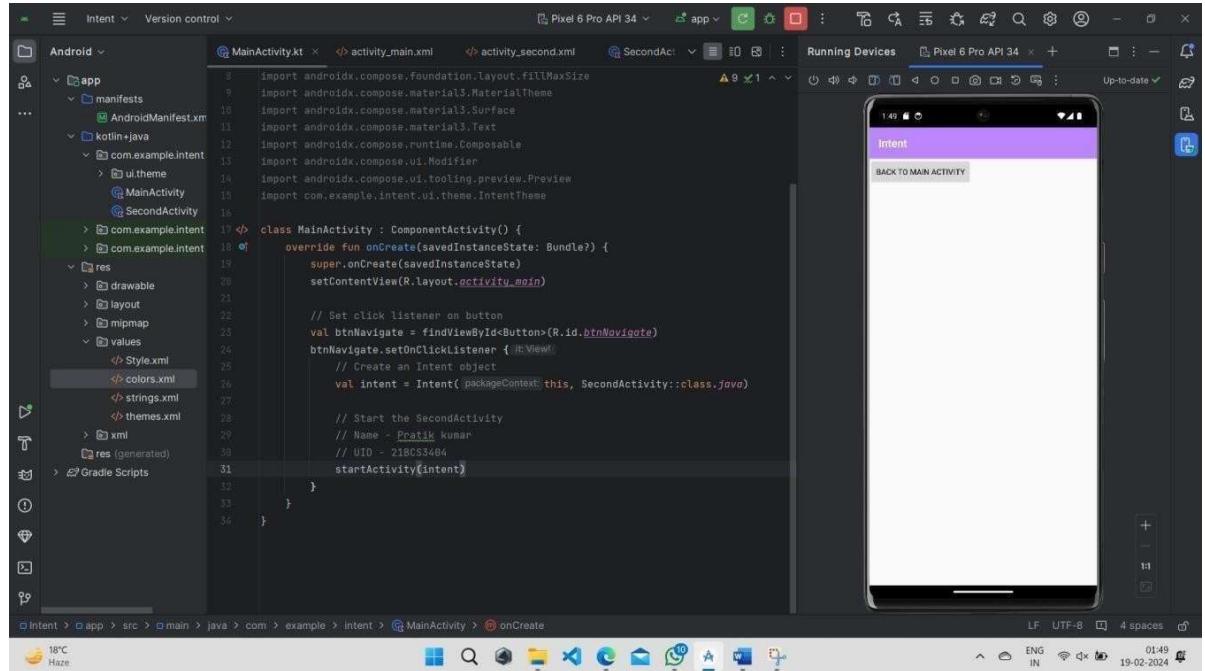
DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

CHANDIGARH
UNIVERSITY

Discover. Learn. Empower.

• Second Screen (SecondActivity):

- On the second screen, you'll see a button labeled "Back to Main Activity".
- Tap on this button to navigate back to the main screen (MainActivity).



6. Learning Outcome:

- We have successfully created the program.
- We have learned how to run the intent android studio.
- Now we can make intent application without any issue.



DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

Discover. Learn. Empower.

Experiment 2.2

Name- Aafreen Khan

UID:21BCS1397

Branch: BE-CSE

Section: 606 A

Semester: 6th

DateofPerformance:27-02-2024

Subject Name: MAD LAB

Subject Code: 21CSH-355

1. Aim:

Create an Android App using various controls such TexEdit, CheckBox, RadioButton, RadioGroup, etc.

The objective of an Android app using various controls such as EditText, CheckBox, RadioButton, and RadioGroup could be to create a user interface that involves user input, selection, and interaction with different types of controls. This type of app aims to showcase the usage and functionalities of these UI elements to enhance the user experience.

2. 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 we will need before start our Android application programming.

- Android Studio
- Java JDK 21.0.1 version

3. Code:

- **MainActivity.java**

```
package com.example.exp_22;

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);
    }
}
```



DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

Discover. Learn. Empower.

• 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="wrap_content"
        android:layout_height="wrap_content"
        android:text="SELECT ALL THAT APPLY"
        android:textColor="@color/teal_200"
        android:textSize="20sp"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        tools:ignore="MissingConstraints"
        tools:layout_editor_absoluteY="61dp">

    </TextView>

    <CheckBox
        android:id="@+id/simpleCheckBox"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_margin="200dp"
        android:text="OPTION 1"
        android:textSize="15sp"
        android:textColor="@color/purple_200"
        app:layout_constraintBottom_toTopOf="@+id/checkBox1"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.555"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        app:layout_constraintVertical_bias="0.353"
        tools:ignore="MissingConstraints" />

    <CheckBox
        android:id="@+id/checkBox1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginBottom="76dp"
        android:drawableLeft="@android:color/transparent"
```



DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

Discover. Learn. Empower.

```
        android:drawablePadding="10dp"
        android:text="OPTION 2"
        android:textSize="15sp"
        android:textColor="@color/purple_200"
        app:layout_constraintBottom_toTopOf="@+id/checkBox2"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.452"
        app:layout_constraintStart_toStartOf="parent"
        tools:ignore="DuplicateIds,MissingConstraints" />
```

```
<CheckBox
    android:id="@+id/checkBox2"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginBottom="76dp"
    android:drawableLeft="@android:color/transparent"
    android:drawablePadding="10dp"
    android:text="OPTION 3"
    android:textSize="15sp"
    android:textColor="@color/purple_200"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.452"
    app:layout_constraintStart_toStartOf="parent"
    tools:ignore="MissingConstraints" />
```

```
<CheckBox
    android:id="@+id/checkBox3"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginBottom="64dp"
    android:drawableLeft="@android:color/transparent"
    android:drawablePadding="10dp"
    android:text="OPTION 4"
    android:textSize="15sp"
    android:textColor="@color/purple_200"
    app:layout_constraintBottom_toTopOf="@+id/checkBox1"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.452"
    app:layout_constraintStart_toStartOf="parent"
    tools:ignore="MissingConstraints" />
```

```
<CheckBox
    android:id="@+id/checkBox4"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:drawableLeft="@android:color/transparent"
```

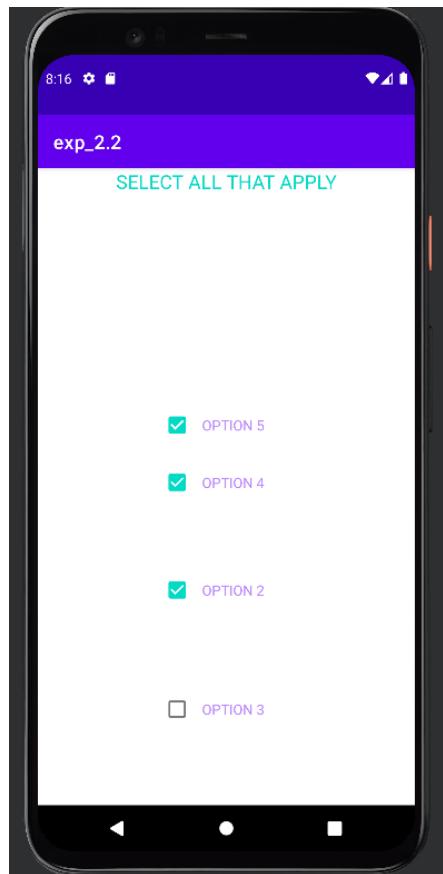


DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

Discover. Learn. Empower.

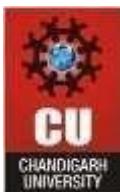
```
        android:drawablePadding="10dp"
        android:text="OPTION 5"
        android:textSize="15sp"
        android:textColor="@color/purple_200"
        app:layout_constraintBottom_toTopOf="@+id/checkBox3"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.452"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/simpleCheckBox"
        app:layout_constraintVertical_bias="0.376"
        tools:ignore="MissingConstraints" />
    </androidx.constraintlayout.widget.ConstraintLayout>
```

4. Output:



5. Learning Outcome:

- We have successfully created the program.
- We have learned how to execute various checkboxes, radio buttons, etc. over android studio.
- Learned to improve user interface/experience.



DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

Discover. Learn. Empower.

Experiment 2.3

Student Name: Aafreen Khan

UID: 21BCS1397

Branch: CSE

Section/Group: 606 A

Semester: 6th

Date of Performance: 12.4.24

Subject Name: MAD Lab

Subject Code: 21CSH-355

1. **Aim:** To design an android application Send SMS using Intent.

2. Objective: The objective of an Android-based application that uses Intent to send SMS can be to create a convenient and user-friendly tool for sending text messages. This type of app aims to leverage the Android platform's capabilities to provide a seamless and efficient way for users to compose and send SMS messages.

3. Input/Apparatus Used:

Input:

- **Computer:** Android Studio is compatible with Windows, macOS, and Linux. Ensure that your computer meets the minimum system requirements for the chosen operating system.
- **Internet Connection:** A reliable internet connection is required to download Android Studio and the necessary SDK components during the installation process.

Apparatus:

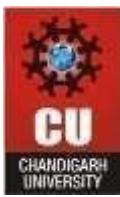
- **Android Studio Installer:** Download the Android Studio installer from the official Android Studio website (<https://developer.android.com/studio>). Choose the appropriate version for your operating system.
- **Computer Mouse, Keyboard and Monitor/Display:** Use a mouse and keyboard to interact with the installation process, configure settings, and navigate through Android Studio. Android Studio requires a monitor or display to visualize the installation process and subsequently to develop and test Android applications.
- **Storage Space:** Ensure sufficient free storage space on your computer to accommodate the Android Studio installation and any additional SDK components you may download.

4. Procedure:

Step1: creating XML File for designing as:

Activitymain.xml:

```
<TextView  
    android:id="@+id/textView"  
    android:layout_width="414dp"  
    android:layout_height="60dp"  
    android:background="#E91E63"  
    android:text="Welcome "  
    android:textAlignment="center"  
    android:textAppearance="@style/TextAppearance.AppCompat.Large"  
    android:textSize="48sp"  
    app:layout_constraintBottom_toBottomOf="parent"  
    app:layout_constraintEnd_toEndOf="parent"  
    app:layout_constraintHorizontal_bias="0.666"
```



DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

Discover. Learn. Empower.

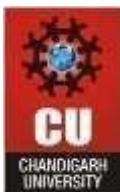
```
app:layout_constraintStart_toStartOf="parent"
app:layout_constraintTop_toTopOf="parent"
app:layout_constraintVertical_bias="0.0" />

<TextView
    android:id="@+id/textView2"
    android:layout_width="191dp"
    android:layout_height="43dp"
    android:text="Phone No :"
    android:textAppearance="@style/TextAppearance.AppCompat.Large"
    android:textSize="28sp"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.072"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/textView"
    app:layout_constraintVertical_bias="0.042" />

<EditText
    android:layout_width="249dp"
    android:layout_height="51dp"
    android:id="@+id/editText"
    android:hint="9876xxxxxx"
    android:textSize="24sp"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.993"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="@+id/textView"
    app:layout_constraintVertical_bias="0.13" />

<ImageView
    android:id="@+id/imageView3"
    android:layout_width="378dp"
    android:layout_height="519dp"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.969"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/textView"
    app:layout_constraintVertical_bias="0.453"
    app:srcCompat="@drawable/sms" />

<TextView
    android:id="@+id/textView3"
    android:layout_width="165dp"
    android:layout_height="37dp"
    android:text="Message:"
    android:textAlignment="viewStart"
    android:textAppearance="@style/TextAppearance.AppCompat.Large"
    android:textSize="28sp"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
```



DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

Discover. Learn. Empower.

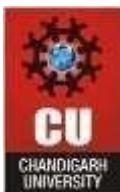
```
app:layout_constraintHorizontal_bias="0.065"
app:layout_constraintStart_toStartOf="parent"
app:layout_constraintTop_toBottomOf="@+id/textView2"
app:layout_constraintVertical_bias="0.01" />

<EditText
    android:id="@+id/editText2"
    android:layout_width="221dp"
    android:layout_height="108dp"
    android:hint="hello! I am ......."
    android:textAppearance="@style/TextAppearance.AppCompat.Large"
    android:textSize="28sp"
    app:layout_constraintBottom_toBottomOf="@+id/imageView3"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.084"
    app:layout_constraintStart_toStartOf="@+id/imageView3"
    app:layout_constraintTop_toTopOf="@+id/imageView3"
    app:layout_constraintVertical_bias="0.11" />

<Button
    android:id="@+id/button"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Send"
    android:textSize="20sp"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.867"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/imageView3"
    app:layout_constraintVertical_bias="0.542" />
```

Step2: Layout File for designing as:





DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

Discover. Learn. Empower.

Step3: java.main:

```
public class MainActivity extends AppCompatActivity {

    private static final int MY_PERMISSIONS_REQUEST_SEND_SMS = 0;
    EditText editText, editText2;

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

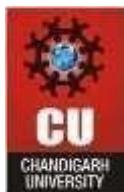
        editText = findViewById(R.id.editText);
        editText2 = findViewById(R.id.editText2);

        Button button = findViewById(R.id.button);
        button.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                sendSMS();
            }
        });
    }

    private void sendSMS() {
        String phoneNumber = editText.getText().toString();
        String message = editText2.getText().toString();

        if (ContextCompat.checkSelfPermission(this, Manifest.permission.SEND_SMS) != PackageManager.PERMISSION_GRANTED) {
            ActivityCompat.requestPermissions(this, new String[]{Manifest.permission.SEND_SMS},
                MY_PERMISSIONS_REQUEST_SEND_SMS);
        } else {
            SmsManager smsManager = SmsManager.getDefault();
            smsManager.sendTextMessage(phoneNumber, null, message, null, null);
            Toast.makeText(getApplicationContext(), "SMS Sent Successfully!", Toast.LENGTH_SHORT).show();
        }
    }

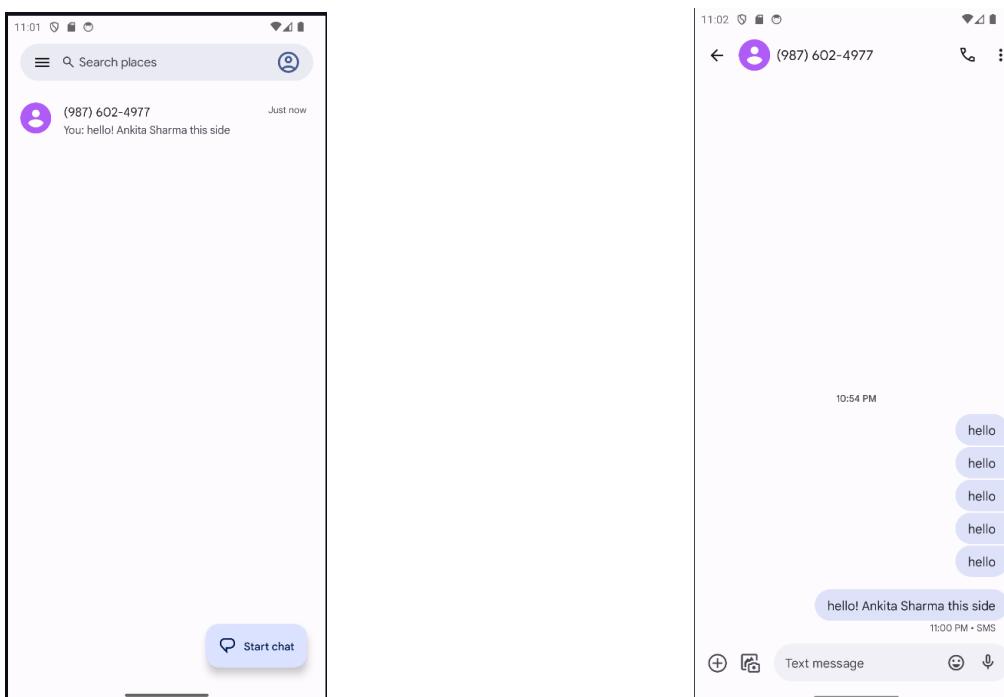
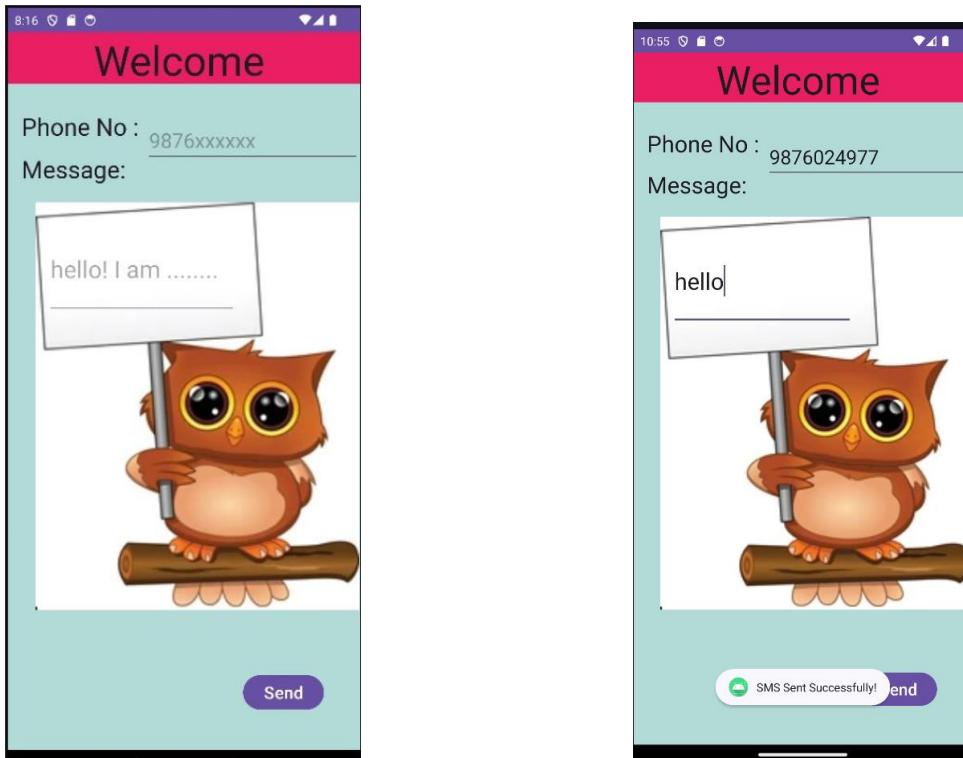
    @Override
    public void onRequestPermissionsResult(int requestCode, @NonNull String[] permissions, @NonNull int[] grantResults) {
        super.onRequestPermissionsResult(requestCode, permissions, grantResults);
        if (requestCode == MY_PERMISSIONS_REQUEST_SEND_SMS) {
            if (grantResults.length > 0 && grantResults[0] == PackageManager.PERMISSION_GRANTED) {
                sendSMS();
            } else {
                Toast.makeText(getApplicationContext(), "SMS Sending Failed, Please grant permission.", Toast.LENGTH_SHORT).show();
            }
        }
    }
}
```



DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

Discover. Learn. Empower.

Step 4: Running Emulator as:





DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

Discover. Learn. Empower.

Experiment 2.4

Student Name: Aafreen Khan

UID: 21BCS1397

Branch: B.E-CSE

Section/Group: CC-606 A

Semester: 6th

Date of Performance: 19/03/2024

Subject Name: MAD with Lab

Subject Code: 21CSH-355

1. Aim: Create an Android application using Fragments

2. Objective: The objective of an Android application using Fragments can be to enhance the user interface and improve the overall user experience by leveraging the benefits of fragment-based design. Fragments allow developers to create more modular, flexible, and scalable UI components.

3. Software Used : Android Studio

4. Theory :

Fragments are modular components in Android development that represent a portion of a user interface or behavior within an activity. They enable the creation of flexible and reusable UI elements, allowing developers to build responsive and adaptable applications. Fragments can be combined to construct multi-pane layouts, and enhance the overall scalability of Android applications.

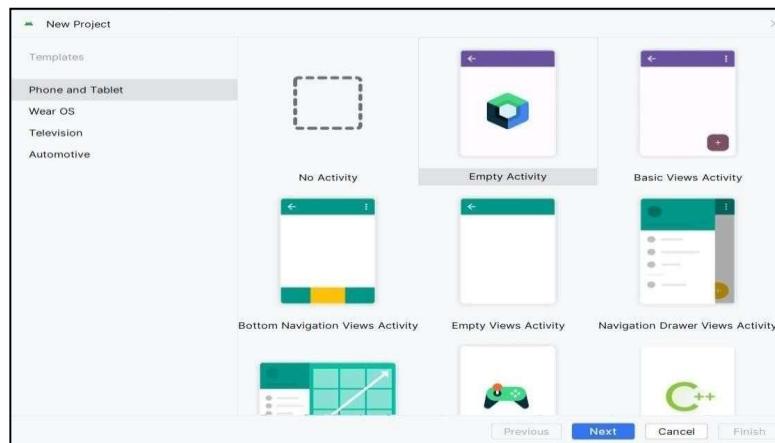
5. Procedure:

Step 1: Create a new project and select an empty view activity as we have done previous activities.



DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

Discover. Learn. Empower.



Step 2 : Create a new Android Application. This will create an XML file “activity_main.xml” and a Java File “MainActivity.Java”.

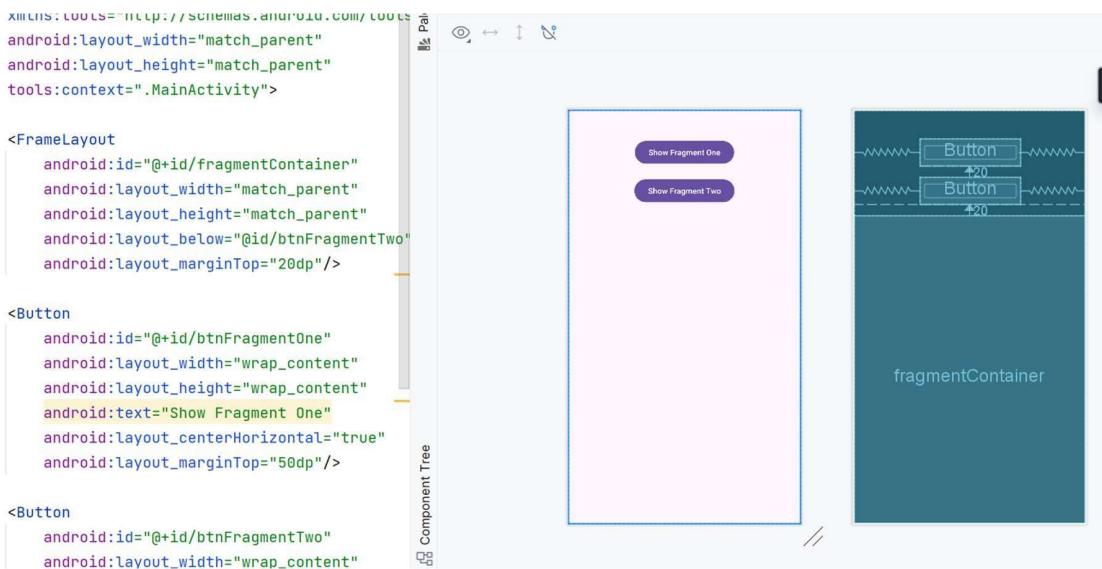
Step 3: Create Fragments

Fragment 1:

In the res/layout directory, create a new XML layout file for fragment_first.xml with a RelativeLayout and a Button.

Fragment 2:

Similarly, create a new XML layout file for fragment_second.xml with a RelativeLayout and a TextView to display text.



Step 4 : To create the Backend of Application, open the “MainActivity.java” file and instantiate the components made in the XML file using findViewById() method. This method binds the created object to the UI Components with the help of the assigned ID.

Step 5 :- Open the emulator and run the application to see how it works.



DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

Discover. Learn. Empower.

6. Code:

Mainactivity.java:

```
package com.example.exppp7;

import androidx.appcompat.app.AppCompatActivity;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;

public class MainActivity extends AppCompatActivity {

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

        final FragmentOne fragmentOne = new FragmentOne();
        final FragmentTwo fragmentTwo = new FragmentTwo();

        getSupportFragmentManager().beginTransaction()
            .replace(R.id.fragmentContainer, fragmentOne)
            .commit();

        Button btnFragmentOne = findViewById(R.id.btnFragmentOne);
        btnFragmentOne.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                getSupportFragmentManager().beginTransaction()
                    .replace(R.id.fragmentContainer, fragmentOne)
                    .commit();
            }
        });

        Button btnFragmentTwo = findViewById(R.id.btnFragmentTwo);
        btnFragmentTwo.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                getSupportFragmentManager().beginTransaction()
                    .replace(R.id.fragmentContainer, fragmentTwo)
                    .commit();
            }
        });
    }
}
```



DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

Discover. Learn. Empower.

}

Fragment_one.xml

```
<!-- fragment_one.xml -->
<FrameLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:background="#2a9d8f"
    android:layout_height="match_parent">

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Hello Everyone!!"
        android:textSize="32sp"
        android:textColor="#ffffffff"
        android:textStyle="bold"
        android:layout_gravity="center"/>

</FrameLayout>
```

Fragment_two.xml

```
<!-- fragment_two.xml -->
<FrameLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:background="#264653"
    android:layout_height="match_parent">

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Sonu this side!!"
        android:textColor="#ffffffff"
        android:textSize="32sp"
        android:textStyle="bold"
        android:layout_gravity="center"/>

</FrameLayout>
```

Layout File:

```
<RelativeLayout      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"
    tools:context=".MainActivity">
```



DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

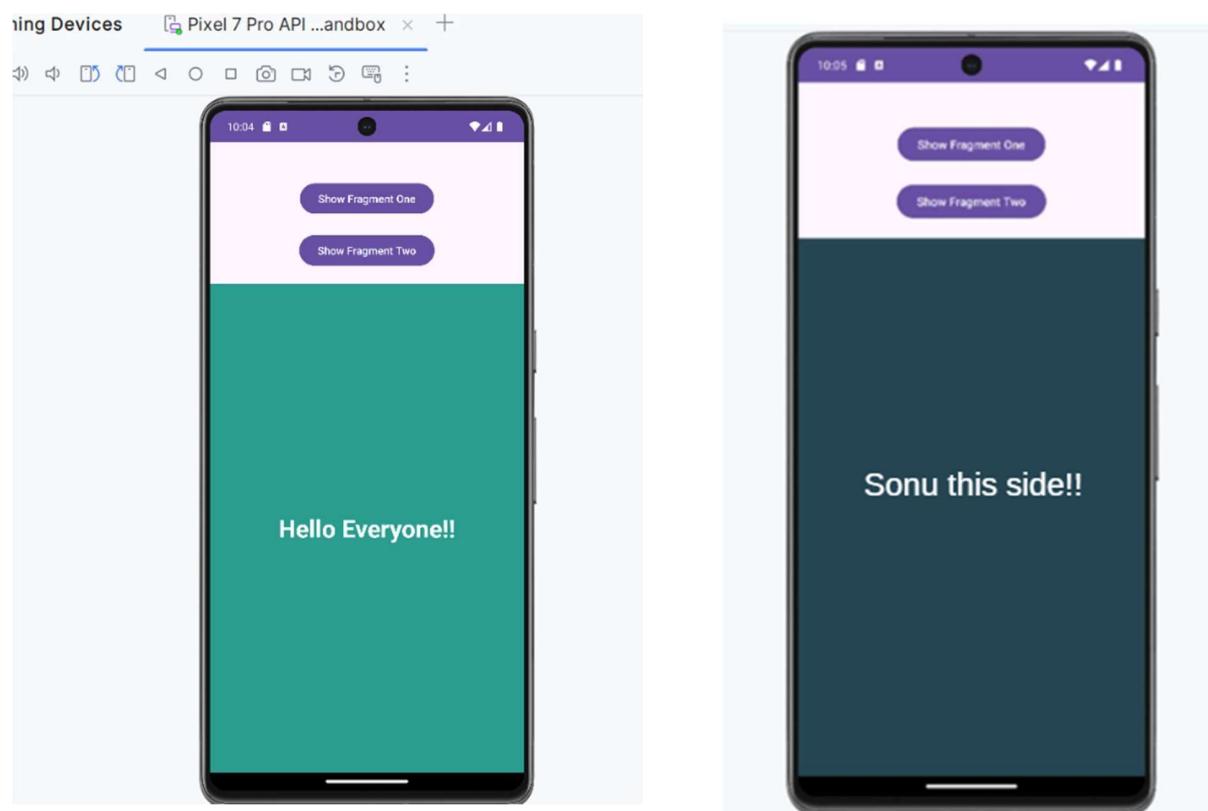
Discover. Learn. Empower.

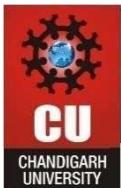
```
<FrameLayout
    android:id="@+id/fragmentContainer"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:layout_below="@+id/btnFragmentTwo"
    android:layout_marginTop="20dp"/>

<Button
    android:id="@+id	btnFragmentOne"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Show Fragment One"
    android:layout_centerHorizontal="true"
    android:layout_marginTop="50dp"/>

<Button
    android:id="@+id	btnFragmentTwo"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Show Fragment Two"
    android:layout_below="@+id/btnFragmentOne"
    android:layout_centerHorizontal="true"
    android:layout_marginTop="20dp"/>
</RelativeLayout>
```

7. OUTPUT





DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

Discover. Learn. Empower.

Experiment-3.3

Student Name: Aafreen Khan
Branch: BE-CSE
Semester: 6th
Subject Name: MAD LAB

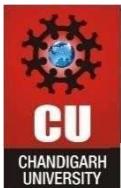
UID: 21BCS1397
Section/Group: 606-A
Date of Performance: 02/04/24
Subject Code: 21CSH-355

- 1. Aim:** Design the Android application using menus and action bar.
- 2. Objective:** The objective of designing an Android application using menus and the action bar is to create a user-friendly and consistent interface that allows users to navigate, access functionality, and perform actions efficiently.

3. Script:

XML:-

```
<?xml version="1.0" encoding="utf-8"?>
<menu xmlns:android="http://schemas.android.com/apk/res/android" >
    <item android:id="@+id/search_item" android:title="Search" />
    <item android:id="@+id/upload_item" android:title="Upload" />
    <item android:id="@+id/copy_item" android:title="Copy" />
    <item android:id="@+id/print_item" android:title="Print" />
    <item android:id="@+id/share_item" android:title="Share" />
    <item android:id="@+id/bookmark_item" android:title="BookMark" />
</menu>
<AbsoluteLayout
    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="fill_parent" android:layout_height="fill_parent"
    tools:context=".MainActivity">
    <!-- Setting up TextViews -->
    <TextView
        android:layout_width="wrap_content" android:layout_height="wrap_content"
        android:layout_x="100px" android:layout_y="300px" />
    <TextView
        android:layout_width="wrap_content" android:layout_height="wrap_content"
        android:layout_x="120px" android:layout_y="350px" />
</AbsoluteLayout>
<RelativeLayout
    android:layout_width="fill_parent" android:layout_height="fill_parent"
    xmlns:android="http://schemas.android.com/apk/res/android">
    <Button
        android:id="@+id/button1" android:layout_width="wrap_content"
        android:layout_height="wrap_content" android:text="Top Left Button"
        android:layout_alignParentLeft="true" android:layout_alignParentTop="true"/>
```



DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

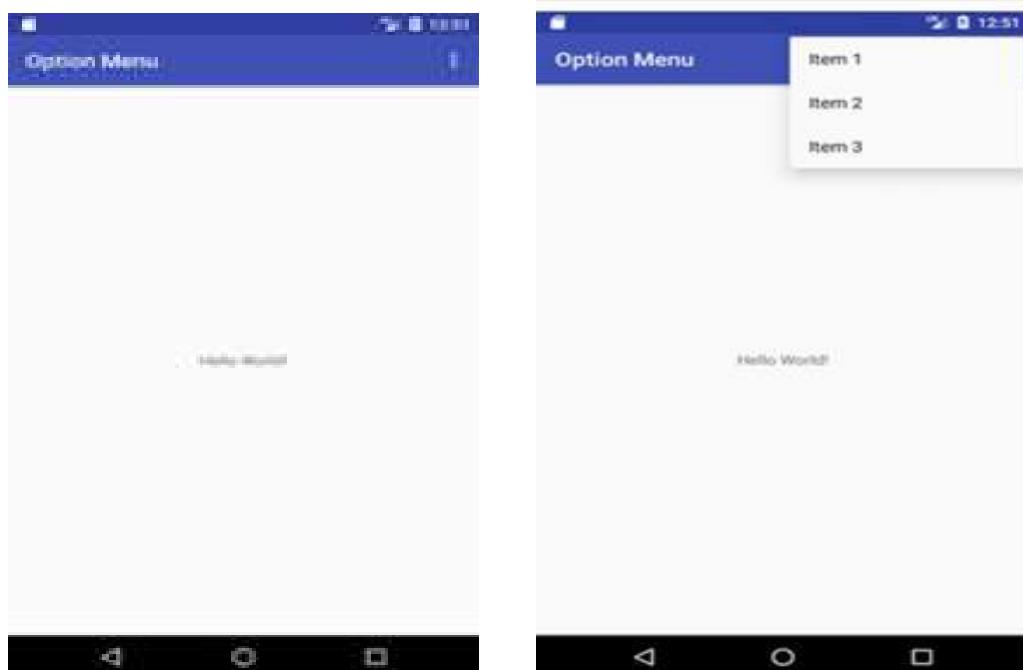
Discover. Learn. Empower.

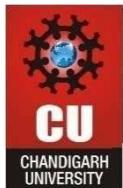
```
<Button
    android:id="@+id/button2" android:layout_width="wrap_content"
    android:layout_height="wrap_content" android:text="Top Right Button"
    android:layout_alignParentTop="true" android:layout_alignParentRight="true"/>
<Button
    android:id="@+id/button3" android:layout_width="wrap_content"
    android:layout_height="wrap_content" android:text="Bottom Left Button"
    android:layout_alignParentLeft="true" android:layout_alignParentBottom="true"/>
<Button
    android:id="@+id/button5" android:layout_width="fill_parent"
    android:layout_height="wrap_content" android:text="Middle Button"
    android:layout_centerVertical="true" android:layout_centerHorizontal="true"/>
</RelativeLayout>
```

JAVA Code:

```
import androidx.appcompat.app.AppCompatActivity;
import android.os.Bundle;
import android.widget.TextView;
public class MainActivity extends AppCompatActivity {
    TextView heading, subHeading;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        // Referencing the TextViews
        heading = (TextView) findViewById(R.id.heading);
        subHeading = (TextView) findViewById(R.id.subHeading);
    }
}
```

4. Output:





DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

Discover. Learn. Empower.



5. Learning Outcomes:

- Successful USE of your Android development environment.
- Project Workspace.
- Configuration button Completion.
- How to add new layouts



DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

Discover. Learn. Empower.

Experiment 3.2

Student Name: Aafreen Khan
Branch: BE-CSE-IT
Semester: 6th
Subject Name: MAD Lab

UID: 21BCS1397
Section/Group: CC_606-A
Date of Performance: 02/04/2024
Subject Code: 21CSH-355

Aim of the practical: Implement building blocks for Android Application using different layouts such as linear, relative and absolute.

Objective:- The objective of implementing building blocks for an Android application using different layouts such as linear, relative, and absolute is to create a diverse and visually appealing user interface that accommodates various design requirements. Different layout types offer flexibility in organizing UI components, and understanding their usage is crucial for effective Android app development.

Theory :-

- Linear Layout:** Arranges views linearly either horizontally or vertically.
- Relative Layout:** Positions views relative to each other or to the parent container.
- Absolute Layout:** Positions views at exact coordinates on the screen.
- Constraint Layout:** Positions views relative to each other using constraints for flexible UI.
- Grid Layout:** Arranges views in a grid-like structure of rows and columns.

1. Code:

XML Code:

```
activity_main.xml(Absolute Layout)
<?xml version="1.0" encoding="utf-8"?>
<AbsoluteLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent">

    <!-- Setting up TextViews -->
    <TextView
        android:id="@+id/heading"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Computer Science Portal"
        android:layout_x="100px"
        android:layout_y="300px" />
```



DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

Discover. Learn. Empower.

```
<TextView
    android:id="@+id/subHeading"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="GeeksForGeeks"
    android:layout_x="120px"
    android:layout_y="350px" />

</AbsoluteLayout>
```

Java Code:

MainActivity.java

```
package com.example.expt8;

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

public class MainActivity extends AppCompatActivity {

    TextView heading, subHeading;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        // Referencing the TextViews
        heading = (TextView) findViewById(R.id.heading);
        subHeading = (TextView) findViewById(R.id.subHeading);

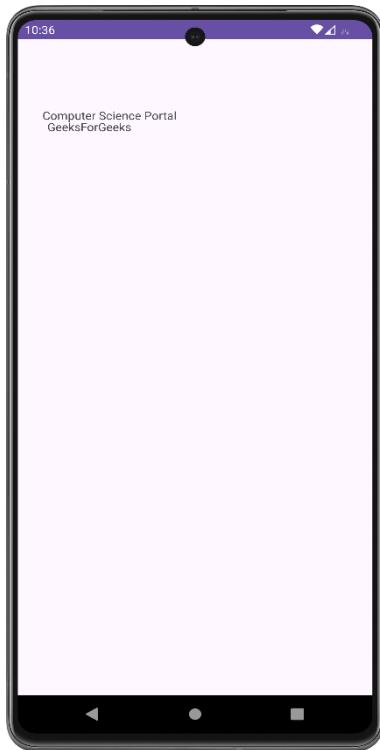
        // Setting text dynamically
        heading.setText("Computer Science Portal");
        subHeading.setText("GeeksForGeeks");
    }
}
```



DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

Discover. Learn. Empower.

2. OUTPUT:



Absolute Layout



Relative Layout



Linear Layout

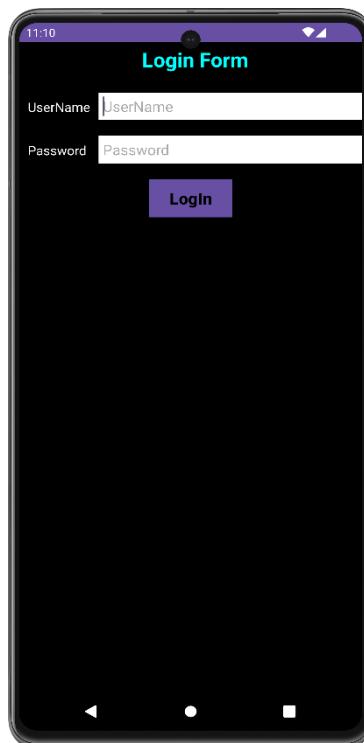


Table Layout

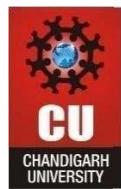


DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

Discover. Learn. Empower.

3. Learning Outcomes:

- Successful USE of your Android development environment.
- Learnt the concepts of layouts in Android Studio.
- Learnt about decrypted layouts.



DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

Discover Learn Improve

Experiment – 3.4

Student Name: AAFREEN KHAN
Branch: BE-CSE
Semester: 6th
Subject Name: MAD Lab

UID: 21BCS1397
Section/Group: CC-606-A
Date of Performance: 16/04/2024
Subject Code: 21CSH-355

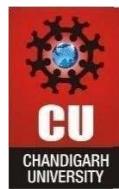
Aim of the practical: Create an Android application for user registration that stores the user details in a database table

Objective: The objective of an Android application for user registration that stores user details in a database table is to create a secure, efficient, and user-friendly registration system. This type of app is commonly developed for services that require user accounts, such as social media platforms, e-commerce applications, or any service where personalized user data needs to be stored.

1. CODE: XML:-

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout 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:orientation="vertical"
    tools:context=".MainActivity">
    <!--Edit text to enter course name-->
    <EditText
        android:id="@+id/idEdtCourseName"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_margin="10dp"
        android:hint="Enter course Name" />

    <!--edit text to enter course duration-->
    <EditText
        android:id="@+id/idEdtCourseDuration"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_margin="10dp"
```



DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

```
        android:hint="Enter Course Duration" />    <!--edit
text to display course tracks-->
<EditText
    android:id="@+id/idEdtCourseTracks"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_margin="10dp"
    android:hint="Enter Course Tracks" />

<!--edit text for course description-->
<EditText
    android:id="@+id/idEdtCourseDescription"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_margin="10dp"
    android:hint="Enter Course Description" />

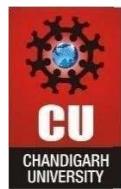
<!--button for adding new course-->
<Button
    android:id="@+id/idBtnAddCourse"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_margin="10dp"
    android:text="Add Course"
    android:textAllCaps="false" />

</LinearLayout> Java Code:-
```

Main.activity.java

```
import android.os.Bundle; import android.view.View;
import android.widget.Button; import
android.widget.EditText; import android.widget.Toast;
import androidx.appcompat.app.AppCompatActivity
public class MainActivity extends AppCompatActivity {
```

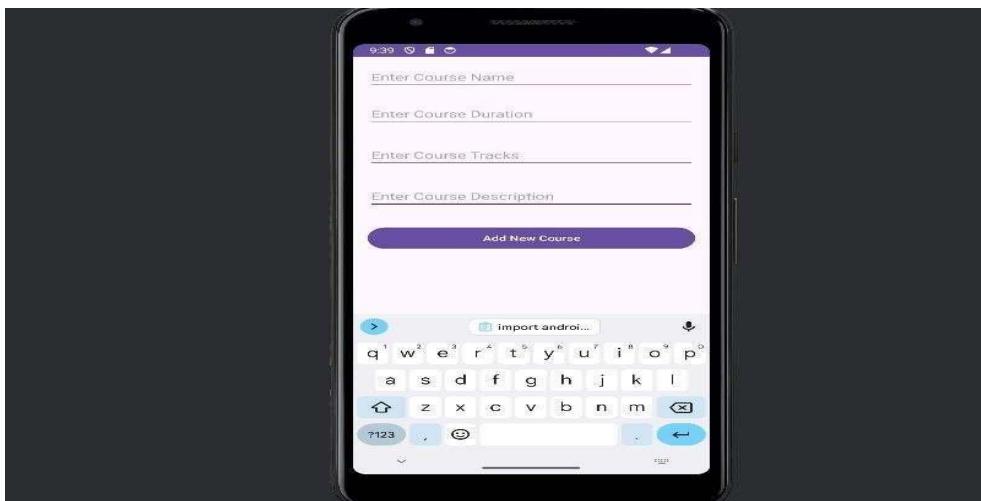
```
// creating variables for our edittext, button and dbhandler
private EditText
courseNameEdt, courseTracksEdt, courseDurationEdt, courseDescriptionEdt;
private Button addCourseBtn; private DBHandler dbHandler; @Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main) // initializing all our variables.
    courseNameEdt = findViewById(R.id.idEdtCourseName);
```



DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

```
Toast.LENGTH_SHORT).show();
        courseNameEdt.setText("");
        courseDurationEdt.setText("");
        courseTracksEdt.setText("");
        courseDescriptionEdt.setText("");
    }
});  
}  
DBHandler.java import  
android.content.ContentValues; import  
{@Override public void onUpgrade(SQLiteDatabase db, int oldVersion, int  
newVersion)  
{  
    db.execSQL("DROP TABLE IF EXISTS " + TABLE_NAME);  
onCreate(db);  
}  
}  
}
```

2. OUTPUT:



3. Learning Outcomes:

- Working with SQLite databases
- Implementing CRUD operations (Create, Read, Update, Delete)