

**MOBILE APPLICATION
DEVELOPMENT
(3170726)**

Index

Experiment Sr. No.	Tasks
1.	Install Android Studio with Latest Configuration in your System.
2.	Develop an android app which displays “Hello World” message
3.	Design Login activity and implement control events : Use EditText, Checkbox and Buttons
4.	Develop an android app which displays a form to get following information from user. 1) Username 2) Password 3) Email Address 4) Phone Number 5) Country
5.	Implement Practical 3 and 4 using following layouts: 1. Linear Layout 2. Relative Layout 3. Table Layout
6.	Create sample application that demonstrates activity life cycle’s all methods.
7.	Create Activities & implement following 1. Implicit intent 2. Explicit Intent 3. StartActivityForResult
8.	Using Android, Create a login Activity. It asks “username” and “password” from user. If username and password are valid, it displays Welcome message using new activity
9.	“Happy Birth Day” App using TextView and ImageView
10.	Create “Hello Toast” App by implementing a click handler method for the button to display a message on the screen when the user clicks. Use Linear Layout for creating view.
11.	Create a Recycler View and list the details of student using following fields: 1. Name 2. Address 3. Photo (Image) 4. Delete (Button Operation)
12.	Practical: Theme, Custom Styles, Drawables

13.	Create the MP3 player like application with service
14.	The Simple Calculator app has two edit texts and four buttons. When you enter two numbers and click a button, the app performs the calculation for that button and displays the result.
15.	Develop one App. Which Contains Specific User Interface and design Interface.
16.	Using experiment 4 and Save user data in a database.

Mandatory Online Courses:

1. <https://learn.udacity.com/courses/ud851>
2. <https://www.udemy.com/course/learn-android-application-development-y/>

Other

1. Mini Project by including all relevant concepts. (In team of max. 3)

EXPERIMENT-1

Title: Install Android Studio with Latest Configuration in your System.

- **Step 1:** Head over to [this link](#) to get the Android Studio executable or zip file.
- **Step 2:** Click on the **Download Android Studio** Button.



Android Studio provides the fastest tools for building apps on every type of Android device.

DOWNLOAD ANDROID STUDIO

4.1.3 for Windows 64-bit (896 MiB)



Click on the “I have read and agree with the above terms and conditions” checkbox followed by the download button.

The screenshot shows the 'Download Android Studio' page. At the top, it says 'Before downloading, you must agree to the following terms and conditions.' Below this is a scrollable box titled 'Terms and Conditions' containing the Android Software Development Kit License Agreement. The visible text includes: 'This is the Android Software Development Kit License Agreement', '1. Introduction', '1.1 The Android Software Development Kit (referred to in the License Agreement as the "SDK" and specifically including the Android system files, packaged APIs, and Google APIs add-ons) is licensed to you subject to the terms of the License Agreement. The License Agreement forms a legally binding contract between you and Google in relation to your use of the SDK.', '1.2 "Android" means the Android software stack for devices, as made available under the Android Open Source Project, which is located at the following URL: <http://source.android.com/>, as updated from time to time.', and '1.3 A "compatible implementation" means any Android device that (i) complies with the Android Compatibility Definition document, which can be found at the Android compatibility website (<http://source.android.com/compatibility>) and which may be updated from time to time; and (ii) successfully passes the Android'. Below the terms box is a checkbox labeled 'I have read and agree with the above terms and conditions'. Underneath the checkbox is a button labeled 'DOWNLOAD ANDROID STUDIO FOR WINDOWS'. At the bottom, the filename 'android-studio-ide-173.4819257-windows.exe' is displayed.

Click on the Save file button in the appeared prompt box and the file will start downloading.

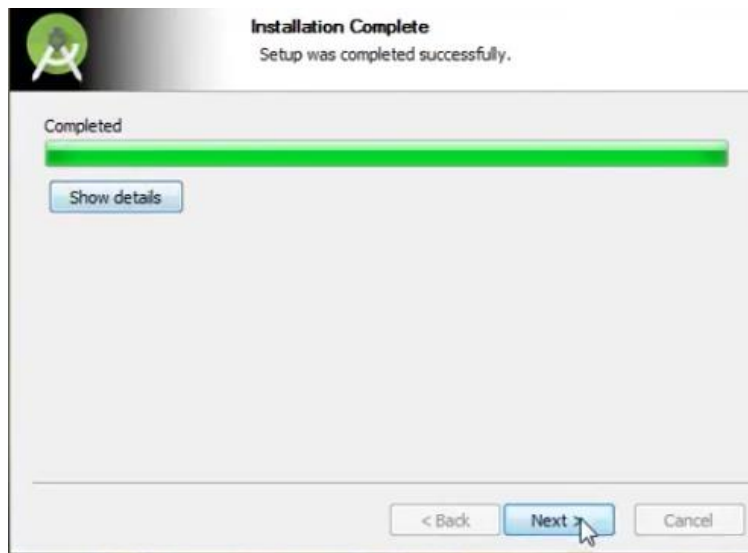


- **Step 3:** After the downloading has finished, open the file from downloads and run it. It will prompt the following dialog box.



Click on next. In the next prompt, it'll ask for a path for installation. Choose a path and hit next.

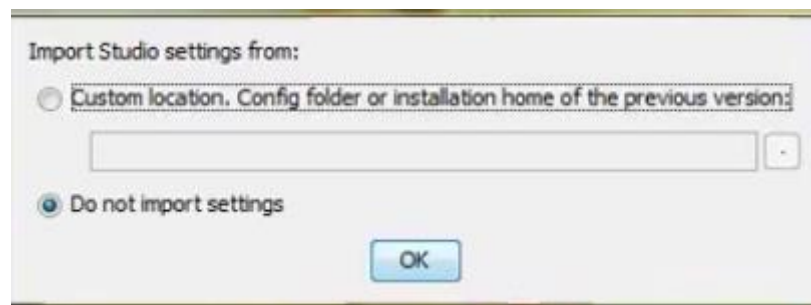
- **Step 4:** It will start the installation, and once it is completed, it will be like the image shown below.



Click on next.



- **Step 5:** Once “**Finish**” is clicked, it will ask whether the previous settings need to be imported [if the android studio had been installed earlier], or not. It is better to choose the ‘Don’t import Settings option’.



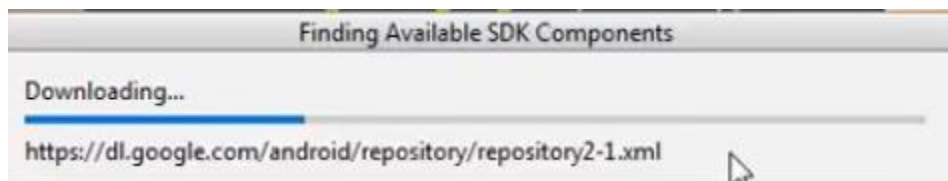
Click the **OK** button.



- **Step 6:** This will start the Android Studio.

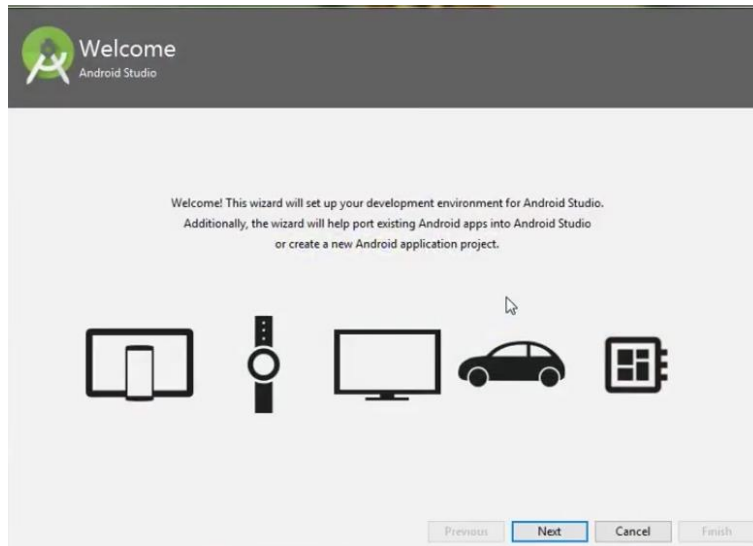


Meanwhile, it will be finding the available SDK components.

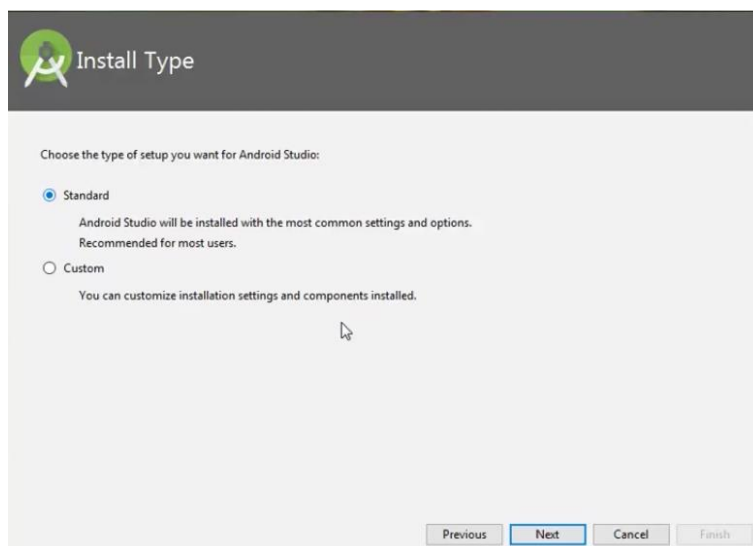




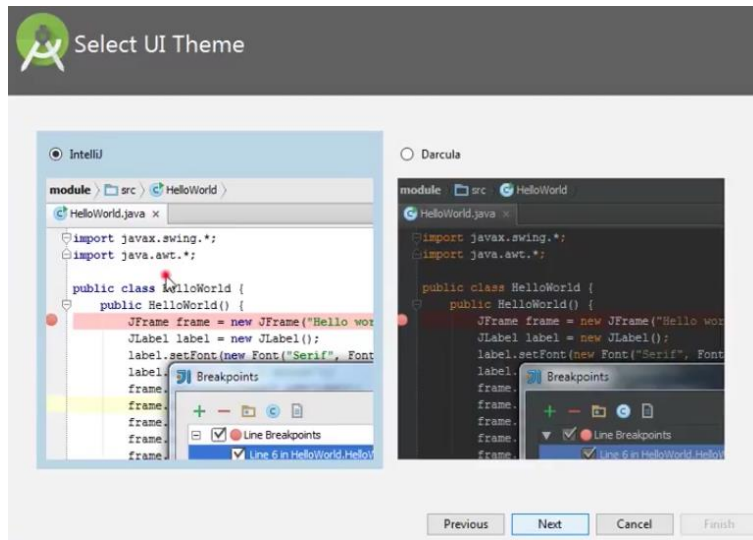
- **Step 7:** After it has found the SDK components, it will redirect to the Welcome dialog box.



Click on **Next**.

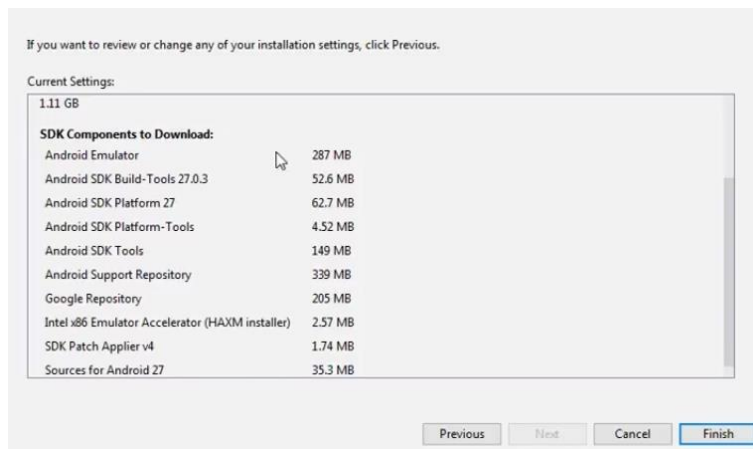


Choose Standard and click on Next. Now choose the theme, whether the **Light** theme or the **Dark** one. The light one is called the **IntelliJ** theme whereas the dark theme is called **Darcula**. Choose as required.

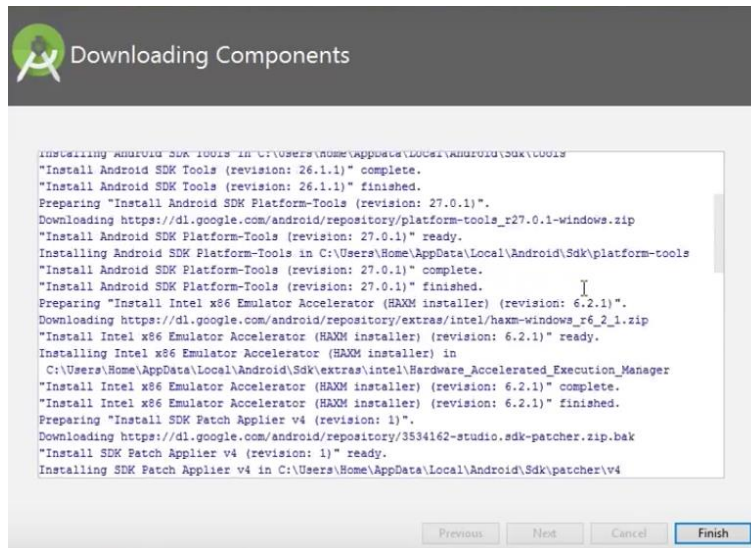


Click on the **Next** button.

- **Step 8:** Now it is time to download the SDK components.

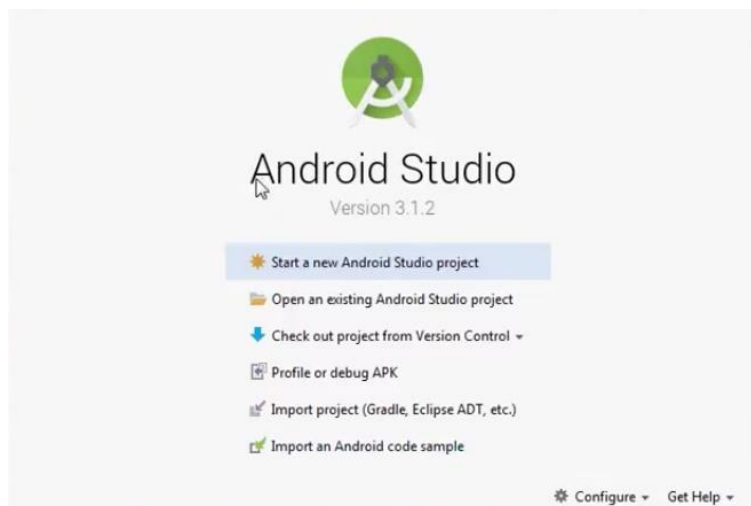


Click on Finish. Components begin to download let it complete.



The Android Studio has been successfully configured. Now it's time to launch and build apps. Click on the Finish button to launch it.

- **Step 9:** Click on **Start a new Android Studio project** to build a new app.



EXPERIMENT-2

Title: Develop an android app which displays “Hello World” message.

Components Used to Develop App:

<TextView/> :- To print static text on screen

CODE (LAYOUTS+ACTIVITIES):

Layouts:

Activity_main.xml

Description: activity_main.xml file works as home screen of android application. In our app it will show Hello World simple text using one textview component of android UI.

```
<?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="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>
```

Activities

MainActivity.java

Description: MainActivity.java file is used for business logic of different components of activity_main.xml file. It is used to execute different callback methods of android activity and creates the view of activity using onCreate() method.

```
package com.example.temp;

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);
    }
}
```

AndroidManifest.xml

Description: Manifest file is used for configuration and context information about the android application like icon of application, name, theme and it will also contains intents of the application.

```
<?xml version="1.0" encoding="utf-8"?>

<manifest xmlns:android="http://schemas.android.com/apk/res/android"

    package="com.example.myapplication">

    <application

        android:allowBackup="true"

        android:icon="@mipmap/ic_launcher"

        android:label="@string/app_name"

        android:roundIcon="@mipmap/ic_launcher_round"

        android:supportsRtl="true"
```

```
android:theme="@style/Theme.MyApplication">

<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>
```

Output Screenshot:

MainActivity:

Description: : This is MainActivity of the application which is created while we run the application first time. Here it will show Hello World text in the middle of the screen.



Conclusion: We are able to learn how to create first android application and how to use simple text view component in the application.

EXPERIMENT-3

Title: Design Login activity and implement control events : Use EditText, Checkbox and Buttons.

Components used to develop app:

<TextView/>: to print static text on the screen

<EditText/>: to take username, password, email, phone no. and country from the user

<CheckBox/>: to print user input on screen with if condition

<Button/>: clickable component to submit the user entered values

CODE (LAYOUTS+ACTIVITIES):

Layouts:-

Activity_main.xml

Description: TextView, EditText and Buttons are majorly used components in any application. TextView is just static text on the screen. EditText is used for user input box and button is used to perform some activity or event while clicking happens on it. Different attributes of this components are used to give style and orientation to it.

```
<?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:background="#040404"
tools:context=".MainActivity">

    <TextView
        android:id="@+id/tvword"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:textSize="26sp"
        android:text=""
        android:textColor="#FF0000"
        app:layout_constraintBottom_toBottomOf="parent"
```

```
app:layout_constraintEnd_toEndOf="parent"  
app:layout_constraintStart_toStartOf="parent"  
app:layout_constraintTop_toTopOf="parent"  
app:layout_constraintVertical_bias="0.165" />
```

```
<TextView  
    android:id="@+id/tvun"  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:layout_marginTop="36dp"  
    android:text="User Name"  
    android:textColor="#FD0000"  
    android:textSize="26sp"  
    app:layout_constraintEnd_toEndOf="parent"  
    app:layout_constraintHorizontal_bias="0.109"  
    app:layout_constraintStart_toStartOf="parent"  
    app:layout_constraintTop_toBottomOf="@+id/tvword" />
```

```
<Button  
    android:id="@+id/button"  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:layout_marginTop="56dp"  
    android:text="Login"  
    app:layout_constraintBottom_toBottomOf="parent"  
    app:layout_constraintEnd_toEndOf="parent"  
    app:layout_constraintHorizontal_bias="0.498"  
    app:layout_constraintStart_toStartOf="parent"  
    app:layout_constraintTop_toBottomOf="@+id/editTextTextPassword"  
    app:layout_constraintVertical_bias="0.0" />
```

```
<TextView  
    android:id="@+id/tvpw"  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:layout_marginTop="32dp"  
    android:text="Password"  
    android:textColor="#FF0000"  
    android:textSize="26sp"  
    app:layout_constraintEnd_toEndOf="parent"  
    app:layout_constraintHorizontal_bias="0.104"  
    app:layout_constraintStart_toStartOf="parent"  
    app:layout_constraintTop_toBottomOf="@+id/editTextTextPersonName" />
```



```
<EditText
    android:id="@+id/editTextTextPersonName"
    android:layout_width="350dp"
    android:layout_height="wrap_content"
    android:layout_marginTop="36dp"
    android:ems="10"
    android:inputType="textPersonName"
    android:textSize="26sp"
    android:hint="Name"
    android:textColorHint="#FF0000"
    android:textColor="#FF0000"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/tvun" />

<CheckBox
    android:id="@+id/cbSave"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Save your details"
    android:textSize="26sp"
    android:textColor="#FF0000"
    app:layout_constraintBottom_toBottomOf="@id/button"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.113"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="@id/editTextTextPassword" />

<EditText
    android:id="@+id/editTextTextPassword"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginTop="36dp"
    android:ems="10"
    android:hint="Type Password here"
    android:textColorHint="#FF0000"
    android:inputType="textPassword"
    android:textColor="#FF0202"
    android:textSize="26sp"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.292"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/tvpw" />
```

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

Activities

MainActivity.java

Description: onCreate() is the first callback method which will execute while loading the application. It contains business logic of the layout components.

```
package com.example.myapplication;

import androidx.appcompat.app.AppCompatActivity;

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

public class MainActivity extends AppCompatActivity {

    TextView tvuser, tvpass, tvmsg;
    EditText etuser, etpass;
    Button btnLogin;
    CheckBox ckbx;
    String un = "admin", pwd = "admin", user, pass;

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

        tvuser = findViewById(R.id.tvun);
        tvpass = findViewById(R.id.tvpw);
        tvmsg = findViewById(R.id.tvword);
        etuser = findViewById(R.id.editTextTextPersonName);
        etpass = findViewById(R.id.editTextTextPassword);
        btnLogin = findViewById(R.id.button);
        ckbx = findViewById(R.id.cbSave);
    }
}
```

```
btnLogin.setOnClickListener(new View.OnClickListener() {  
    @Override  
    public void onClick(View view) {  
        user = etuser.getText().toString();  
        pass = etpass.getText().toString();  
        Log.d("Login", "clicked on login");  
        if (ckbx.isChecked()){  
            if (user.equals(un) && pass.equals(pwd)) {  
                tvmsg.setText("Correct");  
                Log.d("Login", "user and password is Correct");  
            } else {  
                tvmsg.setText("Incorrect");  
                Log.d("Login", "user and password is Incorrect");  
            }  
        }  
        else  
        {  
            tvmsg.setText("Checkbox is not selected");  
        }  
    }  
});  
  
}  
  
}
```

AndroidManifest.xml

Description: Manifest file is used for configuration and context information about the android application like icon of application, name, theme and it will also contains intents of the application.

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

```
android:allowBackup="true"

android:icon="@mipmap/ic_launcher"

android:label="@string/app_name"

android:roundIcon="@mipmap/ic_launcher_round"

android:supportsRtl="true"

android:theme="@style/Theme.LoginPage">

<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>

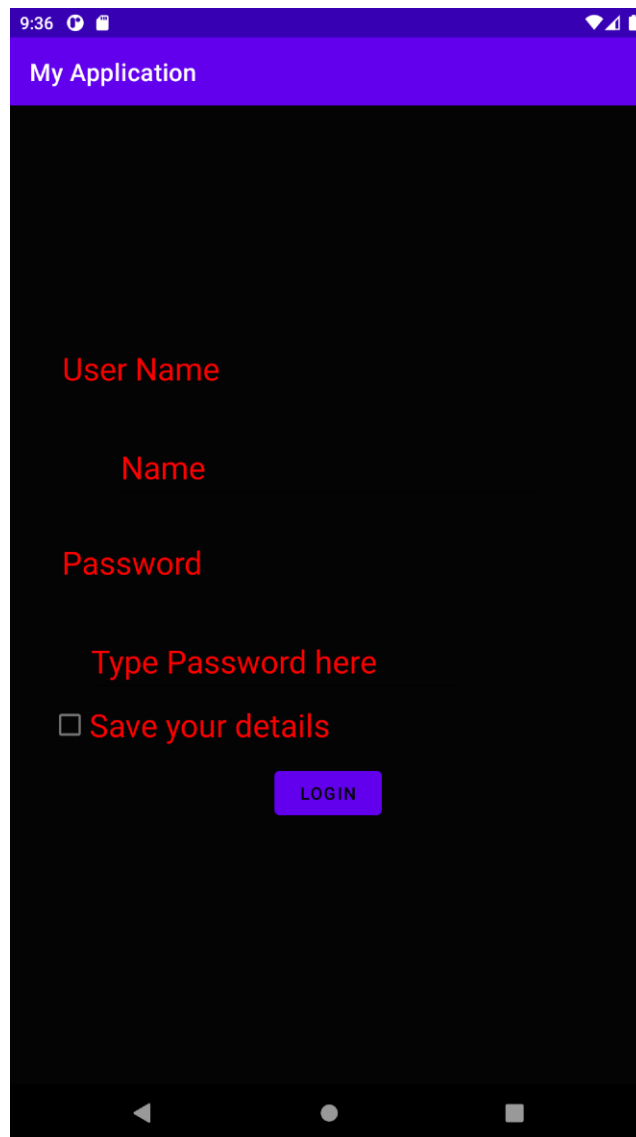
</application>

</manifest>
```

Output Screenshot:

MainActivity

Description: This main activity will show textview and edittext for username and password with one check box and submit button. Here all the components are aligned in center and spacing between them is equal which occupies entire screen.



EXPERIMENT-4

Title: Develop an android app which displays a form to get following information from user. 1) Username 2) Password 3) Email Address 4) Phone Number 5) Country

Components used to develop app:

<TextView/>: to print static text on the screen

<EditText/>: to take username, password, email, phone no. and country from the user

<CheckBox/>: to print user input on screen with if condition

<Button/>: clickable component to submit the user entered values

CODE (LAYOUTS+ACTIVITIES):

Layouts:-

Activity_main.xml

Description: This main activity layout shows textviews and edittexts to take input from user like username, password, email, phone no., country with center alignment and equal spacing.

```
<?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/fname"
        android:layout_width="250sp"
        android:layout_height="wrap_content"
        android:hint="Username"
        app:layout_constraintBottom_toBottomOf="@id/lname"
        app:layout_constraintLeft_toLeftOf="parent"
        app:layout_constraintRight_toRightOf="parent"
        app:layout_constraintTop_toTopOf="parent" />
```

```
<EditText
    android:id="@+id/lname"
    android:layout_width="250sp"
    android:layout_height="wrap_content"
    android:hint="Password"
    android:inputType="textPassword"
    app:layout_constraintBottom_toBottomOf="@id/email"
    app:layout_constraintLeft_toLeftOf="parent"
    app:layout_constraintRight_toRightOf="parent"
    app:layout_constraintTop_toTopOf="@id/fname" />

<EditText
    android:id="@+id/email"
    android:layout_width="250dp"
    android:layout_height="wrap_content"
    android:hint="@string/enter_email"
    app:layout_constraintBottom_toBottomOf="@id/phn"
    app:layout_constraintLeft_toLeftOf="parent"
    app:layout_constraintRight_toRightOf="parent"
    app:layout_constraintTop_toTopOf="@id/lname"
    android:inputType="textEmailAddress"/>

<EditText
    android:id="@+id/phn"
    android:layout_width="250dp"
    android:layout_height="wrap_content"
    android:hint="@string/enter_ph"
    app:layout_constraintBottom_toBottomOf="@id/country"
    app:layout_constraintLeft_toLeftOf="parent"
    app:layout_constraintRight_toRightOf="parent"
    app:layout_constraintTop_toTopOf="@id/email"
    android:inputType="phone"/>

<EditText
    android:id="@+id/country"
    android:layout_width="250dp"
    android:layout_height="wrap_content"
    android:hint="@string/enter_country"
    app:layout_constraintBottom_toBottomOf="@id/checkbox"
    app:layout_constraintLeft_toLeftOf="parent"
    app:layout_constraintRight_toRightOf="parent"
    app:layout_constraintTop_toTopOf="@id/phn" />

<CheckBox
```

```
android:id="@+id/checkbox"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:text="@string/check_msg"
app:layout_constraintBottom_toBottomOf="@id/button"
app:layout_constraintLeft_toLeftOf="parent"
app:layout_constraintRight_toRightOf="parent"
app:layout_constraintTop_toTopOf="@id/country" />
```

<Button

```
android:id="@+id/button"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:background="#4CAF50"
android:paddingStart="10dp"
android:paddingEnd="10dp"
android:text="@string/btn"
android:textColor="@android:color/background_light"
android:textSize="24sp"
app:layout_constraintBottom_toBottomOf="parent"
app:layout_constraintEnd_toEndOf="parent"
app:layout_constraintStart_toStartOf="parent"
app:layout_constraintTop_toTopOf="@id/checkbox" />
```

<TextView

```
android:id="@+id/txtinfo"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:text=""
android:textSize="32sp"
android:textAlignment="viewStart"
android:textColor="@color/purple_500"
app:layout_constraintBottom_toBottomOf="parent"
app:layout_constraintLeft_toLeftOf="parent"
app:layout_constraintRight_toRightOf="parent"
app:layout_constraintTop_toTopOf="@id/button"
/>
```

</androidx.constraintlayout.widget.ConstraintLayout>

Activities

MainActivity.java

Description: This MainActivity class file contains business logic to handle user input data. onClick() and OnClickListener() methods will be used to get user input string and store it in variables respectively. isChecked() method in if condition is used to print user input data on the screen if the check box is checked.

```
package com.example.loginpage;

import androidx.appcompat.app.AppCompatActivity;

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

public class MainActivity extends AppCompatActivity {

    EditText fn, ln, em, ph, cn;
    Button btn;
    CheckBox ck;
    TextView info;
    String FName, Lname, Email, Phone, Country;

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

        fn = findViewById(R.id.fname);
        ln = findViewById(R.id.lname);
        em = findViewById(R.id.email);
        ph = findViewById(R.id.phn);
        cn = findViewById(R.id.country);
        ck = findViewById(R.id.checkBox);
        btn = findViewById(R.id.button);
        info = findViewById(R.id.txtinfo);

        btn.setOnClickListener(new View.OnClickListener() {
            @Override
```

```
public void onClick(View view) {
    Fname = fn.getText().toString();
    Lname = ln.getText().toString();
    Email = em.getText().toString();
    Phone = ph.getText().toString();
    Country = cn.getText().toString();

    if(ck.isChecked())
    {
        info.setText(Fname + "\n" + Lname + "\n" + Email + "\n" + Phone + "\n" +
Country);
    }
    else{
        info.setText("Information Saved");
    }
}
});
}
```

AndroidManifest.xml

Description: Manifest file is used for configuration and context information about the android application like icon of application, name, theme and it will also contains intents of the application.

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

    <application
        android:allowBackup="true"
        android:icon="@mipmap/ic_launcher"
        android:label="@string/app_name"
        android:roundIcon="@mipmap/ic_launcher_round"
        android:supportsRtl="true"
        android:theme="@style/Theme.LoginPage">
        <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>
```

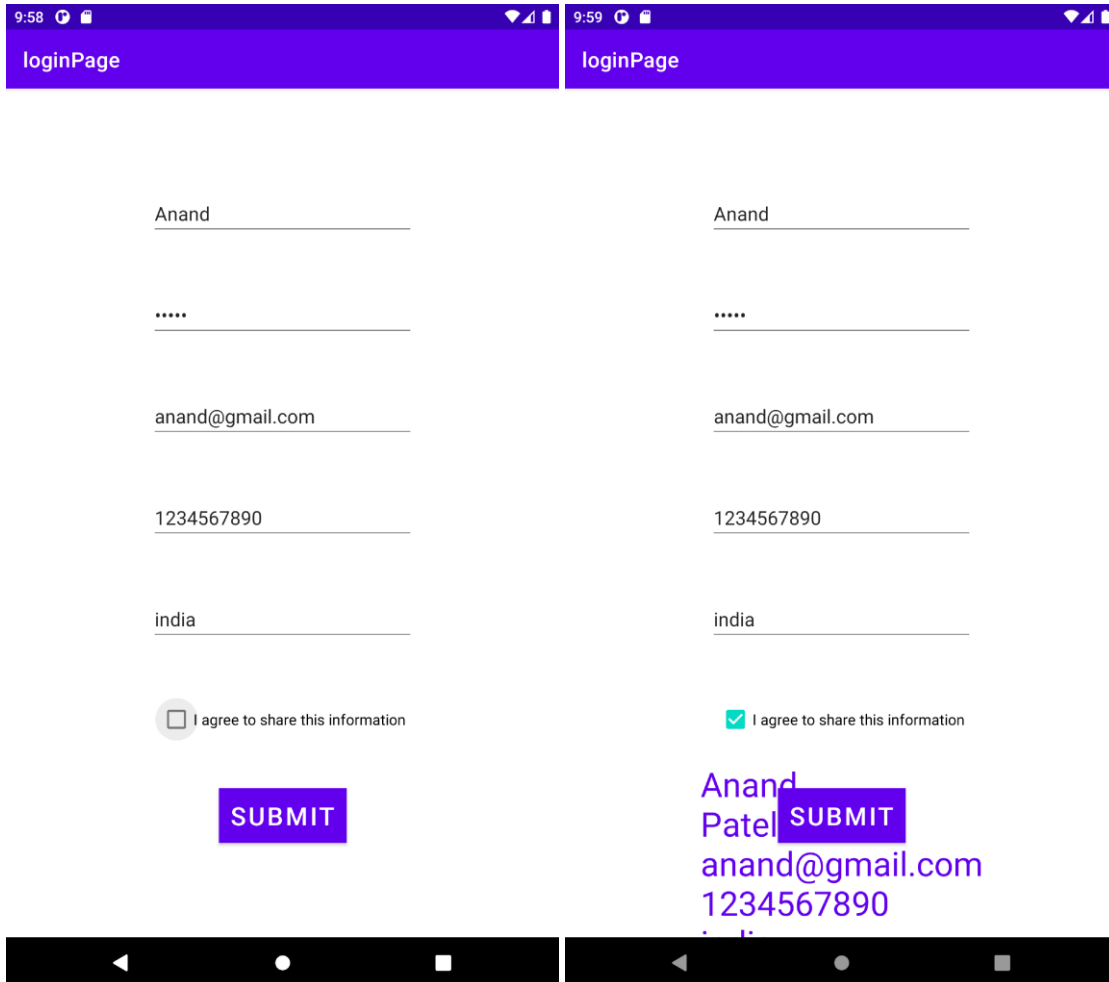
</activity>
 </application>

</manifest>

Output Screenshot:

MainActivity

Description: This layout contains textview, edittext for username, password, email, phone no. and country along with check box and submit button.



The image displays two side-by-side screenshots of a mobile application's login screen. Both screens have a dark blue header bar with the text 'loginPage'. The status bar at the top shows the time as 9:58 on the left and 9:59 on the right.

The login form consists of the following fields and elements:

- Username:** A text input field containing 'Anand'.
- Password:** A text input field with masked characters '*****'.
- Email:** A text input field containing 'anand@gmail.com'.
- Phone Number:** A text input field containing '1234567890'.
- Country:** A text input field containing 'india'.
- Agreement:** A checkbox labeled 'I agree to share this information'. In the right screenshot, the checkbox is checked.
- Submit Button:** A blue button with the text 'SUBMIT'.

In the right screenshot, the text 'Anand Patel', 'anand@gmail.com', and '1234567890' is overlaid on the bottom right of the form, likely representing the user's details after a successful login.

EXPERIMENT-5

Title: Implement Practical 3 and 4 using following layouts:

- 1. Linear Layout**
- 2. Relative Layout**
- 3. Table Layout**

Components used to develop app:

<TextView/>: to print static text on the screen

<EditText/>: to take username, password, email, phone no. and country from the user

<CheckBox/>: to print user input on screen with if condition

<Button/>: clickable component to submit the user entered values

1. Linear Layout

Layouts:-

Description: This main activity will show textview and edittext for username and password with one check box and submit button. Here all the components are in Linear layout.

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">
<LinearLayout
    android:id="@+id/linearLayout"
    android:layout_width="270dp"
    android:layout_height="486dp"
    android:layout_marginStart="66dp"
    android:layout_marginTop="131dp"
    android:layout_marginEnd="66dp"
    android:orientation="vertical"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent">
    <TextView
        android:id="@+id/textView"
```



```
        android:layout_width="match_parent"
        android:layout_height="40dp"
        android:text=""
        android:textAlignment="center"
        android:textAllCaps="true"
        android:textColor="#090909"
        android:textSize="24sp"
        android:textStyle="bold" />
    <EditText
        android:id="@+id/editTextTextPersonName"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_marginTop="50dp"
        android:ems="10"
        android:inputType="textPersonName"
        android:text="Username" />
    <EditText
        android:id="@+id/editTextTextPersonName2"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_marginLeft="0dp"
        android:layout_marginTop="50dp"
        android:layout_marginEnd="0dp"
        android:layout_marginRight="0dp"
        android:layout_marginBottom="0dp"
        android:ems="10"
        android:inputType="textPersonName"
        android:text="Password" />
    <CheckBox
        android:id="@+id/checkBox"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_marginLeft="16dp"
        android:layout_marginTop="50dp"
        android:layout_marginEnd="50dp"
        android:layout_marginRight="16dp"
        android:text="Terms and Conditions" />
    <Button
        android:id="@+id/button"
        android:layout_width="match_parent"
        android:layout_height="160dp"
        android:insetLeft="80dp"
        android:insetTop="80dp"
        android:insetRight="80dp"
```

```
        android:insetBottom="16dp"
        android:paddingBottom="4dp"
        android:text="@string/submit" />
    </LinearLayout>
</androidx.constraintlayout.widget.ConstraintLayout>
```

MainActivity.java

Description: MainActivity.java file is used for business logic of different components of activity_main.xml file. It is used to execute different callback methods of android activity and creates the view of activity using onCreate() method.

```
package com.example.linearlayout;

import androidx.appcompat.app.AppCompatActivity;

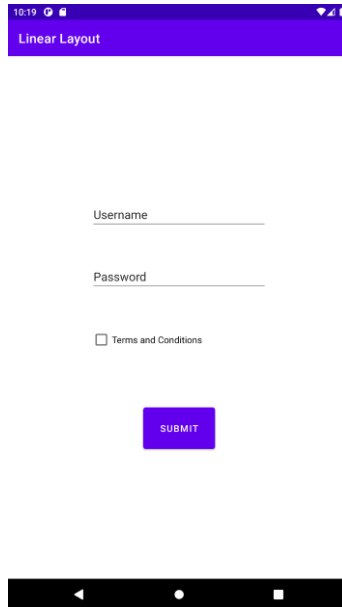
import android.os.Bundle;

public class MainActivity extends AppCompatActivity {

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

Output Screenshot

Description: This main activity will show textview and edittext for username and password with one check box and submit button. Here all the components are in linear layout.



2. Relative Layout

Layouts:-

Description: This main activity will show textview and edittext for username and password with one check box and submit button. Here all the components are in Relative layout.

Activity_main.xml

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

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_above="@id/edEmail"
        android:layout_centerHorizontal="true"
        android:text="LOGIN"
        android:textSize="28sp" />

    <EditText
        android:id="@+id/edEmail"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_above="@id/edPass"
        android:layout_marginLeft="20dp"
        android:layout_marginTop="15dp"
        android:layout_marginRight="20dp"
        android:hint="Email" />

    <EditText
        android:id="@+id/edPass"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_above="@id/centerLine"
        android:layout_marginLeft="20dp"
        android:layout_marginTop="15dp"
        android:layout_marginRight="20dp"
        android:hint="Password"
        android:inputType="textPassword" />

    <RelativeLayout
        android:id="@+id/centerLine"
```



```
android:layout_width="match_parent"  
android:layout_height="wrap_content"  
android:layout_centerInParent="true" />
```

```
<CheckBox  
    android:id="@+id/cb"  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:layout_below="@id/edPass"  
    android:layout_marginLeft="20dp"  
    android:layout_marginTop="15dp"  
    android:text="Remember Me?" />
```

```
<Button  
    android:id="@+id/btnLogin"  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:layout_below="@id/cb"  
    android:layout_centerHorizontal="true"  
    android:layout_marginTop="15dp"  
    android:text="Login" />
```

```
</RelativeLayout>
```

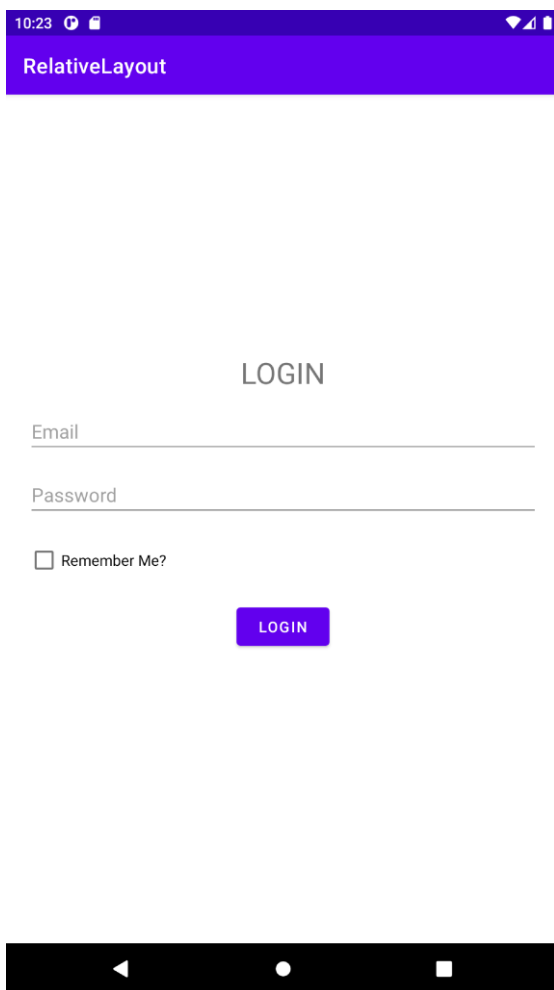
MainActivity.java

Description: MainActivity.java file is used for business logic of different components of activity_main.xml file. It is used to execute different callback methods of android activity and creates the view of activity using onCreate() method.

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

Output Screenshot

Description: This main activity will show textview and edittext for username and password with one check box and submit button. Here all the components are in Relative layout.



3. Table Layout

LAYOUTS:-

Activity_main.xml

Description: This main activity will show textview and edittext for username and password with one check box and submit button. Here all the components are in Table layout.

```
<?xml version="1.0" encoding="utf-8"?>
<TableLayout 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">

    <TableRow
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_marginTop="40dp">

        <TextView
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:gravity="center"
            android:text="LOGIN"
            android:textSize="28sp" />
    </TableRow>

    <TableRow
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_marginLeft="20dp"
        android:layout_marginTop="20dp">

        <TextView
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:text="Email : " />

        <EditText
            android:layout_width="200dp"
            android:layout_height="wrap_content"
```

```
        android:layout_marginLeft="20dp"
        android:hint="Email" />
</TableRow>

<TableRow
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_marginLeft="20dp">

    <TextView
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="Password : " />

    <EditText
        android:layout_width="200dp"

        android:layout_height="wrap_content"
        android:layout_marginLeft="20dp"
        android:hint="Password" />
</TableRow>

<TableRow
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginLeft="20dp">

    <CheckBox
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="Remeber Me?" />
</TableRow>

<TableRow
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginLeft="20dp">

    <Button
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Login" />
</TableRow>
</TableLayout>
```

MainActivity.java

Description: MainActivity.java file is used for business logic of different components of activity_main.xml file. It is used to execute different callback methods of android activity and creates the view of activity using onCreate() method.

```
package com.example.linearlayout;

import androidx.appcompat.app.AppCompatActivity;

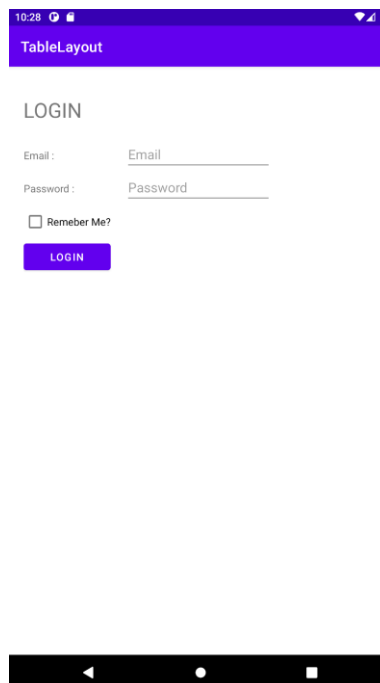
import android.os.Bundle;

public class MainActivity extends AppCompatActivity {

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

Output Screenshot

Description: This main activity will show textview and edittext for username and password with one check box and submit button. Here all the components are in Table layout.



EXPERIMENT-6

Title: Create sample application that demonstrates activity life cycle's all methods.

Components used to develop app:

<TextView/>: to print static text on the screen

CODE (LAYOUTS+ACTIVITIES):

Layouts:-

Activity_main.xml

Description: activity_main.xml file works as home screen of android application. In our app it will show Hello World simple text using one textview component of android UI.

```
<?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="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>
```

Activities

MainActivity.java

Description: This file is used to see the activity life cycle of the android. Android life cycle contains onCreate(), onStart(), onResume(), onPause(), onStop(), onRestart(), onDestroy() methods

```
package com.example.activitylifecycle;

import androidx.appcompat.app.AppCompatActivity;

import android.os.Bundle;
import android.util.Log;

public class MainActivity extends AppCompatActivity {

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

    @Override
    protected void onStart() {
        super.onStart();
        Log.d("lifecycle", "onStart invoked");
    }

    @Override
    protected void onResume() {
        super.onResume();
        Log.d("lifecycle", "onResume invoked");
    }

    @Override
    protected void onPause() {
        super.onPause();
        Log.d("lifecycle", "onPause invoked");
    }

    @Override
    protected void onStop() {
        super.onStop();
        Log.d("lifecycle", "onStop invoked");
    }

    @Override
```



```
protected void onRestart() {  
    super.onRestart();  
    Log.d("lifecycle","onRestart invoked");  
}  
@Override  
protected void onDestroy() {  
    super.onDestroy();  
    Log.d("lifecycle","onDestroy invoked");  
}  
}
```

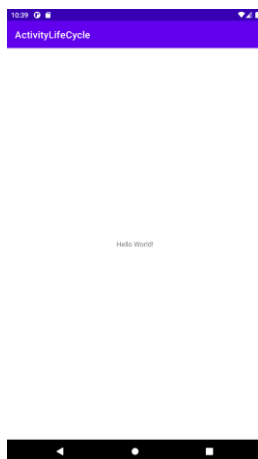

Output Screenshot:

Description: This is MainActivity of the application which is created while we run the application first time. Here it will show Hello World text in the middle of the screen. And we perform activity life cycle over there.

OnStart()



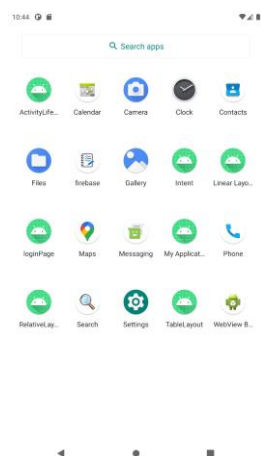
OnResume()



OnPause()



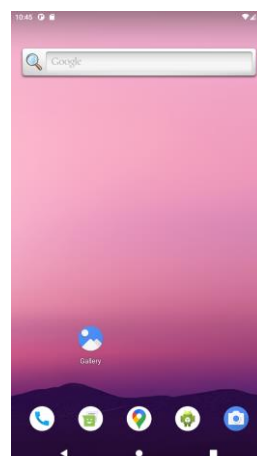
OnStop()



OnRestart()



OnDestroy()



EXPERIMENT-7

Title: Create Activities & implement following

- 1. Implicit intent**
- 2. Explicit Intent**
- 3. StartActivityForResult**

Components used to develop app:

<TextView/>: to print static text on the screen

<EditText/>: to take username, password, email, phone no. and country from the user

<ImageView/>: Displays image resources, for example Bitmap or Drawable resources

<Button/>: clickable component to submit the user entered values

CODE (LAYOUTS+ACTIVITIES):

Layouts:-

Activity_main.xml

Description: This file contains four different button to implement explicit intent.

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".MainActivity">

    <Button
        android:id="@+id/Liner_layout"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginTop="120dp"
        android:onClick="Activity_Linear_Layout"
        android:text="Linear Layout"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent" />

    <Button
        android:id="@+id/Relative_layout"
```

```
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_marginTop="80dp"
android:onClick="Activity_Relative_Layout"
android:text="Relative Layout"
app:layout_constraintEnd_toEndOf="parent"
app:layout_constraintStart_toStartOf="parent"
app:layout_constraintTop_toBottomOf="@+id/Liner_layout" />
```

```
<Button
    android:id="@+id/Table_layout"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginTop="80dp"
    android:onClick="Activity_Table_Layout"
    android:text="Table Layout"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/Relative_layout" />
```

```
<Button
    android:id="@+id/Other_app"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginTop="80dp"
    android:onClick="Activity_Other_App"
    android:text="Other App"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.498"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/Table_layout" />
```

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

Activity_linear

Description: it contains 2 text views, edit text, button and imageView.

```
<?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:paddingLeft="16dp"
android:paddingRight="16dp"
android:orientation="vertical" >
<TextView
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:text="hiiii" />
<TextView
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:text="this is linear layout" />

<EditText
    android:layout_width="match_parent"
    android:layout_height="50dp"
    android:layout_weight="1"
    android:gravity="start"
    android:hint="you want to say something?????" />

<ImageView
    android:id="@+id/imageView"
    android:layout_width="match_parent"
    android:layout_height="500dp"
    app:srcCompat="@mipmap/ic_banner_foreground" />

<Button
    android:layout_width="150dp"
    android:layout_height="wrap_content"
    android:layout_gravity="right"
    android:text="send"
    app:icon="@android:drawable/ic_menu_send" />
</LinearLayout>
```

Activity_relative

Description: it contains 3 edit text, views, text views and button.

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:paddingLeft="16dp"
    android:paddingRight="16dp" >
```

```
<EditText
    android:id="@+id/name"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:hint="@string/reminder" />
```

```
<EditText
    android:id="@+id/dates"
    android:layout_width="0dp"
    android:layout_height="wrap_content"
    android:layout_below="@id/name"
    android:layout_alignParentStart="true"
    android:layout_toStartOf="@+id/times"
    android:editable="false"
    android:hint="Date"
    android:inputType="date"
    android:minHeight="48dp"
    android:onClick="calender" />
```

```
<EditText
    android:id="@id/times"
    android:layout_width="96dp"
    android:layout_height="wrap_content"
    android:layout_below="@id/name"
    android:layout_alignParentEnd="true"
    android:editable="false"
    android:hint="Time"
    android:inputType="time"
    android:minHeight="48dp"
    android:onClick="Time" />
```

```
<View
    android:id="@+id/divider"
    android:layout_width="match_parent"
    android:layout_height="1dp"
    android:layout_below="@+id/button"
    android:layout_alignParentStart="true"
    android:layout_alignParentEnd="true"
    android:layout_marginStart="16dp"
    android:layout_marginTop="24dp"
    android:layout_marginEnd="16dp"
    android:background="?android:attr/listDivider" />
```

```
<TextView
    android:id="@+id/reminder"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_below="@+id/divider"
    android:layout_alignParentStart="true"
    android:layout_alignParentEnd="true"
    android:layout_marginTop="54dp"
    android:text="@string/reminder"
    android:textAlignment="center" />
```

```
<Button
    android:id="@+id/button"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_below="@+id/times"
    android:layout_marginTop="16dp"
    android:layout_marginEnd="10dp"
    android:layout_alignParentEnd="true"
    android:onClick="Reminder"
    android:text="Button" />
```

```
</RelativeLayout>
```

Activity_search

Description: it contains edit text, views, text views and button.

```
<?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=".Search_Activity">
```

```
<EditText
    android:id="@+id/editTextURL"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginTop="150dp"
    android:ems="10"
    android:hint="URL"
```

```
android:inputType="textPersonName"
android:minHeight="48dp"
android:textAlignment="center"
app:layout_constraintEnd_toEndOf="parent"
app:layout_constraintHorizontal_bias="0.497"
app:layout_constraintStart_toStartOf="parent"
app:layout_constraintTop_toTopOf="parent" />
```

<Button

```
android:id="@+id/URLSubmit"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_marginTop="32dp"
android:text="Button"
android:onClick="search"
app:layout_constraintEnd_toEndOf="parent"
app:layout_constraintStart_toStartOf="parent"
app:layout_constraintTop_toBottomOf="@+id/editTextURL" />
```

<View

```
android:id="@+id/divider2"
android:layout_width="409dp"
android:layout_height="1dp"
android:layout_marginTop="68dp"
android:background="?android:attr/listDivider"
app:layout_constraintTop_toBottomOf="@+id/URLSubmit"
tools:layout_editor_absoluteX="1dp" />
```

<TextView

```
android:id="@+id/textView"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_marginTop="120dp"
android:text="TextView"
app:layout_constraintEnd_toEndOf="parent"
app:layout_constraintHorizontal_bias="0.498"
app:layout_constraintStart_toStartOf="parent"
app:layout_constraintTop_toBottomOf="@+id/divider2" />
```

</androidx.constraintlayout.widget.ConstraintLayout>

Activity_table

Description: this layout build with various tableRow and textViews and we represent the information at there.

```
<?xml version="1.0" encoding="utf-8"?>
<TableLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:stretchColumns="1">

    <TableRow>
        <TextView
            android:layout_column="1"
            android:text="Open..."
            android:padding="3dip" />
        <TextView
            android:text="Ctrl-O"
            android:gravity="right"
            android:padding="3dip" />
    </TableRow>

    <TableRow>
        <TextView
            android:layout_column="1"
            android:text="Save..."
            android:padding="3dip" />
        <TextView
            android:text="Ctrl-S"
            android:gravity="right"
            android:padding="3dip" />
    </TableRow>

    <TableRow>
        <TextView
            android:layout_column="1"
            android:text="Save As..."
            android:padding="3dip" />
        <TextView
            android:text="Ctrl-Shift-S"
            android:gravity="right"
            android:padding="3dip" />
    </TableRow>

    <View
        android:layout_height="2dip"
        android:background="#FF909090" />

    <TableRow>
```



```
<TextView
    android:text="X"
    android:padding="3dip" />
<TextView
    android:text="Import..."
    android:padding="3dip" />
</TableRow>

<TableRow>
    <TextView
        android:text="X"
        android:padding="3dip" />
    <TextView
        android:text="Export..."
        android:padding="3dip" />
    <TextView
        android:text="Ctrl-E"
        android:gravity="right"
        android:padding="3dip" />
</TableRow>

<View
    android:layout_height="2dip"
    android:background="#FF909090" />

<TableRow>
    <TextView
        android:layout_column="1"
        android:text="Quit"
        android:padding="3dip" />
</TableRow>
</TableLayout>
```

Activities

MainActivity.java

Description: In this file we can switch one layout to another layout package com.example.intent;

```
import androidx.appcompat.app.AppCompatActivity;

import android.content.Intent;
import android.os.Bundle;
import android.util.Log;
```

```
import android.view.View;

public class MainActivity extends AppCompatActivity {

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

    public void Activity_Linear_Layout(View view) {
        Log.d("Button", "Activity_Linear_Layout");
        Intent intent_linear = new Intent(MainActivity.this, Linear_Activity.class);
        startActivity(intent_linear);
    }

    public void Activity_Relative_Layout(View view) {
        Log.d("Button", "Activity_Relative_Layout");
        Intent intent_relative = new Intent(MainActivity.this, Relative_Activity.class);
        startActivity(intent_relative);
    }

    public void Activity_Table_Layout(View view) {
        Log.d("Button", "Activity_Table_Layout");
        Intent intent_table = new Intent(MainActivity.this, Table_Activity.class);
        startActivity(intent_table);
    }

    public void Activity_Other_App(View view) {
        String FN = "anand", LN = "patel";
        Log.d("Button", "Activity_Other_App");
        Intent intent_search = new Intent(MainActivity.this, Search_Activity.class);
        intent_search.putExtra("FN", FN);
        intent_search.putExtra("LN", LN);
        startActivity(intent_search);
    }
}
```

Linear_activity.java

Description: In this file we can have onCreate activity for the linear layout

```
package com.example.intent;

import androidx.appcompat.app.AppCompatActivity;
```

```
import android.os.Bundle;

public class Linear_Activity extends AppCompatActivity {

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

Relative_activity.java

Description: In this file we can have onCreate activity for the linear layout

```
package com.example.intent;

import androidx.appcompat.app.AppCompatActivity;

import android.annotation.SuppressLint;
import android.app.DatePickerDialog;
import android.app.TimePickerDialog;
import android.os.Bundle;
import android.util.Log;
import android.view.View;
import android.widget.DatePicker;
import android.widget.EditText;
import android.widget.TextView;
import android.widget.TimePicker;

import java.util.Calendar;

public class Relative_Activity extends AppCompatActivity {

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

    public void Reminder(View view) {
        Log.d("imgbtn", "Reminder: ");
        TextView t = findViewById(R.id.reminder);
        t.setText("Reminder Recorded");
    }
}
```

```
}
```

```
public void Time(View view) {
    int mHour, mMinute;
    EditText txtTime = findViewById(R.id.times);
    final Calendar c = Calendar.getInstance();
    mHour = c.get(Calendar.HOUR_OF_DAY);
    mMinute = c.get(Calendar.MINUTE);

    // Launch Time Picker Dialog
    TimePickerDialog timePickerDialog = new TimePickerDialog(this,
        new TimePickerDialog.OnTimeSetListener() {

            @Override
            public void onTimeSet(TimePicker view, int hourOfDay,
                int minute) {

                txtTime.setText(hourOfDay + ":" + minute);
            }
        }, mHour, mMinute, false);
    timePickerDialog.show();
}
```

```
public void calender(View view) {

    EditText txtDate = findViewById(R.id.dates);
    int mYear, mMonth, mDay;
    final Calendar c = Calendar.getInstance();
    mYear = c.get(Calendar.YEAR);
    mMonth = c.get(Calendar.MONTH);
    mDay = c.get(Calendar.DAY_OF_MONTH);

    DatePickerDialog datePickerDialog = new DatePickerDialog(this,
        new DatePickerDialog.OnDateSetListener() {

            @Override
            public void onDateSet(DatePicker view, int year,
                int monthOfYear, int dayOfMonth) {

                txtDate.setText(dayOfMonth + "-" + (monthOfYear + 1) + "-" + year);
            }
        }
    );
}
```

```
        }, mYear, mMonth, mDay);  
        datePickerDialog.show();  
    }  
}
```

Search_activity.java

```
package com.example.intent;  
  
import androidx.appcompat.app.AppCompatActivity;  
  
import android.annotation.SuppressLint;  
import android.content.Intent;  
import android.net.Uri;  
import android.os.Bundle;  
import android.view.View;  
import android.widget.EditText;  
import android.widget.TextView;  
  
public class Search_Activity extends AppCompatActivity {  
  
    @Override  
    protected void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
        setContentView(R.layout.activity_search);  
  
        TextView tv = findViewById(R.id.textView);  
        Bundle extras = getIntent().getExtras();  
        if(extras != null){  
            String v1 = extras.getString("FN");  
            String v2 = extras.getString("LN");  
  
            tv.setText(v1+"\n"+v2);  
        }  
    }  
  
    @SuppressLint("WrongViewCast")  
    public void search(View view) {  
        EditText getUrl;  
        getUrl = findViewById(R.id.editTextURL);  
        String url = getUrl.getText().toString();  
        Intent src = new Intent(Intent.ACTION_VIEW, Uri.parse(url));  
        startActivity(src);  
    }  
}
```

}

Table_activity.java

Descriptive: In this file we can have onCreate activity for the Table layout.

```
package com.example.intent;

import androidx.appcompat.app.AppCompatActivity;

import android.os.Bundle;

public class Table_Activity extends AppCompatActivity {

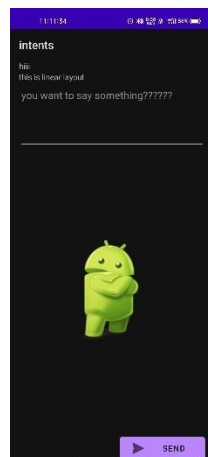
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_table);
    }
}
```

Output Screenshot:

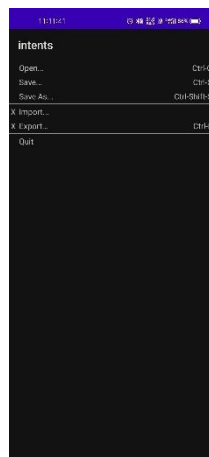
MainActivity()



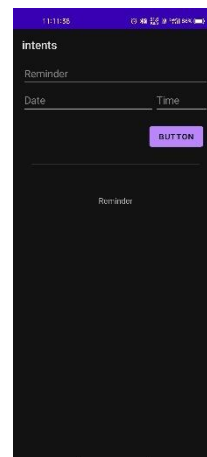
LinearLayout



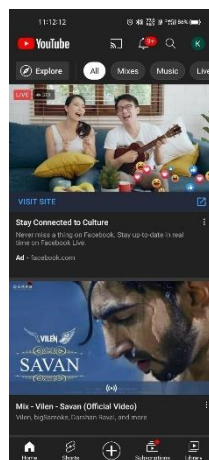
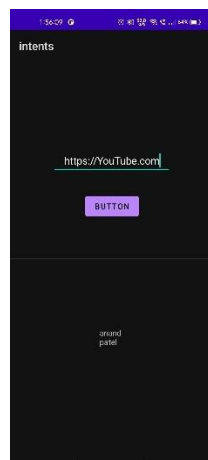
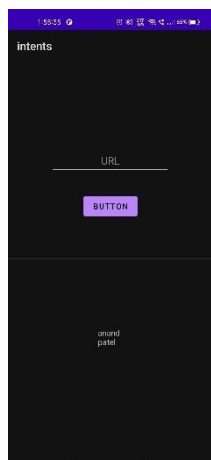
RelativeLayout



TableLayout



OtherApp



EXPERIMENT-8

Title: Using Android, Create a login Activity. It asks “username” and “password” from user. If username and password are valid, it displays Welcome message using new activity.

Components used to develop app:

<TextView/>: to print static text on the screen

<EditText/>: to take username, password, email, phone no. and country from the user

<Button/>: clickable component to submit the user entered values

CODE (LAYOUTS+ACTIVITIES):

Layouts:-

Activity_main.xml

Description: TextView, EditText and Buttons are majorly used components in any application. TextView is just static text on the screen. EditText is used for user input box and button is used to perform some activity or event while clicking happens on it. Different attributes of this components are used to give style and orientation to it..

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent">
    <RelativeLayout
        android:layout_width="match_parent"
        android:layout_centerInParent="true"
        android:id="@+id/centerLine"
        android:layout_height="wrap_content"/>
    <EditText
        android:id="@+id/userName"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_above="@id/centerLine"
        android:layout_marginLeft="15dp"
        android:layout_marginTop="20dp"
        android:layout_marginRight="15dp"
        android:hint="username"
    />
```



```
<TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="LOGIN"
    android:layout_above="@id/userName"
    android:layout_centerHorizontal="true"
    android:textSize="28sp"
/>
<EditText
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:hint="Password"
    android:inputType="textPassword"
    android:id="@+id/userPassword"
    android:layout_marginRight="15dp"
    android:layout_marginLeft="15dp"
    android:layout_marginTop="20dp"
    android:layout_below="@id/centerLine"/>
<Button
    android:id="@+id/btnLogin"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_below="@id/userPassword"
    android:layout_marginTop="24dp"
    android:layout_marginEnd="92dp"
    android:layout_toStartOf="@+id/btnReset"
    android:text="Login" />
<Button
    android:id="@+id/btnReset"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_below="@id/userPassword"
    android:layout_alignParentEnd="true"
    android:layout_marginTop="23dp"
    android:layout_marginEnd="81dp"
    android:text="RESET" />
</RelativeLayout>

activity_main2.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=".MainActivity2">
</androidx.constraintlayout.widget.ConstraintLayout>
```

Main_Activity.java

Description: onCreate() is the first callback method which will execute while loading the application. It contains business logic of the layout components.

```
package com.example.myapplication;

import android.app.Activity;
import android.content.Intent;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.Toast;
public class MainActivity extends Activity {
    EditText mEdmail,mEdPass;
    Button btn1;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        mEdmail = findViewById(R.id.userName);
        mEdPass = findViewById(R.id.userPassword);
        btn1 = findViewById(R.id.btnLogin);
        btn1.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                if(mEdmail.getText().toString().equals("Admin") &&
                    mEdPass.getText().toString().equals("123456")){
                    Intent i = new Intent(getApplicationContext(),MainActivity2.class);
                    Toast.makeText(getApplicationContext(),"WelCome",
                        Toast.LENGTH_SHORT).show();
                    startActivity(i);
                }else{
                    Toast.makeText(getApplicationContext(),"Wrong",
                        Toast.LENGTH_SHORT).show();
                }
            }
        });
    }
}
```

```
}  
}
```

Main_Activity2.java

```
package com.example.myapplication;  
  
import androidx.appcompat.app.AppCompatActivity;  
import android.os.Bundle;  
  
public class MainActivity2 extends AppCompatActivity {  
    @Override  
    protected void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
        setContentView(R.layout.activity_main2);  
    }  
}
```

AndroidManifest.xml

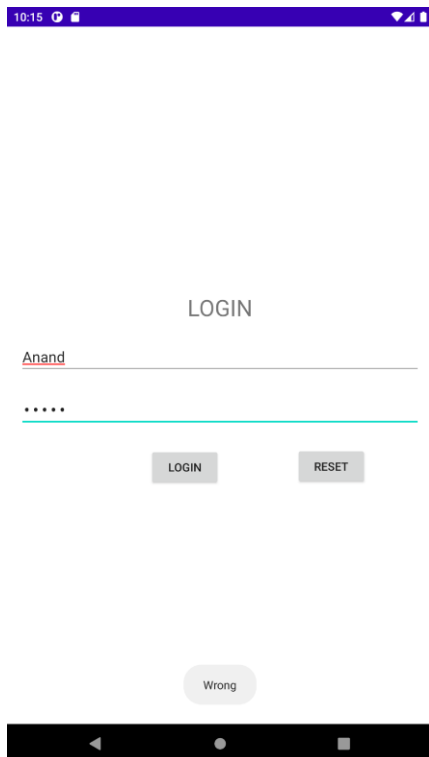
Description: Manifest file is used for configuration and context information about the android application like icon of application, name, theme and it will also contains intents of the application.

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

Output Screenshot:

Description: This main activity will show textview and edittext for username and password with one check box and submit button. Here all the components are aligned in center and spacing between them is equal which occupies entire screen.



EXPERIMENT-9

Title: “Happy Birthday” App using TextView and ImageView.

Components used to develop app:

<TextView/>: to print static text on the screen

<EditText/>: to take username, password, email, phone no. and country from the user

<ImageView/>: Displays image resources, for example Bitmap or Drawable resources

<Button/>: clickable component to submit the user entered values

CODE (LAYOUTS+ACTIVITIES):

Layouts:-

Activity_main.xml

Description: Here we take the name from user and then we wish them happy birthday with their name.

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".MainActivity">
    <LinearLayout
        android:layout_width="match_parent"
        android:id="@+id/l1"
        android:layout_height="wrap_content"
        android:layout_centerInParent="true"
        android:orientation="vertical" >
        <TextView
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="Whose birthday"
            android:layout_gravity="center_horizontal"
            android:textSize="26sp"
            android:textColor="@color/black"/>
        <EditText
            android:layout_width="match_parent"
            android:layout_marginRight="20dp"
```

```
        android:layout_marginLeft="20dp"
        android:id="@+id/name"
        android:layout_height="match_parent"/>
    <Button
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_gravity="center_horizontal"
        android:text="Click"
        android:layout_marginTop="15dp"
        android:id="@+id/btn1"/>
    </LinearLayout>
</RelativeLayout>
<?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"
    tools:context=".MainActivity2">
    <LinearLayout
        android:layout_width="match_parent"
        android:orientation="vertical"
        android:layout_gravity="center"
        android:layout_height="wrap_content"
        >

        <ImageView
            android:layout_width="122dp"
            android:layout_height="281dp"
            android:layout_gravity="center_horizontal"
            android:src="@drawable/img" />
        <TextView
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:id="@+id/tv1"
            android:textColor="@color/black"
            android:textSize="28sp"
            android:layout_marginTop="20dp"
            android:layout_gravity="center_horizontal"
            />
    </LinearLayout>
</LinearLayout>
```

Main_Activity.java

Description: In this file we take the input from the user and assign it to the text view.

```
package com.example.prac9;
import androidx.appcompat.app.AppCompatActivity;
import android.content.Intent;
import android.os.Bundle;
import android.util.Log;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
public class MainActivity extends AppCompatActivity {
    EditText ed1;
    Button btn1;
    String n;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        ed1 = findViewById(R.id.name);
        btn1 = findViewById(R.id.btn1);
        btn1.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                n = ed1.getText().toString();
                Intent i = new Intent(getApplicationContext(),MainActivity2.class);
                i.putExtra("name",n);
                Log.e("First name" , n);
                startActivity(i);
            }
        });
    }
}
```

Main_Activity2.java

```
package com.example.prac9;

import androidx.appcompat.app.AppCompatActivity;
import android.annotation.SuppressLint;
import android.content.Intent;
import android.os.Bundle;
import android.util.Log;
import android.widget.TextView;
public class MainActivity2 extends AppCompatActivity {
MAD (3170726)
```

```
TextView tv;  
@SuppressWarnings("SetTextI18n")  
@Override  
protected void onCreate(Bundle savedInstanceState) {  
    super.onCreate(savedInstanceState);  
    setContentView(R.layout.activity_main2);  
    tv = findViewById(R.id.tv1);  
    Intent i = getIntent();  
    String n = i.getStringExtra("name");  
    Log.e("name",n);  
    tv.setText("Happ B'day! " + n);  
}  
}
```

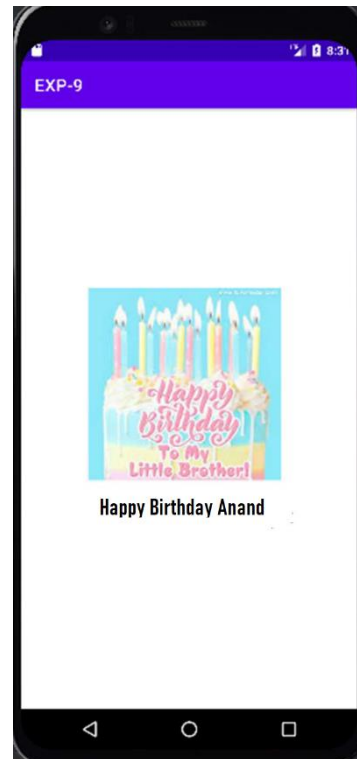
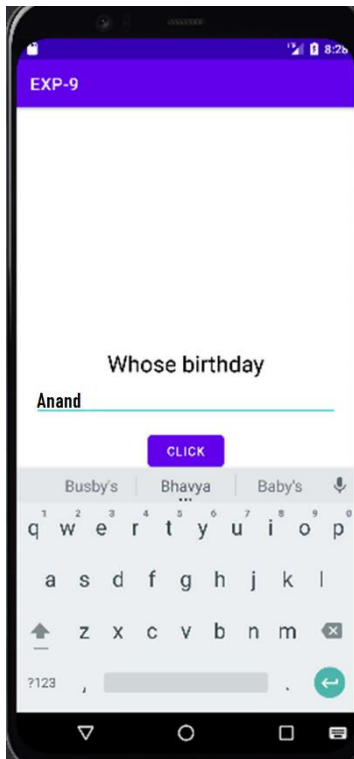
AndroidManifest.xml

Description: Manifest file is used for configuration and context information about the android application like icon of application, name, theme and it will also contains intents of the application.

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


Output Screenshot:

Description: Here in this app there are two screens. In the first screen, we take the input from the user and in the second screen we wish them happy birthday with their name.



EXPERIMENT-10

Title: Create “Hello Toast” App by implementing a click handler method for the button to display a message on the screen when the user clicks. Use Linear Layout for creating view.

Components used to develop app:

<TextView/>: to print static text on the screen

<EditText/>: to take username, password, email, phone no. and country from the user

<Button/>: clickable component to submit the user entered values

CODE (LAYOUTS+ACTIVITIES):

Layouts:-

Activity_main.xml

Description: In this app there are four button. Each button has its own toast message. When button pressed it display the toast message.

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".MainActivity">

    <Button
        android:id="@+id/button1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:background="#4CAF50"
        android:paddingStart="10dp"
        android:paddingEnd="10dp"
        android:text="Friend1"
        android:textColor="@android:color/background_light"
        android:textSize="24sp"
        app:layout_constraintBottom_toBottomOf="@id/button2"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent" />
```

```
<Button
    android:id="@+id/button2"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:background="#4CAF50"
    android:paddingStart="10dp"
    android:paddingEnd="10dp"
    android:text="Friend2"
    android:textColor="@android:color/background_light"
    android:textSize="24sp"
    app:layout_constraintBottom_toBottomOf="@id/button3"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="@id/button1" />
```

```
<Button
    android:id="@+id/button3"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:background="#4CAF50"
    android:paddingStart="10dp"
    android:paddingEnd="10dp"
    android:text="Friend3"
    android:textColor="@android:color/background_light"
    android:textSize="24sp"
    app:layout_constraintBottom_toBottomOf="@id/button4"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="@id/button2" />
```

```
<Button
    android:id="@+id/button4"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:background="#4CAF50"
    android:paddingStart="10dp"
    android:paddingEnd="10dp"
    android:text="Friend4"
    android:textColor="@android:color/background_light"
    android:textSize="24sp"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
```

```
app:layout_constraintTop_toTopOf="@id/button3" />  
</androidx.constraintlayout.widget.ConstraintLayout>
```

Main_Activity.java

Description: this file handles the toast messages. When a button pressed, which toast message is to be displayed is managed by this file.

```
package com.example.prac_10;  
  
import androidx.appcompat.app.AppCompatActivity;  
  
import android.os.Bundle;  
import android.view.View;  
import android.widget.Button;  
import android.widget.Toast;  
  
public class MainActivity extends AppCompatActivity {  
  
    Button btn1, btn2, btn3, btn4;  
  
    @Override  
    protected void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
        setContentView(R.layout.activity_main);  
  
        btn1 = findViewById(R.id.button1);  
        btn2 = findViewById(R.id.button2);  
        btn3 = findViewById(R.id.button3);  
        btn4 = findViewById(R.id.button4);  
  
        btn1.setOnClickListener(new View.OnClickListener() {  
            @Override  
            public void onClick(View view) {  
                Toast T = Toast.makeText(getApplicationContext(), "Friend 1 called",  
Toast.LENGTH_LONG);  
                T.show();  
            }  
        });  
        btn2.setOnClickListener(new View.OnClickListener() {  
            @Override  
            public void onClick(View view) {  
                Toast T = Toast.makeText(getApplicationContext(), "Friend 2 called",  
Toast.LENGTH_LONG);
```

```

        T.show();
    }
});
btn3.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        Toast T = Toast.makeText(getApplicationContext(), "Friend 3 called",
Toast.LENGTH_LONG);
        T.show();
    }
});
btn4.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        Toast T = Toast.makeText(getApplicationContext(), "Friend 4 called",
Toast.LENGTH_LONG);
        T.show();
    }
});
}
}

```

AndroidManifest.xml

Description: Manifest file is used for configuration and context information about the android application like icon of application, name, theme and it will also contains intents of the application.

```

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

    <application
        android:allowBackup="true"
        android:icon="@mipmap/ic_launcher"
        android:label="@string/app_name"
        android:roundIcon="@mipmap/ic_launcher_round"
        android:supportsRtl="true"
        android:theme="@style/Theme.LoginPage">
        <activity
            android:name=".MainActivity"
            android:exported="true">
            <intent-filter>

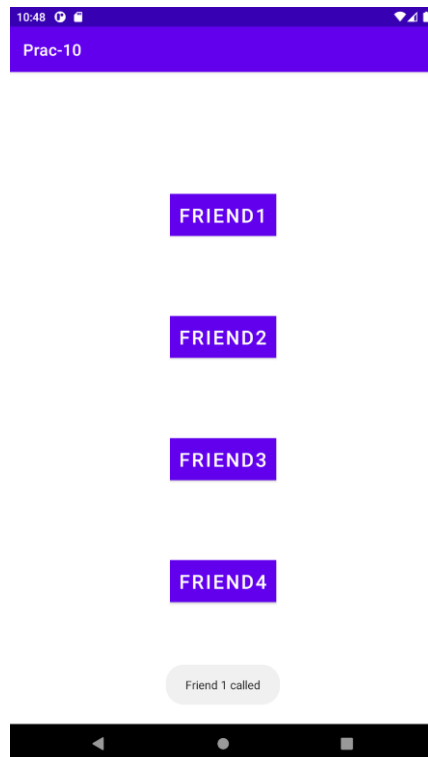
```

```
<action android:name="android.intent.action.MAIN" />
<category android:name="android.intent.category.LAUNCHER" />
</intent-filter>
</activity>
</application>

</manifest>
```

Output Screenshot:

Description: this is main activity. In this when a pressed any button its display the toast message assigned to it.



EXPERIMENT-11

Title: Create a Recycler View and list the details of student using following fields:

- 1. Name**
- 2. Address**
- 3. Photo (Image)**
- 4. Delete (Button Operation)**

Components used to develop app:

<TextView/>: to print static text on the screen

<EditText/>: to take username, password, email, phone no. and country from the user

<Button/>: clickable component to submit the user entered values

CODE (LAYOUTS+ACTIVITIES):

Layouts:-

Activity_main.xml

Description: In this there is text view, two edit text and a material button to enter the details from the user.

```
<?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">

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

        <TextView
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:textSize="26sp"
            android:textColor="@color/black"
```



```
android:layout_marginTop="20dp"
android:layout_gravity="center_horizontal"
android:text="Add Details"/>
```

```
<EditText
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_marginLeft="30dp"
    android:layout_marginTop="20dp"
    android:hint="Name"

    android:layout_marginRight="30dp"
    android:id="@+id/edName" />
```

```
<EditText
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_marginLeft="30dp"
    android:layout_marginTop="20dp"
    android:hint="Address"
    android:layout_marginRight="30dp"
    android:id="@+id/edAddress" />
```

```
<com.google.android.material.button.MaterialButton
    android:id="@+id/button"
    style="@style/Widget.MaterialComponents.Button.OutlinedButton"
    android:layout_width="wrap_content"
    android:layout_gravity="center_horizontal"
    android:layout_height="wrap_content"
    android:layout_marginTop="20dp"
    android:text="Pick Image and Add Data"
    android:textAllCaps="false"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.445"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    app:strokeColor="@color/purple_700" />
</LinearLayout>
```

```
<androidx.recyclerview.widget.RecyclerView
    android:layout_width="match_parent"
    android:layout_marginTop="15dp"
    android:layout_marginLeft="15dp"
    android:layout_marginRight="15dp"
```



```
        android:layout_height="match_parent"
        android:id="@+id/recycler"/>
</LinearLayout>
```

item_layout.xml :

Description: In this when a button pressed it store the information in the file and then it displays all the saved record to the user.

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_marginTop="10dp"
    android:clickable="false"
    android:layout_marginLeft="5dp"
    android:layout_marginRight="5dp"
    xmlns:app="http://schemas.android.com/apk/res-auto">

    <RelativeLayout
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:clickable="false"
        >
        <androidx.cardview.widget.CardView

            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:padding="10dp"
            android:id="@+id/cardview"
            android:clickable="false"

            android:background="#d3ee81"
            app:cardCornerRadius="6dp"
            android:elevation="30dp">

            <LinearLayout
                android:id="@+id/L1"
                android:background="@color/white"
                android:layout_width="match_parent"
                android:layout_height="wrap_content"
                android:clickable="false"
                android:orientation="vertical">
```

```
<RelativeLayout
    android:layout_width="match_parent"
    android:background="#f7f7f7"
    android:layout_height="wrap_content">

    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:orientation="horizontal">

        <ImageView android:id="@+id/img"
            android:layout_width="200dp"
            android:layout_height="150dp"
            android:src="@drawable/ic_launcher_background" />

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

            <LinearLayout
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:orientation="horizontal">

                <TextView
                    android:layout_width="wrap_content"
                    android:layout_height="wrap_content"
                    android:text="Name : "
                    android:textColor="@color/black"
                    android:textSize="20sp" />

                <TextView
                    android:id="@+id/txtName"
                    android:layout_width="wrap_content"
                    android:layout_height="wrap_content"
                    android:text="Name "
                    android:textColor="@color/black"
                    android:textSize="20sp" />
            </LinearLayout>
        </LinearLayout>
    </LinearLayout>
</RelativeLayout>
```



```
</LinearLayout>

<LinearLayout
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:orientation="horizontal">

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Address : "
        android:textColor="@color/black"
        android:textSize="20sp" />

    <TextView android:id="@+id/txtAddress"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Name "
        android:textColor="@color/black"
        android:textSize="20sp" />

</LinearLayout>

<LinearLayout
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    >
    <Button
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Delete"
        android:id="@+id/remvoeItem"/>
</LinearLayout>

</LinearLayout>
</LinearLayout>
</RelativeLayout>
</LinearLayout>
</androidx.cardview.widget.CardView>

</RelativeLayout>
</RelativeLayout>
```

MainActivity.java :

Description: this is main activity. In this it helps the user to choose a image file and it helps to store the user data in the array.

```
package com.example.prac_11; import

androidx.annotation.Nullable;
import androidx.appcompat.app.AppCompatActivity;
import androidx.core.app.ActivityCompat;
import androidx.recyclerview.widget.LinearLayoutManager;
import androidx.recyclerview.widget.RecyclerView;

import android.Manifest;
import android.annotation.SuppressLint;
import android.content.Intent;

import android.content.pm.PackageManager;
import android.net.Uri;
import android.os.Build; import
android.os.Bundle; import
android.util.Log; import
android.view.Display; import
android.view.View; import
android.widget.Button;
import android.widget.EditText;

import java.util.ArrayList;
import java.util.List;

public class MainActivity extends AppCompatActivity {
    Button mbtnAdd;
    RecyclerView r1;
    List<Model> models;
    Adapter adapter;
    EditText medName,medAdd;
    Button button;

    int SELECT_PICTURE = 200;
    ArrayList<Uri> list;
    String colum[]={
        Manifest.permission.WRITE_EXTERNAL_STORAGE,
        Manifest.permission.READ_EXTERNAL_STORAGE};

    @Override
```

```
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);
    mbtnAdd = findViewById(R.id.btnAdd);r1
    = findViewById(R.id.recycler);

    medName = findViewById(R.id.edName);
    medAdd = findViewById(R.id.edAddress);

    button = findViewById(R.id.button);

    models = new ArrayList<>();

    list=new ArrayList<>();

    //adapter=new Adapter(getApplicationContext(),models,list);
    r1.setLayoutManager(new LinearLayoutManager(getApplicationContext()));

    if((ActivityCompat.checkSelfPermission(
        this,column[0])!= PackageManager.PERMISSION_GRANTED)&&
        (ActivityCompat.checkSelfPermission(
            this,column[1])!= PackageManager.PERMISSION_GRANTED)){
        if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.M) {
            requestPermissions(column,123);
        }
    }

    button.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View view) {
            imageChooser();
        }
    });

    /* mbtnAdd.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View view) {

            //addStudent();
```

```
    }
  });*/
}
```

```
void imageChooser() {
```

```
    // create an instance of the
    // intent of the type image
    Intent i = new Intent();
    i.setType("image/*");
    i.setAction(Intent.ACTION_GET_CONTENT);

    // pass the constant to compare it
    // with the returned requestCode
    startActivityForResult(Intent.createChooser(i, "Select Picture"), SELECT_PICTURE);
}
```

```
@Override
```

```
public void onActivityResult(int requestCode, int resultCode, Intent data) {
    super.onActivityResult(requestCode, resultCode, data);
```

```
    if (resultCode == RESULT_OK) {
```

```
        // compare the resultCode with the
        // SELECT_PICTURE constant
        if (requestCode == SELECT_PICTURE) {
            // Get the url of the image from data Uri
            selectedImageUri = data.getData();if (null
            != selectedImageUri) {
                // update the preview image in the layout
                Log.e("H", "" + selectedImageUri);
                list.add(selectedImageUri);
                models.add(new
Model(medName.getText().toString(),medAdd.getText().toString()));
                adapter = new Adapter(getApplicationContext(), models, list);
                r1.setAdapter(adapter);
                adapter.notifyDataSetChanged();
                //adapter.notifyDataSetChanged();
                //IVPreviewImage.setImageURI(selectedImageUri);
            }
        }
    }
}
```

```
}

// @SuppressWarnings("NotifyDataSetChanged")
/* private void addStudent() {

    models.add(new Model(medName.getText().toString(),medAdd.getText().toString()));
    Log.e("name :",medName.getText().toString());
    adapter = new Adapter(getApplicationContext(),models);
    r1.setAdapter(adapter);
    adapter.notifyDataSetChanged();

}*/
}
```

Model.java :

Description: In this when a button pressed it retrieve the data from the array.

```
package com.example.prac_11;

public class Model {
    String name,address;

    public Model(String name, String address) {
        this.name = name;
        this.address = address;
    }

    public String getName() {
        return name;
    }

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

    public String getAddress() {
        return address;
    }

    public void setAddress(String address) {
        this.address = address;
    }
}
```

```
}
```

Adapter.java :

Description: it is the activity to display the information of the saved data. It find the element and assign value to it.

```
package com.example.prac_11;

import android.annotation.SuppressLint;
import android.content.Context;
import android.net.Uri;
import android.util.Log;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.Button; import
android.widget.ImageView;import
android.widget.TextView;

import androidx.annotation.NonNull;
import androidx.recyclerview.widget.RecyclerView;

import java.util.ArrayList;
import java.util.List;

public class Adapter extends RecyclerView.Adapter<Adapter.ViewHolder> {
    Context ctx;
    List<Model> models;
    private ArrayList list;

    public Adapter(Context ctx, List<Model> models, ArrayList list) {
        this.ctx = ctx; this.models
        = models;this.list = list;
    }

    @NonNull
    @Override
    public ViewHolder onCreateViewHolder(@NonNull ViewGroup parent, int viewType) {
        View v =
        LayoutInflater.from(parent.getContext()).inflate(R.layout.item_layout,parent,false);
        return new ViewHolder(v);
    }
```



```
@Override
public void onBindViewHolder(@NonNull ViewHolder holder,
@SuppressLint("RecyclerView") int position) {
    holder.mTvName.setText(models.get(position).getName());
    Log.e("Name : ",models.get(position).getName());
    holder.mTvAdd.setText(models.get(position).getAddress());
    holder.img.setImageURI((Uri) list.get(position));

    holder.btnRemove.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View view) {
            models.remove(position);
            list.remove(position);
            notifyDataSetChanged();
        }
    });

}

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

public class ViewHolder extends RecyclerView.ViewHolder {
    TextView mTvName,mTvAdd;
    ImageView img; Button
    btnRemove;
    public ViewHolder(@NonNull View itemView) {
        super(itemView);

        mTvName = itemView.findViewById(R.id.txtName);
        mTvAdd = itemView.findViewById(R.id.txtAddress);
        img=itemView.findViewById(R.id.img);
        btnRemove = itemView.findViewById(R.id.remvoeItem);

    }
}
```

AndroidManifest.xml :

Description: Manifest file is used for configuration and context information about the android application like icon of application, name, theme and it will also contains intents of the application.

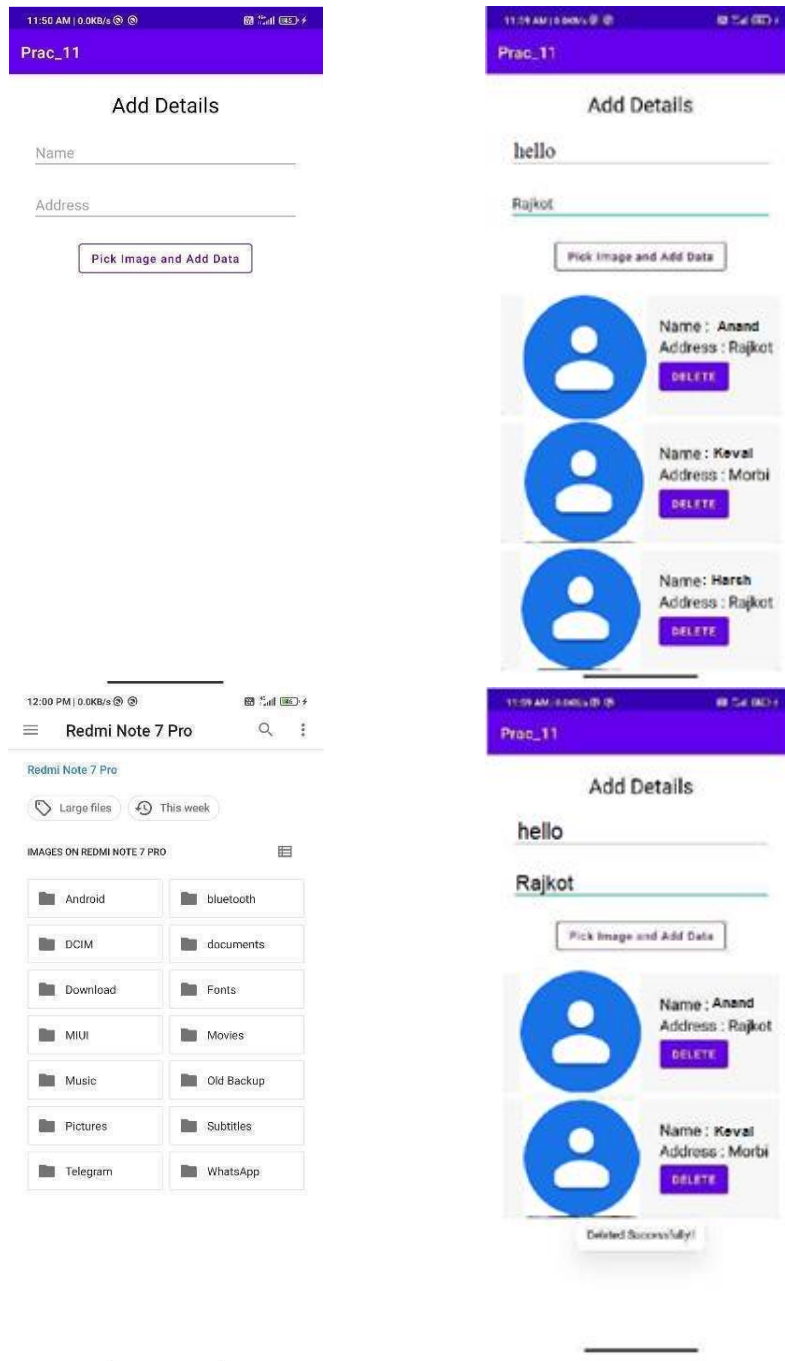
```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    package="com.example.prac_11">

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

    <application
        android:allowBackup="true"
        android:dataExtractionRules="@xml/data_extraction_rules"
        android:fullBackupContent="@xml/backup_rules"
        android:icon="@mipmap/ic_launcher"
        android:label="@string/app_name"
        android:roundIcon="@mipmap/ic_launcher_round"
        android:supportsRtl="true"
        android:theme="@style/Theme.Prac_11"
        tools:targetApi="31">
        <activity
            android:name=".MainActivity2"
            android:exported="false" />
        <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>
```

Output Screenshot:

Description: this is the output of the application. In this first we add name and address and then we pick the image and then it store the data into array and then it display all the stored data to the user.



EXPERIMENT-12

Title: Practical: Theme, Custom Styles, Drawables

Components used to develop app:

<TextView/>: to print static text on the screen

<EditText/>: to take username, password, email, phone no. and country from the user

<Button/>: clickable component to submit the user entered values

CODE (LAYOUTS+ACTIVITIES):

Layouts:-

Activity_main.xml

Description: In this app there are card view, progress bar, pinner, edittext and button.

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:id="@+id/mainrl"

    android:background="@drawable/gradient"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:android="http://schemas.android.com/apk/res/android">

    <RelativeLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:id="@+id/topLine"/>

    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:id="@+id/searchL"
        android:padding="5dp"
        android:layout_below="@id/topLine">

        <androidx.cardview.widget.CardView
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:layout_marginTop="10dp"
```



```
android:layout_margin="5dp"
android:elevation="30dp"
app:cardCornerRadius="10dp"
>
```

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

```
<EditText
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_marginRight="20dp"
    android:layout_marginTop="20dp"
    android:layout_marginLeft="20dp"
    android:hint="Enter Worker Type"
    android:padding="8dp"
    android:background="@drawable/edittext_rounded" >
```

```
</EditText>
```

```
<Spinner
    android:id="@+id/search_loc"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_marginRight="20dp"
    android:layout_marginTop="20dp"
    android:layout_marginLeft="20dp"
    android:padding="8dp"
    android:background="#f5f5f5"
    android:entries="@array/array_gujarat_districts"
/>
```

```
<Button
    android:id="@+id/btn_search"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_marginLeft="30dp"
    android:layout_marginTop="20dp"
    android:layout_marginRight="30dp"
    android:padding="10dp"
    android:layout_marginBottom="20dp"
    android:text="Search"
```

```
android:textAllCaps="true" />
```

```
</LinearLayout>
```

```
</androidx.cardview.widget.CardView>
```

```
</LinearLayout>
```

```
<TextView
```

```
    android:layout_width="match_parent"  
    android:layout_height="wrap_content"  
    android:id="@+id/text"  
    android:gravity="center"  
    android:textSize="20sp"  
    android:textStyle="bold"  
    android:layout_below="@id/searchL"  
    android:text="Best Rated Workers">
```

```
</TextView>
```

```
<androidx.recyclerview.widget.RecyclerView
```

```
    android:id="@+id/recycfsearch"  
    android:layout_width="match_parent"  
    android:layout_height="wrap_content"  
    android:layout_below="@id/text"  
    android:layout_marginTop="20dp"  
    android:fitsSystemWindows="true"  
    android:padding="10dp">
```

```
</androidx.recyclerview.widget.RecyclerView>
```

```
<RelativeLayout
```

```
    android:id="@+id/loading"  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:layout_centerInParent="true"  
    android:background="#f5f5f5"  
    android:padding="15dp"  
    android:visibility="invisible">
```

```
<ProgressBar
```

```
    android:id="@+id/prog"  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:layout_centerVertical="true"  
    android:visibility="invisible">
```

```
</ProgressBar>

<TextView
    android:id="@+id/tv"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_centerVertical="true"
    android:layout_marginLeft="10dp"
    android:layout_toRightOf="@id/prog"
    android:text="Loading.."
    android:textSize="20sp"
    android:visibility="invisible">

    </TextView>
</RelativeLayout>

</RelativeLayout>
```

edittext_rounded.xml :

```
<?xml version="1.0" encoding="utf-8"?>
<shape xmlns:android="http://schemas.android.com/apk/res/android">
    <solid android:color="#f9f9f9"> </solid>

    <corners android:radius="7dp"> </corners>
</shape>
```

themes.xml :

```
<resources xmlns:tools="http://schemas.android.com/tools">
    <!-- Base application theme. -->
    <style name="Theme.EServices"
        parent="Theme.MaterialComponents.DayNight.DarkActionBar">

        <!-- Custom Theme Color -->
        <item name="colorPrimary">@color/button</item>
        <item name="colorPrimaryVariant">@color/teal_700</item>
        <item name="colorOnPrimary">@color/black</item>

        <!-- Secondary brand color. -->
```

```

<item name="colorSecondary">@color/teal_200</item>
<item name="colorSecondaryVariant">@color/teal_700</item>
<item name="colorOnSecondary">@color/black</item>
<!-- Status bar color. -->
<item name="android:statusBarColor"
tools:targetApi="1">?attr/colorPrimaryVariant</item>
<!-- Customize your theme here. -->
</style>

<style name="Theme.EServices.NoActionBar">
    <item name="windowActionBar">false</item>
    <item name="windowNoTitle">true</item>
</style>

<style name="Theme.EServices.AppBarOverlay"
parent="ThemeOverlay.AppCompat.Dark.ActionBar" />

<style name="Theme.EServices.PopupOverlay"
parent="ThemeOverlay.AppCompat.Light" />

```

</resources>

colors.xml :

```

<?xml version="1.0" encoding="utf-8"?>
<resources>
    <color name="purple_200">#FFBB86FC</color>
    <color name="purple_500">#FF6200EE</color>
    <color name="purple_700">#FF3700B3</color>
    <color name="black">#FF000000</color>
    <color name="white">#FFFFFFFF</color>

    <color name="button"> #ff5f6d </color>
    <color name="teal_200">#FF03DAC5</color>
    <color name="teal_700">#FF018786</color>
</resources>

```

Main_Activity.java

```

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

AndroidManifest.xml

Description: Manifest file is used for configuration and context information about the android application like icon of application, name, theme and it will also contains intents of the application.

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

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

    <application
        android:allowBackup="true"
        android:dataExtractionRules="@xml/data_extraction_rules"
        android:fullBackupContent="@xml/backup_rules"
        android:icon="@mipmap/ic_launcher"
        android:label="@string/app_name"
        android:roundIcon="@mipmap/ic_launcher_round"
        android:supportsRtl="true"
        android:theme="@style/Theme.E-Services"
        tools:targetApi="31">

        <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>
```

Output Screenshot:



The screenshot shows a web application interface with a light blue and orange gradient background. At the top, there is a search form with two input fields: "Enter Worker Type" and "Select Your District". Below these fields is a red button labeled "SEARCH". Under the search form, there is a section titled "Best Rated Workers" which contains a list of items labeled "Item 0" through "Item 9".

Best Rated Workers	
Item 0	
Item 1	
Item 2	
Item 3	
Item 4	
Item 5	
Item 6	
Item 7	
Item 8	
Item 9	

EXPERIMENT-13

Title: Create the MP3 player like application with service.

Components used to develop app:

<TextView/>: to print static text on the screen

<EditText/>: to take username, password, email, phone no. and country from the user

<Button/>: clickable component to submit the user entered values

CODE (LAYOUTS+ACTIVITIES):

Layouts:-

Activity_main.xml

Description: In this app there are textview and listview.

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:background="@color/black"
    android:layout_height="match_parent"
    tools:context=".MainActivity">

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:textColor="@color/white"
        android:id="@+id/head"
        android:layout_marginTop="15dp"
        android:textSize="22dp"
        android:layout_centerHorizontal="true"
        android:text="Devarsh's Music Player" >

    </TextView>

    <View
        android:layout_width="match_parent"
        android:layout_height="2dp"
```

```
android:background="@color/white"
android:layout_below="@id/head"
android:layout_marginTop="10dp"
android:id="@+id/line">
```

```
</View>
```

```
<ListView
    android:id="@+id/ListView"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:layout_below="@id/line"
    android:layout_marginTop="15dp"
    android:divider="@android:color/transparent"
    android:dividerHeight="10sp"
    android:padding="9dp" />
```

```
</RelativeLayout>
```

song_name_layout.xml :

Description: this layout displays song name and icon.

```
<?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:background="@color/black"
    android:layout_height="wrap_content"
    android:orientation="vertical">

    <androidx.cardview.widget.CardView
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_margin="5dp"
        android:backgroundTint="@color/black"
        app:cardCornerRadius="10dp">

        <LinearLayout
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
```

```
        android:orientation="horizontal"
        android:padding="8dp">

        <ImageView
            android:layout_width="45dp"
            android:layout_height="45dp"
            android:background="#f3f3f3"
            android:outlineSpotShadowColor="@color/white"
            android:src="@drawable/music_icon"
            android:layout_marginRight="20dp"/>
        <TextView
            android:id="@+id/SongName"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="Song Name"
            android:layout_gravity="center_vertical"
            android:textColor="@color/white"

            android:textSize="18dp"
            android:singleLine="true"
            android:marqueeRepeatLimit="marquee_forever"
            android:ellipsize="marquee"
            android:scrollHorizontally="true" />

    </LinearLayout>
</androidx.cardview.widget.CardView>
</LinearLayout>
```

activity_player.xml :

Description: this is a activity which song Is playing and its progress bar.

```
<?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"
    android:weightSum="10"
    android:background="@drawable/gradiant"
    tools:context=".Activities.PlayerActivity"
    >
```



<LinearLayout

```
    android:layout_width="match_parent"
    android:layout_height="0dp"
    android:layout_weight="7"
    android:gravity="center"
    android:orientation="vertical">
```

<TextView

```
    android:id="@+id/SongTxt"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_margin="20dp"
    android:ellipsize="marquee"
    android:textColor="@color/white"
    android:marqueeRepeatLimit="marquee_forever"
    android:padding="10dp"
    android:singleLine="true"
    android:text="My Song Name"
    android:textAlignment="center"
    android:textSize="21dp" />
```

<ImageView

```
    android:id="@+id/MusicImage"
    android:layout_width="250dp"
    android:layout_height="250dp"
    android:layout_marginBottom="8dp"
    android:src="@drawable/music_icon" />
```

<RelativeLayout

```
    android:layout_width="match_parent"
    android:layout_height="68dp">
```

<SeekBar

```
    android:id="@+id/SeekBar"
    android:layout_width="250dp"
    android:layout_height="wrap_content"
    android:layout_alignParentBottom="true"
    android:layout_centerInParent="true"
    android:layout_marginStart="20dp"
    android:layout_marginTop="20dp"
    android:layout_marginEnd="20dp"
    android:progressBackgroundTint="@color/white"
    android:layout_marginBottom="20dp"
```



```
android:outlineAmbientShadowColor="@color/white"  
android:outlineSpotShadowColor="@color/white" />
```

```
<TextView  
    android:id="@+id/TxtSongStart"  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:layout_centerInParent="true"  
    android:layout_marginLeft="20dp"  
    android:layout_toLeftOf="@id/SeekBar"  
    android:singleLine="true"  
    android:text="0:00"  
    android:textColor="@color/white"  
    android:textSize="15dp" />
```

```
<TextView  
    android:id="@+id/TxtSongEnd"  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
  
    android:layout_alignParentRight="false"  
    android:layout_centerInParent="true"  
    android:layout_marginRight="20dp"  
    android:layout_toRightOf="@id/SeekBar"  
    android:singleLine="true"  
    android:text="0:00"  
    android:textColor="@color/white"  
    android:textSize="15dp" />
```

```
</RelativeLayout>
```

```
</LinearLayout>
```

```
<LinearLayout  
    android:layout_width="match_parent"  
    android:layout_height="0dp"  
    android:layout_weight="3">
```

```
<RelativeLayout  
    android:layout_width="match_parent"  
    android:layout_height="match_parent">
```

```
<Button
```



```
android:id="@+id/BtnPlay"  
android:layout_width="70dp"  
android:layout_height="70dp"  
android:layout_centerHorizontal="true"  
android:background="@drawable/pause_song_icon"/>
```

```
<Button  
    android:id="@+id/BtnNext"  
    android:layout_width="50dp"  
    android:layout_height="50dp"  
    android:layout_marginTop="15dp"  
    android:background="@drawable/next_song_icon"  
    android:layout_toRightOf="@id/BtnPlay"/>
```

```
<Button  
    android:id="@+id/BtnPrevious"  
    android:layout_width="50dp"  
    android:layout_height="50dp"  
    android:layout_marginTop="15dp"  
    android:background="@drawable/previews_icon"  
    android:layout_toLeftOf="@id/BtnPlay"/>
```

```
<Button  
    android:id="@+id/BtnFastForward"  
    android:layout_width="40dp"  
    android:layout_height="40dp"  
    android:layout_marginTop="20dp"  
    android:background="@drawable/forward_icon"  
    android:layout_toRightOf="@id/BtnNext"/>
```

```
<Button  
    android:id="@+id/BtnFastRewind"  
    android:layout_width="40dp"  
    android:layout_height="40dp"  
    android:layout_marginTop="20dp"  
    android:background="@drawable/rewind_icon"  
    android:layout_toLeftOf="@id/BtnPrevious"/>
```

```
<com.gauravk.audiovisualizer.visualizer.BarVisualize  
r xmlns:custom="http://schemas.android.com/apk/res-auto"  
    android:id="@+id/wave"  
    android:layout_width="match_parent"  
    android:layout_height="100dp"  
    android:layout_alignParentBottom="true"  
    custom:avColor="#FFFFFF"
```



```
custom:avDensity="0.8"  
android:outlineAmbientShadowColor="@color/white"  
android:outlineSpotShadowColor="@color/white"  
android:backgroundTint="@color/white"  
custom:avSpeed="normal"  
custom:avType="fill" />
```

```
</RelativeLayout>
```

```
</LinearLayout>
```

```
</LinearLayout>
```

Main_Activity.java

```
package com.dataflair.musicplayer;  
  
import androidx.annotation.NonNull;  
import androidx.appcompat.app.AppCompatActivity;  
import androidx.core.app.ActivityCompat;  
import  
androidx.core.content.ContextCompat;  
  
import  
androidx.recyclerview.widget.LinearLayoutManager;  
import androidx.recyclerview.widget.RecyclerView;  
  
import android.Manifest; import  
android.content.Context; import  
android.content.Intent;  
import  
android.content.pm.PackageManager; import  
android.database.Cursor;  
import android.media.MediaPlayer;  
import android.net.Uri;  
import android.os.Bundle;  
import android.os.Environment;  
import android.provider.MediaStore;  
import android.view.View;  
import android.view.ViewGroup;  
import android.widget.Adapter;  
import android.widget.AdapterView;  
import android.widget.BaseAdapter;  
import android.widget.Button;  
import android.widget.ImageView;
```

```
import android.widget.ListView;
import android.widget.TextView;

import
com.dataflair.musicplayer.Activities.PlayerActivity;import
com.karumi.dexter.Dexter;
import com.karumi.dexter.MultiplePermissionsReport;
import com.karumi.dexter.PermissionToken;
import
com.karumi.dexter.listener.PermissionDeniedResponse;
import
com.karumi.dexter.listener.PermissionGrantedResponse;
import com.karumi.dexter.listener.PermissionRequest;
import
com.karumi.dexter.listener.multi.MultiplePermissionsListener;
import com.karumi.dexter.listener.single.PermissionListener;
import org.jetbrains.annotations.NotNull;
import java.io.File;
import java.util.ArrayList;
import java.util.List;

public class MainActivity extends AppCompatActivity {
    ListView listView;
    String[] items;

    @Override
    protected void onCreate(Bundle savedInstanceState)
    { super.onCreate(savedInstanceState);
      setContentView(R.layout.activity_main);
      listView = (ListView) findViewById(R.id.ListView);
      runTimePermission();
    }

    public void runTimePermission() {
        Dexter.withContext(getApplicationContext())
            .withPermissions(Manifest.permission.READ_EXTERNAL_STORAGE,
Manifest.permission.RECORD_AUDIO)
            .withListener(new MultiplePermissionsListener()
                { @Override
                  public void
onPermissionsChecked(MultiplePermissionsReport
multiplePermissionsReport) {
                      displaySong();
                  }
                }
            )
    }
```

```
@Override
public void
onPermissionRationaleShouldBeShown(List<PermissionRequest>list,
PermissionToken permissionToken) {
```

```
    permissionToken.continuePermissionRequest();
}
}).check();
}
```

```
public ArrayList<File> findSong(File file) {
    //ArrayList to store all songs
    ArrayList<File> arrayList = new ArrayList<>();
    File[] files = file.listFiles();

    for (File singleFile : files) {
        //Adding the directory to arrayList if it is not hidden
        if (singleFile.isDirectory() && !singleFile.isHidden()) {

            arrayList.addAll(findSong(singleFile));

        } else {
            //Adding the single music file to ArrayList
            if (singleFile.getName().endsWith(".mp3")
            ||
            singleFile.getName().endsWith(".wav")) {
                arrayList.add(singleFile);
            }
        }
    }

    return arrayList;
}

public void displaySong() {
```

```
    final ArrayList<File> mySongs =
    findSong(Environment.getExternalStorageDirectory());
    items = new String[mySongs.size()];
```

```
//Adding all the music file without extensions to ArrayListfor (int i
= 0; i < mySongs.size(); i++) {
```

```
items[i] = mySongs.get(i).getName().toString().replace(".mp3", "").replace(".wav",
""");
}
//Calling the adapter and setting it to ListView
CustomAdapter customAdapter = new
CustomAdapter(); listView.setAdapter(customAdapter);

//Implementing onClickListener for ListView
listView.setOnItemClickListener(new AdapterView.OnItemClickListener() {
    @Override
    public void onItemClick(AdapterView<?> adapterView, View view, int i, long l) {

        String songName = (String) listView.getItemAtPosition(i);

        //Calling the next intent and sending the Required Details to play the songs
        Intent intent = new Intent(getApplicationContext(), PlayerActivity.class);
        intent.putExtra("songs", mySongs);
        intent.putExtra("songname", songName);
        intent.putExtra("pos", i);
        startActivity(intent);
    }
});
}
```

```
class CustomAdapter extends BaseAdapter {
```

```
    @Override
    public int getCount() {
        //Returning the count of total songs in an
        ArrayListreturn items.length;
    }
```

```
    @Override
    public Object getItem(int position) {
        return null;
    }
```

```
    @Override
    public long getItemId(int position) {
        return 0;
    }
```

```
    @Override
    public View getView(int position, View convertView, ViewGroup parent) {
```

```
//Inflating all the single music files in a Layout File
View view = getLayoutInflater().inflate(R.layout.song_name_layout, null);
TextView txtSong = view.findViewById(R.id.SongName);
txtSong.setSelected(true);
txtSong.setText(items[position]);
return view;
    }
}
```

PlayerActivity.java :

Description: this file is used to play songs.

```
package com.dataflair.musicplayer.Activities;

import

androidx.appcompat.app.AppCompatActivity;

import android.animation.AnimatorSet;
import
android.animation.ObjectAnimator;

import android.content.Intent;
import android.media.MediaPlayer;
import android.net.Uri;
import android.os.Bundle;
import android.os.Handler;
import android.view.View;
import android.view.animation.AccelerateInterpolator;
import android.view.animation.Animation;
import android.view.animation.AnimationSet;
import
android.view.animation.TranslateAnimation;
import android.widget.Button;
import android.widget.ImageView;
import android.widget.SeekBar;
import android.widget.TextView;
import android.widget.Toast;
import com.dataflair.musicplayer.R;
import com.gauravk.audiovisualizer.visualizer.BarVisualizer;
```

```
import com.gauravk.audiovisualizer.visualizer.BlastVisualizer;
import
com.gauravk.audiovisualizer.visualizer.CircleLineVisualizer;
import java.io.File;
import java.util.ArrayList;

public class PlayerActivity extends AppCompatActivity {

    Button btnPlay, btnNext, btnPrevious, btnFastForward, btnFastBackWard;
    TextView txtSongName, txtSongStart, txtSongEnd;
    SeekBar seekMusicBar;
    BarVisualizer barVisualizer;
    ImageView imageView;

    String songName;
    public static final String EXTRA_NAME =
"song_name";static MediaPlayer mediaPlayer;
    int position;

    ArrayList<File>mySongs;

    Thread updateSeekBar;

    @Override
    protected void onCreate(Bundle savedInstanceState) {

        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_player);

        //Assigning the address of the android Materials
        btnPlay = (Button) findViewById(R.id.BtnPlay);
        btnNext = (Button)
        findViewById(R.id.BtnNext);
        btnPrevious = (Button) findViewById(R.id.BtnPrevious);
        btnFastForward = (Button) findViewById(R.id.BtnFastForward);
        btnFastBackWard = (Button) findViewById(R.id.BtnFastRewind);

        txtSongName = (TextView) findViewById(R.id.SongTxt);
        txtSongStart = (TextView)
        findViewById(R.id.TxtSongStart);txtSongEnd =
        (TextView) findViewById(R.id.TxtSongEnd);

        seekMusicBar = (SeekBar)
        findViewById(R.id.SeekBar);barVisualizer =
```



```
findViewById(R.id.wave);

imageView = (ImageView) findViewById(R.id.MusicImage);

//Checking if any song playing or not
if (mediaPlayer != null) {

    //we will start mediaPlayer if currently there is no songs in it
    mediaPlayer.start();
    mediaPlayer.release();
}

//Getting the Required Details from the past
Intent intent = getIntent();
Bundle bundle = intent.getExtras();

mySongs = (ArrayList)
bundle.getIntegerArrayList("songs");String sName =
intent.getStringExtra("songname");
position = bundle.getInt("pos");
txtSongName.setSelected(true);

//Extracting the fileName form the ArrayList
Uri uri = Uri.parse(mySongs.get(position).toString());
songName = mySongs.get(position).getName();
txtSongName.setText(songName);

//passing the song path to the Media Player
mediaPlayer = MediaPlayer.create(getApplicationContext(), uri);
mediaPlayer.start();

//Method to get the current media end
timesongEndTime();

//Method to show bar visualizers
visualizer();

//Thread to update the seekBar while playing
songupdateSeekBar = new Thread() {
    @Override
    public void run() {

        int TotalDuration = mediaPlayer.getDuration();
```



```
int CurrentPosition = 0;

while (CurrentPosition < TotalDuration) {
    try {
        sleep(500);
        CurrentPosition =
            mediaPlayer.getCurrentPosition();
        seekMusicBar.setProgress(CurrentPosition);
    } catch (InterruptedException | IllegalStateException e) {

        e.printStackTrace();
    }
}

};

//Setting the seekbar's max progress to the maximum duration of the media file
seekMusicBar.setMax(mediaPlayer.getDuration());
updateSeekBar.start();

//Setting the Music player from the position of the seekbar
seekMusicBar.setOnSeekBarChangeListener(new SeekBar.OnSeekBarChangeListener()
{
    @Override
    public void onProgressChanged(SeekBar seekBar, int i, boolean b) {
    }
    @Override
    public void onStartTrackingTouch(SeekBar seekBar) {
    }
    @Override
    public void onStopTrackingTouch(SeekBar seekBar) {

        //getting the progress of the seek bar and setting it to Media Player
        mediaPlayer.seekTo(seekBar.getProgress());
    }
});

//Creating the Handler to update the current duration
final Handler handler = new Handler();
final int delay = 1000;
handler.postDelayed(new Runnable() {
    @Override
```




```
public void run() {
    //Getting the current duration from the media player
    String currentTime = createDration(mediaPlayer.getCurrentPosition());
    //Setting the current duration in textView
    txtSongStart.setText(currentTime);
    handler.postDelayed(this, delay);
}
}, delay);

//Implementing OnClicListener for Play and Pause Button
btnPlay.setOnClickListener(new View.OnClickListener()
{
    @Override
    public void onClick(View view) {
        //Checking playing any songs or not
        if (mediaPlayer.isPlaying()) {
            //setting the play icon
            btnPlay.setBackgroundResource(R.drawable.play_song_icon);

            //Pausing the current media
            mediaPlayer.pause();

        } else {

            //Setting the pause icon
            btnPlay.setBackgroundResource(R.drawable.pause_song_icon);

            //Starting the media player
            mediaPlayer.start();

            //Creating the Animation
            TranslateAnimation moveAnim = new TranslateAnimation(-25, 25, -25, 25);
            moveAnim.setInterpolator(new AccelerateInterpolator());
            moveAnim.setDuration(600);
            moveAnim.setFillEnabled(true);
            moveAnim.setFillAfter(true);
            moveAnim.setRepeatMode(Animation.REVERSE);
            moveAnim.setRepeatCount(1);

            //Setting the Animation for the Image
            imageView.startAnimation(moveAnim);

            //Calling the BarVisualizer
```

```
        visualizer();
    }
}
});

//Performing the Button Click Operation after the completion of song
mediaPlayer.setOnCompletionListener(new MediaPlayer.OnCompletionListener() {
    @Override
    public void onCompletion(MediaPlayer mediaPlayer) {
        btnNext.performClick();
    }
});

//Implementing OnclickListener
btnNext.setOnClickListener(new View.OnClickListener()
{
    @Override
    public void onClick(View view) {

        //Stopping the currently playing media
        mediaPlayer.stop();
        mediaPlayer.release();

        //Getting the Current media position and incrementing it by 1
        position = ((position + 1) % mySongs.size());

        //Extracting the media path form the array List
        Uri uri1 = Uri.parse(mySongs.get(position).toString());

        //Setting the path to the media player
        mediaPlayer = MediaPlayer.create(getApplicationContext(), uri1);

        //Getting the current song Name and setting it to TextView
        songName = mySongs.get(position).getName();
        txtSongName.setText(songName);

        //Starting the Media Player
        mediaPlayer.start();

        //Extracting the duration of the song
        songEndTime();
```



```
//Animating the ImageView  
startAnimation(imageView, 360f);  
visualizer();  
}  
});
```

```
//Implementing the OnClick Listener  
btnPrevious.setOnClickListener(new View.OnClickListener()  
{  
    @Override  
    public void onClick(View view) {  
        //Stoping the media Player  
        mediaPlayer.stop();  
        mediaPlayer.release();  
  
        //getting the current media position and decrementing it by one  
        position = ((position - 1) % mySongs.size());  
        if (position < 0)  
            position = mySongs.size() - 1;  
  
        //Extracting the media path  
        Uri uri1 = Uri.parse(mySongs.get(position).toString());  
  
        //Setting the media path to the media player  
        mediaPlayer = MediaPlayer.create(getApplicationContext(),  
            uri1); songName = mySongs.get(position).getName();  
        txtSongName.setText(songName);  
        mediaPlayer.start();  
        songEndTime();  
  
        //Animating the imageView  
        startAnimation(imageView, -360f);  
        visualizer();  
    }  
});
```

```
//Implementing the fastForward  
btnFastForward.setOnClickListener(new View.OnClickListener() {  
    @Override  
    public void onClick(View view) {  
        if (mediaPlayer.isPlaying()) {
```



```
//Getting the current position and adding 10sec to it
mediaPlayer.seekTo(mediaPlayer.getCurrentPosition() +
10000);

}
}
});

//Implementing the FastBackWard
btnFastBackWard.setOnClickListener(new
View.OnClickListener() {
    @Override
    public void onClick(View view) {
        if (mediaPlayer.isPlaying()) {

            //Getting the curent Position of the song and decrease 10sec from it
            mediaPlayer.seekTo(mediaPlayer.getCurrentPosition() - 10000);

        }
    }
});

}

//Method to create animation for imageView
public void startAnimation(View view, Float degree) {
    ObjectAnimator objectAnimator = ObjectAnimator.ofFloat(imageView,
"rotation", 0f,degree);
    objectAnimator.setDuration(1000);

    AnimatorSet animatorSet = new AnimatorSet();
    animatorSet.playTogether(objectAnimator);
    animatorSet.start();

}

//Preparing the Time format for setting to textView
public String createDuration(int duration) {

    String time = "";
    int min = duration / 1000 / 60;
    int sec = duration / 1000 % 60;
    time = time + min + ":";
```

```
if (sec < 10) {

    time += "0";

}
time += sec;
return time;
}

//Setting the Visualizer
public void visualizer() {

    //Extracting and Setting the current media id to the
    Visualizerint audioSessionId =
    mediaPlayer.getAudioSessionId();
    if (audioSessionId != -1) {
        barVisualizer.setAudioSessionId(audioSessionId);
    }
}

//Method To extract the duration of the current media and setting it to
TextViewpublic void songEndTime() {
    String endTime =
    createDuration(mediaPlayer.getDuration());
    txtSongEnd.setText(endTime);
}

//Releasing the BarVisualizer on Closing the Activity
@Override
protected void onDestroy() {
    super.onDestroy();
    if (barVisualizer != null)
        barVisualizer.release();
}
}
```

AndroidManifest.xml

Description: Manifest file is used for configuration and context information about the android application like icon of application, name, theme and it will also contains intents of the application.

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

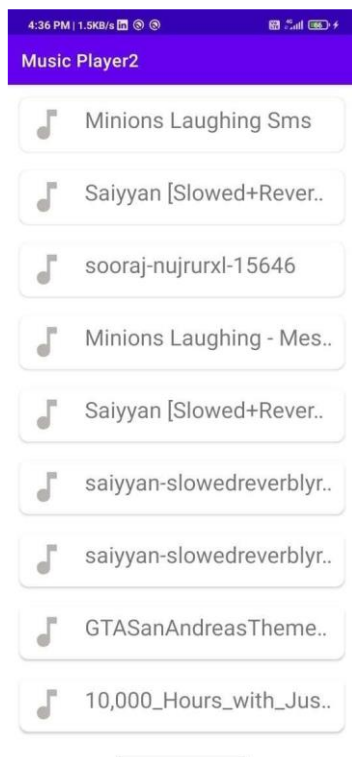
```
package="com.example.loginpage">
```

```
<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.LoginPage">
    <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>
```

Output Screenshot:

Description: this is the screenshot of music player.



EXPERIMENT-14

Title: The Simple Calculator app has two edit texts and four buttons. When you enter two numbers and click a button, the app performs the calculation for that button and displays the result.

Components used to develop app:

<TextView/>: to print static text on the screen

<EditText/>: to take username, password, email, phone no. and country from the user

<Button/>: clickable component to submit the user entered values

CODE (LAYOUTS+ACTIVITIES):

Layouts:-

Activity_main.xml

Description: In this app there are four buttons and two text view. When we enter 2 number and click on button then it display the result assigned to the button.

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

    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content">

        <TextView
            android:layout_width="wrap_content"
            android:layout_height="match_parent"
            android:text="Number1: "
            />

        <EditText
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:id="@+id/edt1"
            />
```



```
</LinearLayout>
```

```
<LinearLayout  
    android:layout_width="match_parent"  
    android:layout_height="wrap_content">
```

```
    <TextView  
        android:layout_width="wrap_content"  
        android:layout_height="match_parent"  
        android:text="Number2: "  
    />
```

```
    <EditText  
        android:id="@+id/edt2"  
        android:layout_width="match_parent"  
        android:layout_height="wrap_content" />
```

```
</LinearLayout>
```

```
<LinearLayout  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:layout_gravity="center_horizontal">
```

```
<Button  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:text="+"  
    android:id="@+id/plus"></Button>  
<Button  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:text="-"  
    android:id="@+id/sub"></Button>  
<Button  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:text="*"  
    android:id="@+id/mul"></Button>  
<Button  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:text="/"
```

```
        android:id="@+id/div"></Button>
    <Button
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text=""></Button>

</LinearLayout>

<TextView
    android:id="@+id/tv1"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Result: "></TextView>

<TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
/>

</LinearLayout>
```

Main_Activity.java

Description: this file take the input from text view and done the calculation when button clicked and get back the output to the user.

```
package com.example.calculator;
import androidx.appcompat.app.AppCompatActivity;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.TextView;
public class MainActivity extends AppCompatActivity {
    Button add1,sub1,mul1,div1;
    Float number1,number2;
    TextView result;
    EditText edittext1,edittext2;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        add1=findViewById(R.id.plus);
        sub1=findViewById(R.id.sub);
        mul1=findViewById(R.id.mul);
```

```
div1=findViewById(R.id.div);
edittext1=findViewById(R.id.edt1);
edittext2=findViewById(R.id.edt2);
result=findViewById(R.id.tv1);
add1.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        number1=Float.parseFloat(edittext1.getText().toString());
        number2=Float.parseFloat(edittext2.getText().toString());
        result.setText(number1+number2+"");
        edittext1.getText().clear();
        edittext2.getText().clear();
    }
});
sub1.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        number1=Float.parseFloat(edittext1.getText().toString());
        number2=Float.parseFloat(edittext2.getText().toString());
        result.setText(number1-number2+"");
        edittext1.getText().clear();
        edittext2.getText().clear();
    }
});
mul1.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        number1=Float.parseFloat(edittext1.getText().toString());
        number2=Float.parseFloat(edittext2.getText().toString());
        result.setText(number1*number2+"");
        edittext1.getText().clear();
        edittext2.getText().clear();
    }
});
div1.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        number1=Float.parseFloat(edittext1.getText().toString());
        number2=Float.parseFloat(edittext2.getText().toString());
        result.setText(number1/number2+"");
        edittext1.getText().clear();
        edittext2.getText().clear();
    }
});
```

```
}  
}
```

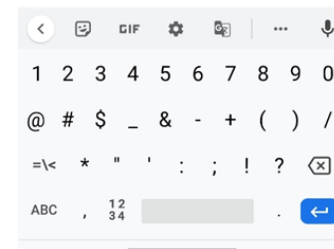
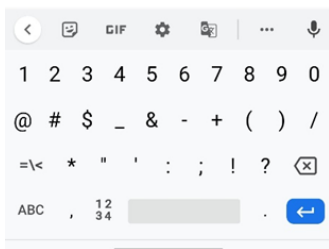
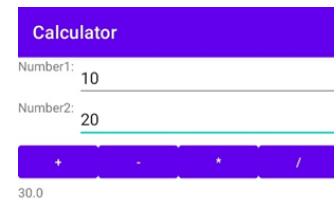
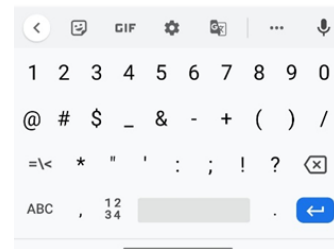
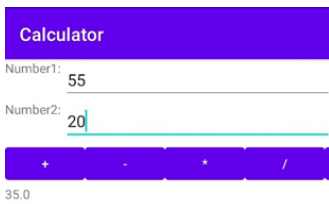
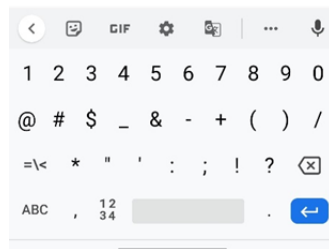
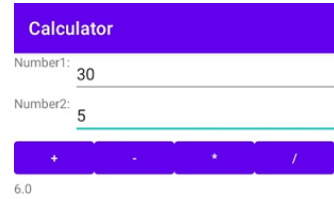
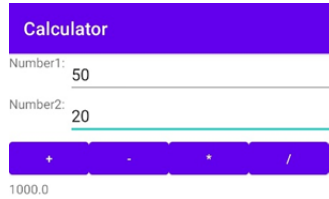
AndroidManifest.xml

Description: Manifest file is used for configuration and context information about the android application like icon of application, name, theme and it will also contains intents of the application.

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

Output Screenshot:

Description: this is main activity. In this it takes two number from user and when a pressed any button its display the calculations of the user given number.



EXPERIMENT-15

Title: Develop one App. Which Contains Specific User Interface and design Interface.

Components used to develop app:

<TextView/>: to print static text on the screen

<EditText/>: to take username, password, email, phone no. and country from the user

<Button/>: clickable component to submit the user entered values

CODE (LAYOUTS+ACTIVITIES):

Layouts:-

Activity_main.xml

Description: this is main layout file which has user interface.

```
<?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"
    android:background="#F6F5F5"
    >

    <RelativeLayout android:layout_width="match_parent"
        android:layout_height="350dp">

        <LinearLayout android:layout_width="match_parent"
            android:layout_height="300dp"
            android:background="#2196F3"
            android:orientation="vertical">

            <androidx.cardview.widget.CardView android:layout_width="140dp"
                android:layout_height="140dp"
                android:layout_gravity="center_horizontal"
                android:layout_marginTop="35dp"
                android:foregroundGravity="center_horizontal"
                app:cardCornerRadius="100dp">

                <ImageView android:layout_width="match_parent"
```



```
        android:layout_height="match_parent"
        android:src="@drawable/profile" />
</androidx.cardview.widget.CardView>

<TextView android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_gravity="center_horizontal"
    android:layout_marginTop="10dp" android:text="Shelvi
    Kamani"
```

```
android:textColor="#fff"
    android:textSize="23dp"
    android:textStyle="bold" />
</LinearLayout>
```

```
<androidx.cardview.widget.CardView
    android:layout_width="match_parent"
    android:layout_height="120dp"
    android:layout_alignParentBottom="true"
    android:layout_marginLeft="20dp"
    android:layout_marginRight="20dp"
    android:layout_centerHorizontal="true"
    app:cardCornerRadius="5dp">
```

```
<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:weightSum="3"
    android:orientation="horizontal"
    android:padding="10dp">
```

```
<LinearLayout
    android:layout_width="0dp"
    android:layout_height="match_parent"
    android:layout_weight="1"
    android:orientation="vertical"
    android:gravity="center">
```

```
<TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Subscribers"
    android:textSize="17dp"
```



```
        android:textColor="#000"/>

        <TextView
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="1000K+"
            android:textSize="27dp"
            android:textColor="#2196F3"
            android:textStyle="bold"/>

    </LinearLayout>
    <LinearLayout
        android:layout_width="0dp"
        android:layout_height="match_parent"
        android:layout_weight="1"
        android:orientation="vertical"
        android:gravity="center">

        android:layout_height="wrap_content"
        android:layout_centerHorizontal="true"
        android:text="Loading"
        android:textColor="@color/black"
        android:textSize="24sp"
        android:textStyle="bold">

        </TextView>

    <TextView android:id="@+id/postTv"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_below="@id/postCount"
        android:layout_centerHorizontal="true"
        android:layout_marginLeft="15dp"
        android:text="Post"
        android:textColor="@color/black"
        android:textSize="20sp">

        </TextView>
    </RelativeLayout>

    <RelativeLayout
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignParentRight="true"
        android:layout_marginTop="20dp"
```



```
android:layout_marginBottom="20dp"  
android:padding="10dp">
```

```
<TextView android:id="@+id/likeCount"  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:layout_centerHorizontal="true"  
    android:text="Loading"  
    android:textColor="@color/black"  
    android:textSize="24sp"  
    android:textStyle="bold">
```

```
</TextView>
```

```
<TextView android:id="@+id/likeTv"  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:layout_below="@id/likeCount"  
    android:layout_centerHorizontal="true"  
    android:layout_marginLeft="15dp"  
    android:text="Total Likes"  
    android:textColor="@color/black"  
    android:textSize="20sp">
```

```
</TextView>
```

```
</RelativeLayout>
```

```
</RelativeLayout>
```

```
</com.google.android.material.card.MaterialCardView>
```

```
<Button
```

```
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:text="Follow me"  
    android:layout_gravity="center_horizontal"  
    android:textSize="18dp"  
    android:layout_marginTop="40dp"/>
```

```
</LinearLayout>
```

Main_Activity.java

Description: this is main activity used to crate and display activity main layout.

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

AndroidManifest.xml

Description: Manifest file is used for configuration and context information about the android application like icon of application, name, theme and it will also contains intents of the application.

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

    <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.LoginPage">
        <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>
```

Output Screenshot:



EXPERIMENT-16

Title: Using experiment 4 and Save user data in a database.

Components used to develop app:

<TextView/>: to print static text on the screen

<EditText/>: to take username, password, email, phone no. and country from the user

<Button/>: clickable component to submit the user entered values

CODE (LAYOUTS+ACTIVITIES):

Layouts:-

Activity_main.xml

Description: In this app there are five edit text and one button.

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="@drawable/reg"
    xmlns:android="http://schemas.android.com/apk/res/android">

    <LinearLayout
        xmlns:android="http://schemas.android.com/apk/res/androi
        d" xmlns:app="http://schemas.android.com/apk/res-auto"
        xmlns:tools="http://schemas.android.com/tools"
        android:layout_gravity="center_vertical"
        android:layout_marginLeft="20dp"
        android:layout_marginRight="20dp"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_centerInParent="true"
        android:orientation="vertical"
        android:alpha="0.8"
        android:background="@drawable/rounded"
        tools:context=".MainActivity">

        <TextView
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
```



```
        android:textSize="30dp"
        android:layout_gravity="center_horizontal"
        android:layout_margin="20dp"
        android:text="Registration"/>

<com.google.android.material.textfield.TextInputLayout
    style="@style/Widget.MaterialComponents.TextInputLayout.OutlinedBox.Dense"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_marginBottom="12dp"
    android:layout_marginStart="20dp"
    android:layout_marginEnd="20dp"
    app:boxStrokeColor="@color/newpurple"
    app:hintTextColor="@color/newpurple"
    android:hint="Enter User Name">

    <com.google.android.material.textfield.TextInputEditTe
        xt android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:cursorVisible="true"
        android:singleLine="true"
        android:textCursorDrawable="@drawable/black_curson"
        android:id="@+id/ed_uname"/>

</com.google.android.material.textfield.TextInputLayout>

<com.google.android.material.textfield.TextInputLayout
    style="@style/Widget.MaterialComponents.TextInputLayout.OutlinedBox.Dense"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_marginBottom="12dp"
    android:layout_marginStart="20dp"
    android:layout_marginEnd="20dp"

    app:boxStrokeColor="@color/newpurple"
    app:hintTextColor="@color/newpurple"
    android:hint="Enter Mobile Number">

    <com.google.android.material.textfield.TextInputEditText
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:singleLine="true"
```



```
android:textCursorDrawable="@drawable/black_cursor"  
android:id="@+id/ed_mobile_number"/>
```

```
</com.google.android.material.textfield.TextInputLayout>
```

```
<com.google.android.material.textfield.TextInputLayout  
    style="@style/Widget.MaterialComponents.TextInputLayout.OutlinedBox.Dense"  
    android:layout_width="match_parent"  
    android:layout_height="wrap_content"  
    android:layout_marginBottom="12dp"  
    android:layout_marginStart="20dp"  
    android:layout_marginEnd="20dp"  
    app:boxStrokeColor="@color/newpurple"  
    app:hintTextColor="@color/newpurple"  
    android:hint="Enter Password"
```

```
    app:endIconMode="password_toggle">
```

```
<com.google.android.material.textfield.TextInputEditText  
    android:layout_width="match_parent"  
    android:layout_height="wrap_content"  
    android:singleLine="true"  
    android:textCursorDrawable="@drawable/black_cursor"  
    android:id="@+id/ed_password"/>
```

```
</com.google.android.material.textfield.TextInputLayout>
```

```
<com.google.android.material.textfield.TextInputLayout  
    style="@style/Widget.MaterialComponents.TextInputLayout.OutlinedBox.Dense"  
    android:layout_width="match_parent"  
    android:layout_height="wrap_content"  
    android:layout_marginBottom="12dp"  
    android:layout_marginStart="20dp"  
    app:boxStrokeColor="@color/newpurple"  
    app:hintTextColor="@color/newpurple"  
    android:layout_marginEnd="20dp"  
    android:hint="Re-enter Password"
```

```
    app:endIconMode="password_toggle">
```

```
<com.google.android.material.textfield.TextInputEditText  
    android:layout_width="match_parent"  
    android:singleLine="true"  
    android:layout_height="wrap_content"
```

```
        android:textCursorDrawable="@drawable/black_cursor"
        " android:id="@+id/ed_confo_password"/>

</com.google.android.material.textfield.TextInputLayout>

<com.google.android.material.textfield.TextInputLayout
    style="@style/Widget.MaterialComponents.TextInputLayout.OutlinedBox.Dense"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_marginBottom="12dp"
    app:boxStrokeColor="@color/newpurple"
    app:hintTextColor="@color/newpurple"
    android:layout_marginStart="20dp"

    android:layout_marginEnd="20dp"
    android:hint="Enter Email ID">

    <com.google.android.material.textfield.TextInputEditText
        android:layout_width="match_parent"

        android:layout_height="wrap_content"
        android:singleLine="true"
        android:textCursorDrawable="@drawable/black_cursor"
        " android:id="@+id/ed_email" />

</com.google.android.material.textfield.TextInputLayout>

<Spinner
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:id="@+id/sp_c"
    android:layout_marginStart="20dp"
    android:layout_marginEnd="20dp"
    android:entries="@array/countries_array"
    android:padding="15dp"
    android:backgroundTint="@color/black"
    >

</Spinner>

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

```
        android:layout_gravity="center_horizontal"
        android:layout_marginBottom="20dp"
        android:backgroundTint="@color/newred"
        "android:textColor="@color/white"
        android:text="Create Account"
        android:onClick="insert_user"
    />

</LinearLayout>
</RelativeLayout>
```

Main_Activity.java

Description: this file take the input from text view and aadd this data into database.

```
package com.example.prac4_form;

import static java.security.AccessController.getContext;

import androidx.appcompat.app.AppCompatActivity;
import android.annotation.SuppressLint;
import android.content.Intent;
import android.os.Bundle;
import android.util.Log;

import android.view.View; import android.view.Window;
import android.view.WindowManager;
import android.widget.AdapterView;
import android.widget.AdapterView.OnItemClickListener;
import android.widget.ArrayAdapter;
import android.widget.Button;
import android.widget.EditText;
import android.widget.Spinner;
import android.widget.Toast;

import com.android.volley.Request;
import com.android.volley.RequestQueue;
import com.android.volley.Response;
import com.android.volley.VolleyError;
import com.android.volley.toolbox.JsonObjectRequest;
import com.android.volley.toolbox.Volley;
import org.json.JSONException; import org.json.JSONObject;
import java.util.ArrayList; import java.util.List;

public class MainActivity extends
    AppCompatActivity { Button btn;
```



```
EditText
uname,unnumber,upass,uconf_pass,umail;
Spinner sp1;

@Override

protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    requestWindowFeature(
        Window.FEATURE_NO_TITLE );
    getWindow().setFlags( WindowManager.LayoutParams.FLAG_FULLSCREEN,
        WindowManager.LayoutParams.FLAG_FULLSCREEN );
    setContentView(R.layout.main_activity);

    btn=findViewById(R.id.register_btn);
    uname=findViewById(R.id.ed_uname);
    unnumber=findViewById(R.id.ed_mobile_number);
    upass=findViewById(R.id.ed_password);
    uconf_pass=findViewById(R.id.ed_confo_password);
    umail=findViewById(R.id.ed_email);
    sp1 = findViewById(R.id.sp_c);

    btn.setOnClickListener(new
        View.OnClickListener() { @Override
        public void onClick(View view) {
            if(uname.getText().toString().equals("") || unnumber.getText().toString().equals("") ||
            upass.getText().toString().equals("")
                || umail.getText().toString().equals("") )
            {
                Toast.makeText(getApplicationContext(), "Enter Values in All
                Fields!", Toast.LENGTH_SHORT).show();
            }else {

                if(upass.getText().toString().equals(uconf_pass.getText().toString())){
                    if(sp1.getSelectedItem().toString().equals("Chose
                    your Country...")){
                        Toast.makeText(MainActivity.this, "Please select a
                        Country!", Toast.LENGTH_SHORT).show();
                    }else{
                        new_insert();
                    }
                }
            }
        }
    });
}
```

```

    }
    else{
        Toast.makeText(getApplicationContext(), "Passwords are different in
Both Fileds!",Toast.LENGTH_SHORT).show();
    }

}
}
});

}

private void new_insert() {
    String s = getResources().getString(
R.string.strlink);String str =
s+"registration_user.php?"+"
    "name="+uname.getText().toString().trim()+
    "&mobile_number="+unumber.getText().toString().tri
m()+
    "&password="+upass.getText().toString().trim()+
    "&mail="+umail.getText().toString().trim()+
    "&country="+sp1.getSelectedItem().toString();
    Log.e("str : ", str);
    JSONObjectRequest jsonObjectRequest = new
JsonObjectRequest(Request.Method.GET, str, null,new
Response.Listener<JSONObject>() {
        @Override
        public void onResponse(JSONObject
response) { try{

            if(response.getString("status").equals("0")){
                Toast.makeText(getApplicationContext(), "Fail to
Insert",
Toast.LENGTH_SHORT).show();
            }

            else if(response.getString("status").equals("1")){
                Toast.makeText(getApplicationContext(), "Account Created
Successfully!",
Toast.LENGTH_SHORT).show();
            }
        }
    }
}

```

```

    }
    }catch (JSONException
        e){ e.printStackTrace();

    }
}
}, new
Response.ErrorListener() {
    @Override
    public void onErrorResponse(VolleyError
        error) { error.printStackTrace();
    }
});

RequestQueue a = Volley.newRequestQueue(this);
a.add(jsonObjectRequest);

}
}

```

AndroidManifest.xml

Description: Manifest file is used for configuration and context information about the android application like icon of application, name, theme and it will also contains intents of the application.

```

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

    <application
        android:allowBackup="true"
        android:icon="@mipmap/ic_launcher"
        android:label="@string/app_name"
        android:roundIcon="@mipmap/ic_launcher_round"
        android:supportsRtl="true"
        android:theme="@style/Theme.LoginPage">
        <activity
            android:name=".MainActivity"
            android:exported="true">
            <intent-filter>

```

```

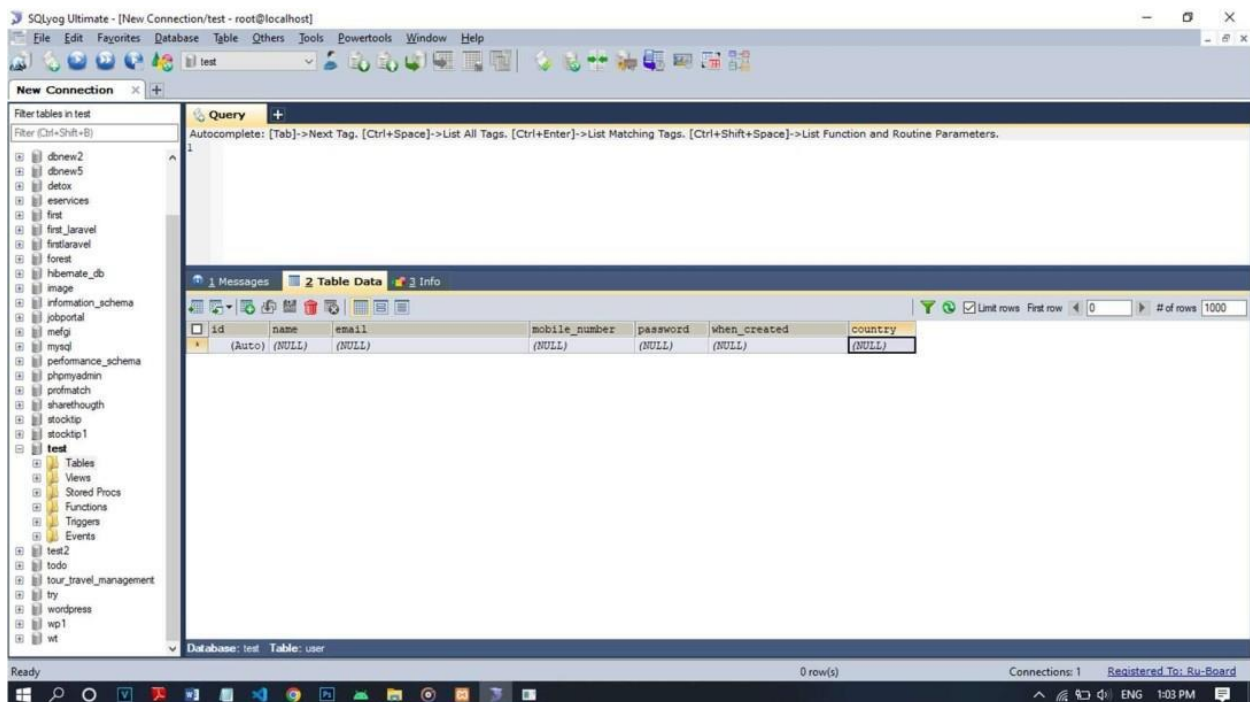
    <action android:name="android.intent.action.MAIN" />
    <category android:name="android.intent.category.LAUNCHER" />
  </intent-filter>
</activity>
</application>

</manifest>

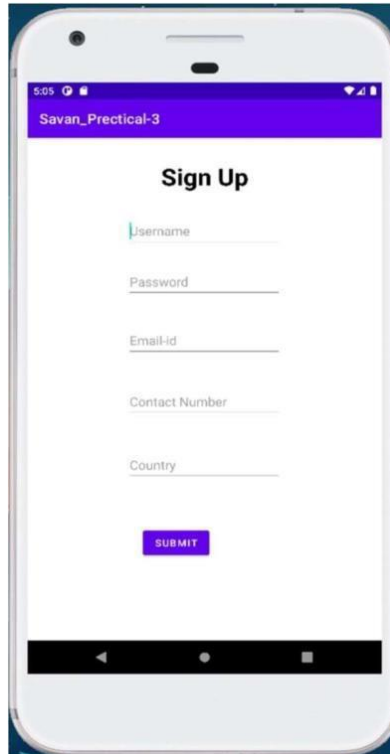
```

Output Screenshot:

Description: this is the screen shot of database table



Description: this is the screenshot of main layout of application when you enter data and click on the submit button your data go to database.



Final Project

Main.dart

Description: this is the main file when the project run this file run first.

```
import 'package:flutter/material.dart';
import 'package:firebase_core/firebase_core.dart';

import 'auth/main_page.dart';

void main() async {
  WidgetsFlutterBinding.ensureInitialized();
  await Firebase.initializeApp();
  runApp(MyApp());
}

class MyApp extends StatelessWidget {
  const MyApp({Key? key}) : super(key: key);

  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      debugShowCheckedModeBanner: false,
      home: MainPage(),
    );
  }
}
```

auth_page.dart

description: this is the authentication page when user enter username and password it authenticate the user with username and password if it correct then it go further.

```
import 'package:flutter/material.dart';
import '../pages/register_page.dart';

import '../pages/login_page.dart';

class AuthPage extends StatefulWidget {
```



```
const AuthPage({Key? key}) : super(key: key);

@override
State<AuthPage> createState() => _AuthPageState();
}

class _AuthPageState extends State<AuthPage> {
  bool showLoginPage = true;

  void toggleScreens(){
    setState() {
      showLoginPage = !showLoginPage;
    });
  }
  @override
  Widget build(BuildContext context) {
    if(showLoginPage)
    {
      return LoginPage(showRegisterPage: toggleScreens);
    }
    else {
      return RegisterPage(showLoginPage: toggleScreens,);
    }
  }
}
```

main_page.dart

description: this is the main page when user enter password and id it displays the information from the database.

```
import 'package:firebase_auth/firebase_auth.dart';
import 'package:flutter/material.dart';

import 'auth_page.dart';
import '../pages/home_page.dart';

class MainPage extends StatelessWidget {
  const MainPage({Key? key}) : super(key: key);

  @override
  Widget build(BuildContext context) {
    return Scaffold(
```



```
body: StreamBuilder<User?>(
  stream: FirebaseAuth.instance.authStateChanges(),
  builder: (context, snapshot)
  {
    if(snapshot.hasData) {
      return HomePage();
    }
    else {
      return AuthPage();
    }
  },
),
);
}
```

register_page.dart

description: this is the register page when new user comes it have to register here.

```
import 'package:cloud_firestore/cloud_firestore.dart';
import 'package:flutter/material.dart';
import 'package:google_fonts/google_fonts.dart';
import 'package:firebase_auth/firebase_auth.dart';
import 'package:animated_text_kit/animated_text_kit.dart';

class RegisterPage extends StatefulWidget {
  final VoidCallback showLoginPage;

  const RegisterPage({Key? key, required this.showLoginPage}) : super(key: key);

  @override
  State<RegisterPage> createState() => _RegisterPageState();
}

class _RegisterPageState extends State<RegisterPage> {
  final _emailController = TextEditingController();
  final _passwordController = TextEditingController();
  final _confirmPasswordController = TextEditingController();
  final _firstNameController = TextEditingController();
  final _lastNameController = TextEditingController();
  final _ageController = TextEditingController();
```




```
Future signUp() async {
  if (passwordConfirmed()) {
    await FirebaseAuth.instance.createUserWithEmailAndPassword(
      email: _emailController.text.trim(),
      password: _passwordController.text.trim(),
    );
    addUserDetails(
      _firstNameController.text.trim(),
      _lastNameController.text.trim(),
      _emailController.text.trim(),
      int.parse(
        _ageController.text.trim(),
      ),
    );
  }
}
```

```
Future addUserDetails(
  String firstName, String lastName, String email, int age) async {
  await FirebaseFirestore.instance.collection('Users').add({
    'first name': firstName,
    'last name': lastName,
    'email': email,
    'age': age,
  });
}
```

```
bool passwordConfirmed() {
  if (_passwordController.text.trim() ==
    _confirmPasswordController.text.trim()) {
    return true;
  } else {
    return false;
  }
}
```

```
@override
void dispose() {
  _emailController.dispose();
  _passwordController.dispose();
  _confirmPasswordController.dispose();
  _firstNameController.dispose();
  _lastNameController.dispose();
  _ageController.dispose();
}
```



```
super.dispose();
}

@override
Widget build(BuildContext context) {
  return Scaffold(
    backgroundColor: Colors.grey[300],
    body: SafeArea(
      child: Center(
        child: SingleChildScrollView(
          child: SingleChildScrollView(
            child: Column(
              mainAxisAlignment: MainAxisAlignment.center,
              children: [
                const SizedBox(
                  height: 75,
                ),
                AnimatedTextKit(
                  animatedTexts: [
                    TypewriterAnimatedText(
                      'Welcome To App',
                      speed: const Duration(milliseconds: 150),
                      textStyle: const TextStyle(
                        fontSize: 36,
                        fontWeight: FontWeight.bold,
                      ),
                    ),
                  ],
                ),
                isRepeatingAnimation: true,
                repeatForever: true,
                displayFullTextOnTap: true,
                stopPauseOnTap: false,
              ),
                const SizedBox(
                  height: 10,
                ),
                Text(
                  "Register Below With Your Details!",
                  style: GoogleFonts.lato(
                    textStyle: const TextStyle(fontSize: 20),
                  ),
                ),
                const SizedBox(
                  height: 50,
                ),
              ],
            ),
          ),
        ),
      ),
    ),
  );
}
```

```
),
Padding(
  padding: const EdgeInsets.symmetric(horizontal: 25.0),
  child: Container(
    decoration: BoxDecoration(
      color: Colors.grey[200],
      border: Border.all(color: Colors.deepPurple),
      borderRadius: BorderRadius.circular(12),
    ),
    child: Padding(
      padding: const EdgeInsets.only(left: 20.0),
      child: TextField(
        style: GoogleFonts.lato(),
        controller: _firstNameController,
        decoration: const InputDecoration(
          border: InputBorder.none,
          hintText: "First Name",
        ),
      ),
    ),
  ),
),
const SizedBox(
  height: 20,
),
Padding(
  padding: const EdgeInsets.symmetric(horizontal: 25.0),
  child: Container(
    decoration: BoxDecoration(
      color: Colors.grey[200],
      border: Border.all(color: Colors.deepPurple),
      borderRadius: BorderRadius.circular(12),
    ),
    child: Padding(
      padding: const EdgeInsets.only(left: 20.0),
      child: TextField(
        style: GoogleFonts.lato(),
        controller: _lastNameController,
        decoration: const InputDecoration(
          border: InputBorder.none,
          hintText: "Last Name",
        ),
      ),
    ),
  ),
),
```

```
    ),
  ),
  const SizedBox(
    height: 20,
  ),
  Padding(
    padding: const EdgeInsets.symmetric(horizontal: 25.0),
    child: Container(
      decoration: BoxDecoration(
        color: Colors.grey[200],
        border: Border.all(color: Colors.deepPurple),
        borderRadius: BorderRadius.circular(12),
      ),
    ),
    child: Padding(
      padding: const EdgeInsets.only(left: 20.0),
      child: TextField(
        style: GoogleFonts.lato(),
        controller: _ageController,
        decoration: const InputDecoration(
          border: InputBorder.none,
          hintText: "Age",
        ),
      ),
    ),
  ),
),
const SizedBox(
  height: 20,
),
Padding(
  padding: const EdgeInsets.symmetric(horizontal: 25.0),
  child: Container(
    decoration: BoxDecoration(
      color: Colors.grey[200],
      border: Border.all(color: Colors.deepPurple),
      borderRadius: BorderRadius.circular(12),
    ),
  ),
  child: Padding(
    padding: const EdgeInsets.only(left: 20.0),
    child: TextField(
      style: GoogleFonts.lato(),
      controller: _emailController,
      decoration: const InputDecoration(
        border: InputBorder.none,
```



```
        hintText: "Email Address",
      ),
    ),
  ),
),
const SizedBox(
  height: 20,
),
Padding(
  padding: const EdgeInsets.symmetric(horizontal: 25.0),
  child: Container(
    decoration: BoxDecoration(
      color: Colors.grey[200],
      border: Border.all(color: Colors.deepPurple),
      borderRadius: BorderRadius.circular(12),
    ),
    child: Padding(
      padding: const EdgeInsets.only(left: 20.0),
      child: TextField(
        style: GoogleFonts.lato(),
        controller: _passwordController,
        obscureText: true,
        decoration: const InputDecoration(
          border: InputBorder.none,
          hintText: "Password",
        ),
      ),
    ),
  ),
),
const SizedBox(
  height: 20,
),
Padding(
  padding: const EdgeInsets.symmetric(horizontal: 25.0),
  child: Container(
    decoration: BoxDecoration(
      color: Colors.grey[200],
      border: Border.all(color: Colors.deepPurple),
      borderRadius: BorderRadius.circular(12),
    ),
    child: Padding(
      padding: const EdgeInsets.only(left: 20.0),
```



```
child: TextField(  
  style: GoogleFonts.lato(),  
  controller: _confirmPasswordController,  
  obscureText: true,  
  decoration: const InputDecoration(  
    border: InputBorder.none,  
    hintText: "Confirm Password",  
  ),  
),  
),  
),  
),  
),  
const SizedBox(  
  height: 20,  
),  
Padding(  
  padding: const EdgeInsets.symmetric(horizontal: 25.0),  
  child: GestureDetector(  
    onTap: signUp,  
    child: Container(  
      padding: const EdgeInsets.all(20),  
      decoration: BoxDecoration(  
        color: Colors.deepPurple,  
        borderRadius: BorderRadius.circular(12),  
      ),  
      child: Center(  
        child: Text('Sign Up',  
          style: GoogleFonts.lato(  
            textStyle: const TextStyle(  
              color: Colors.white,  
              fontWeight: FontWeight.bold,  
              fontSize: 18,  
            ),  
          )),  
      ),  
    ),  
  ),  
),  
const SizedBox(  
  height: 25,  
),  
Row(  
  mainAxisAlignment: MainAxisAlignment.center,  
  children: [
```

login_page.dart

```
import 'package:animated_text_kit/animated_text_kit.dart';
import 'forgot_pw_pages.dart';
import 'package:firebase_auth/firebase_auth.dart';
import 'package:flutter/material.dart';
import 'package:google_fonts/google_fonts.dart';
```

```
import 'dart:math';

class LoginPage extends StatefulWidget {
  final VoidCallback showRegisterPage;

  const LoginPage({Key? key, required this.showRegisterPage}) : super(key: key);

  @override
  State<LoginPage> createState() => _LoginPageState();
}

class _LoginPageState extends State<LoginPage> with TickerProviderStateMixin {
  String status = " ";
  final _emailController = TextEditingController();
  final _passwordController = TextEditingController();
  late AnimationController firstController;
  late Animation<double> firstAnimation;

  late AnimationController secondController;
  late Animation<double> secondAnimation;

  late AnimationController thirdController;
  late Animation<double> thirdAnimation;

  late AnimationController fourthController;
  late Animation<double> fourthAnimation;

  late AnimationController fifthController;
  late Animation<double> fifthAnimation;

  Future signIn() async {
    showDialog(
      context: context,
      builder: (context) {
        return Center(
          child: Container(
            height: 100,
            width: 100,
            child: CustomPaint(
              painter: MyPainter(
                firstAnimation.value,
                secondAnimation.value,
                thirdAnimation.value,
                fourthAnimation.value,
```




```
        fifthAnimation.value,
    ),
),
),
);
},
);

await FirebaseAuth.instance.signInWithEmailAndPassword(
    email: _emailController.text.trim(),
    password: _passwordController.text.trim(),
);

Navigator.of(context).pop();
}

@override
void dispose() {
    _emailController.dispose();
    _passwordController.dispose();
    firstController.dispose();
    secondController.dispose();
    thirdController.dispose();
    fourthController.dispose();
    fifthController.dispose();
    super.dispose();
}

@override
void initState() {
    super.initState();

    firstController =
        AnimationController(vsync: this, duration: Duration(seconds: 6));
    firstAnimation = Tween<double>(begin: -pi, end: pi).animate(firstController)
        ..addListener(() {
            setState(() {});
        })
        ..addStatusListener((status) {
            if (status == AnimationStatus.completed) {
                firstController.repeat();
            } else if (status == AnimationStatus.dismissed) {
                firstController.forward();
            }
        })
    ;
```

```
});

secondController =
    AnimationController(vsync: this, duration: Duration(seconds: 3));
secondAnimation =
    Tween<double>(begin: -pi, end: pi).animate(secondController)
    ..addListener(() {
        setState(() {});
    })
    ..addStatusListener((status) {
        if (status == AnimationStatus.completed) {
            secondController.repeat();
        } else if (status == AnimationStatus.dismissed) {
            secondController.forward();
        }
    });

thirdController =
    AnimationController(vsync: this, duration: Duration(seconds: 2));
thirdAnimation = Tween<double>(begin: -pi, end: pi).animate(thirdController)
    ..addListener(() {
        setState(() {});
    })
    ..addStatusListener((status) {
        if (status == AnimationStatus.completed) {
            thirdController.repeat();
        } else if (status == AnimationStatus.dismissed) {
            thirdController.forward();
        }
    });

fourthController = AnimationController(
    vsync: this, duration: Duration(milliseconds: 1500));
fourthAnimation =
    Tween<double>(begin: -pi, end: pi).animate(fourthController)
    ..addListener(() {
        setState(() {});
    })
    ..addStatusListener((status) {
        if (status == AnimationStatus.completed) {
            fourthController.repeat();
        } else if (status == AnimationStatus.dismissed) {
            fourthController.forward();
        }
    });
```

```

    });

    fifthController =
      AnimationController(vsync: this, duration: Duration(seconds: 1));
    fifthAnimation = Tween<double>(begin: -pi, end: pi).animate(fifthController)
      ..addListener(() {
        setState(() {});
      })
      ..addStatusListener((status) {
        if (status == AnimationStatus.completed) {
          fifthController.repeat();
        } else if (status == AnimationStatus.dismissed) {
          fifthController.forward();
        }
      });

    firstController.forward();
    secondController.forward();
    thirdController.forward();
    fourthController.forward();
    fifthController.forward();
  }

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      backgroundColor: Colors.grey[300],
      body: SafeArea(
        child: Center(
          child: SingleChildScrollView(
            child: Column(
              mainAxisAlignment: MainAxisAlignment.center,
              children: [
                const Icon(
                  Icons.phone_android_rounded,
                  size: 100,
                ),
                const SizedBox(
                  height: 75,
                ),
                AnimatedTextKit(
                  animatedTexts: [
                    TypewriterAnimatedText(
                      'Hello Again!',

```



```
        speed: const Duration(milliseconds: 150),
        textStyle: const TextStyle(
          fontSize: 36,
          fontWeight: FontWeight.bold,
        ),
      ),
    ],
    isRepeatingAnimation: true,
    repeatForever: true,
    displayFullTextOnTap: true,
    stopPauseOnTap: false,
  ),
  const SizedBox(
    height: 10,
  ),
  Text(
    "Welcome Back, You've been missed!",
    style: GoogleFonts.lato(
      textStyle: const TextStyle(
        fontSize: 20,
      ),
    ),
  ),
  SizedBox(
    height: 50,
    child: Text(
      status
    )
  ),
  Padding(
    padding: const EdgeInsets.symmetric(horizontal: 25.0),
    child: Container(
      decoration: BoxDecoration(
        color: Colors.grey[200],
        border: Border.all(color: Colors.deepPurple),
        borderRadius: BorderRadius.circular(12),
      ),
      child: Padding(
        padding: const EdgeInsets.only(left: 20.0),
        child: TextField(
          style: GoogleFonts.lato(),
          controller: _emailController,
          decoration: const InputDecoration(
            border: InputBorder.none,
```



```
        hintText: "Email Address",
      ),
    ),
  ),
),
const SizedBox(
  height: 20,
),
Padding(
  padding: const EdgeInsets.symmetric(horizontal: 25.0),
  child: Container(
    decoration: BoxDecoration(
      color: Colors.grey[200],
      border: Border.all(color: Colors.deepPurple),
      borderRadius: BorderRadius.circular(12),
    ),
    child: Padding(
      padding: const EdgeInsets.only(left: 20.0),
      child: TextField(
        style: GoogleFonts.lato(),
        controller: _passwordController,
        obscureText: true,
        decoration: const InputDecoration(
          border: InputBorder.none,
          hintText: "Password",
        ),
      ),
    ),
  ),
),
const SizedBox(
  height: 20,
),
Padding(
  padding: const EdgeInsets.symmetric(horizontal: 25.0),
  child: Row(
    mainAxisAlignment: MainAxisAlignment.end,
    children: [
      GestureDetector(
        onTap: () {
          Navigator.push(
            context,
            MaterialPageRoute(builder: (context) {
```




```

    ),
    ),
    const SizedBox(
      height: 25,
    ),
    Row(
      mainAxisAlignment: MainAxisAlignment.center,
      children: [
        Text(
          'Not a member?',
          style: GoogleFonts.lato(
            textStyle: const TextStyle(
              fontWeight: FontWeight.bold,
              fontSize: 20,
            ),
          ),
        ),
        GestureDetector(
          onTap: widget.showRegisterPage,
          child: Text(
            'Register now',
            style: GoogleFonts.lato(
              textStyle: const TextStyle(
                fontSize: 20,
                color: Colors.blue,
                fontWeight: FontWeight.bold,
              ),
            ),
          ),
        ),
      ],
    ),
  ],
),
),
),
),
);
}
}

```

```

class MyPainter extends CustomPainter {
  final double firstAngle;
  final double secondAngle;

```



```
final double thirdAngle;  
final double fourthAngle;  
final double fifthAngle;
```

```
MyPainter(  
    this.firstAngle,  
    this.secondAngle,  
    this.thirdAngle,  
    this.fourthAngle,  
    this.fifthAngle,  
);
```

```
@override  
void paint(Canvas canvas, Size size) {  
    Paint myArc = Paint()  
    ..color = Color(0xff00A2FF)  
    ..style = PaintingStyle.stroke  
    ..strokeWidth = 5  
    ..strokeCap = StrokeCap.round;
```

```
    canvas.drawArc(  
        Rect.fromLTRB(  
            0,  
            0,  
            size.width,  
            size.height,  
        ),  
        firstAngle,  
        2,  
        false,  
        myArc,  
    );
```

```
    canvas.drawArc(  
        Rect.fromLTRB(  
            size.width * .1,  
            size.height * .1,  
            size.width * .9,  
            size.height * .9,  
        ),  
        secondAngle,  
        2,  
        false,  
        myArc,  
    );
```




```
canvas.drawArc(  
    Rect.fromLTRB(  
        size.width * .2,  
        size.height * .2,  
        size.width * .8,  
        size.height * .8,  
    ),  
    thirdAngle,  
    2,  
    false,  
    myArc,  
);  
canvas.drawArc(  
    Rect.fromLTRB(  
        size.width * .3,  
        size.height * .3,  
        size.width * .7,  
        size.height * .7,  
    ),  
    fourthAngle,  
    2,  
    false,  
    myArc,  
);  
canvas.drawArc(  
    Rect.fromLTRB(  
        size.width * .4,  
        size.height * .4,  
        size.width * .6,  
        size.height * .6,  
    ),  
    fifthAngle,  
    2,  
    false,  
    myArc,  
);  
}  
  
@override  
bool shouldRepaint(covariant CustomPainter oldDelegate) {  
    return true;  
}  
}
```

home_page.dart

description: this is the main page when it displays the all the information from the database.

```
import 'dart:ui';

import 'package:cloud_firestore/cloud_firestore.dart';
import 'package:flutter/material.dart';
import 'package:firebase_auth/firebase_auth.dart';
import 'package:flutter_staggered_animations/flutter_staggered_animations.dart';
import '../read data/get_user_name.dart';

class HomePage extends StatefulWidget {
  const HomePage({Key? key}) : super(key: key);

  @override
  State<HomePage> createState() => _HomePageState();
}

class _HomePageState extends State<HomePage> {
  final user = FirebaseAuth.instance.currentUser!;

  List<String> docIDs = [];

  Future getDocId() async {
    await FirebaseFirestore.instance.collection('Users').orderBy('first name', descending:
false).get().then(
      (snapshot) => snapshot.docs.forEach(
        (document) {
          docIDs.add(document.reference.id);
        },
      ),
    );
  }

  @override
  Widget build(BuildContext context) {
    double _w = MediaQuery.of(context).size.width;
    return Scaffold(
      appBar: AppBar(
        title: Text(
          user.email!,
          style: TextStyle(fontSize: 16, fontWeight: FontWeight.bold),

```



```

    ),
    centerTitle: true,
    backgroundColor: Colors.deepPurple,
    elevation: 6,
    actions: [
      GestureDetector(
        onTap: () {
          FirebaseAuth.instance.signOut();
        },
        child: Icon(Icons.logout),
      )
    ],
  ),
  body: Center(
    child: Column(
      mainAxisAlignment: MainAxisAlignment.center,
      children: [
        Expanded(
          child: FutureBuilder(
            future: getDocId(),
            builder: (context, snapshot) {
              // return ListView.builder(
              //   itemCount: docIDs.length,
              //   itemBuilder: (context, index) {
              //     return Padding(
              //       padding: const EdgeInsets.all(8.0),
              //       child: ListTile(
              //         title: GetUserName(documentId: docIDs[index]),
              //         tileColor: Colors.grey,
              //       ),
              //     );
              //   },
              // );
              return ListView.builder(
                padding: EdgeInsets.all(_w / 30),
                physics:
                  BouncingScrollPhysics(parent: AlwaysScrollableScrollPhysics()),
                itemCount: docIDs.length,
                itemBuilder: (BuildContext context, int index) {
                  return AnimationConfiguration.staggeredList(
                    position: index,
                    delay: Duration(milliseconds: 100),
                    child: SlideAnimation(
                      duration: Duration(milliseconds: 2500),

```

```

curve: Curves.fastLinearToSlowEaseIn,
verticalOffset: -250,
child: ScaleAnimation(
  duration: Duration(milliseconds: 1500),
  curve: Curves.fastLinearToSlowEaseIn,
  child: Container(
    child: ListTile(
      title: GetUserName(documentId: docIDs[index]),
      tileColor: Colors.white,
    ),
    margin: EdgeInsets.only(bottom: _w / 20),
    height: _w / 4,
    decoration: BoxDecoration(
      color: Colors.white,
      borderRadius: BorderRadius.all(Radius.circular(20)),
      boxShadow: [
        BoxShadow(
          color: Colors.black.withOpacity(0.1),
          blurRadius: 40,
          spreadRadius: 10,
        ),
      ],
    ),
  ),
),
);
},
);
},
);
};
},
);
},
),
),
),
),
),
);
}
}

```

forgot_pw_pages.dart

description: when user forgot it password it enter its email then it receive the reset password link and it change the password.



```
import 'package:flutter/material.dart';
import 'package:firebase_auth/firebase_auth.dart';

class ForgotPasswordPage extends StatefulWidget {
  const ForgotPasswordPage({Key? key}) : super(key: key);

  @override
  State<ForgotPasswordPage> createState() => _ForgotPasswordPageState();
}

class _ForgotPasswordPageState extends State<ForgotPasswordPage> {
  final _emailController = TextEditingController();

  @override
  void dispose() {
    _emailController.dispose();
    super.dispose();
  }

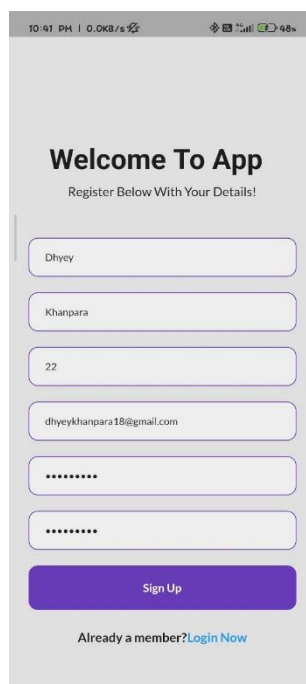
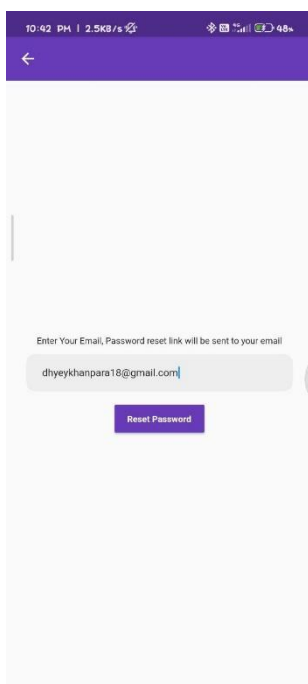
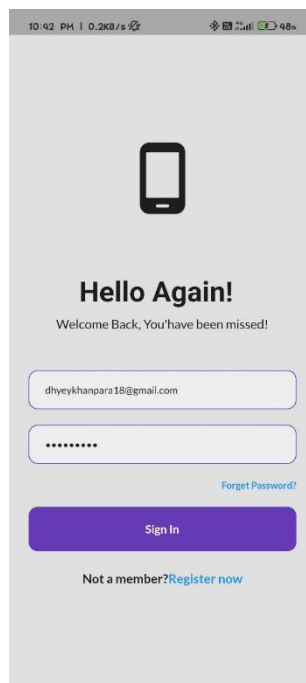
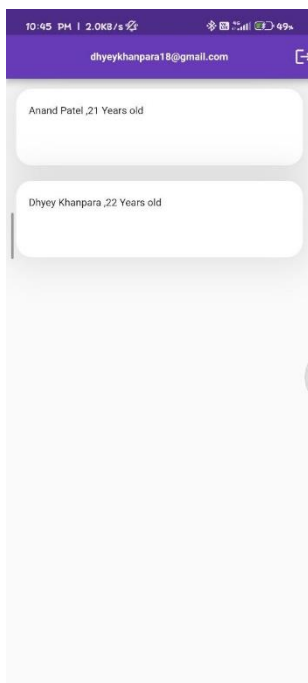
  Future passwordReset() async {
    try {
      await FirebaseAuth.instance
        .sendPasswordResetEmail(email: _emailController.text.trim());
      showDialog(
        context: context,
        builder: (context) {
          return AlertDialog(
            content: Text("Password link Successfully sent to " + _emailController.text.trim()),
          );
        },
      );
    } on FirebaseAuthException catch (e) {
      print(e);
      showDialog(
        context: context,
        builder: (context) {
          return AlertDialog(
            content: Text(e.message.toString()),
          );
        },
      );
    }
  }
}
```

[illegible]

```
        style:
            TextStyle(color: Colors.white, fontWeight: FontWeight.bold),
        ),
        color: Colors.deepPurple,
    )
    ],
),
);
}
}
```

Output:

Description: when we run app, it display the login page, If there is new user it have to register otherwise it login and it displays the information from the database





Certificate

