



# **Channabasaveshwara Institute of Technology**

(Affiliated to VTU, Belgaum & Approved by AICTE, New Delhi)

**(ISO 9001:2015 Certified Institution)**

NH 206 (B.H. Road), Gubbi, Tumkur – 572 216. Karnataka.



# **Mobile Application Development Laboratory [18CSMP68]**

**Department of Computer Science & Engineering**

**VI Semester**

**Academic Year : 2022-23**



# Channabasaveshwara Institute of Technology

(Affiliated to VTU, Belgaum & Approved by AICTE, New Delhi)  
 (NAAC Accredited & ISO 9001:2015 Certified Institution)  
 NH 206 (B.H. Road), Gubbi, Tumkur – 572216. Karnataka.



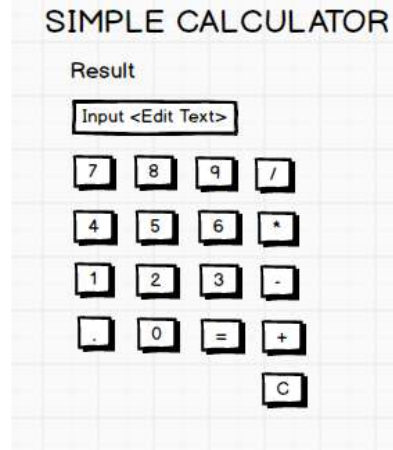
## DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

# Syllabus

MOBILE APPLICATION DEVELOPMENT (Effective from the academic year 2020 -2021) SEMESTER – VI			
Course Code	18CSMP68	IA Marks	40
Number of Contact Hours/Week	0:0:2	Exam Marks	60
Total Number of Contact Hours	3 Hours/Week	Exam Hours	03
CREDITS – 02			
<b>Laboratory Objectives:</b> This laboratory (18CSMP68) will enable students to			
<ul style="list-style-type: none"> <li>• Learn and acquire the art of Android Programming.</li> <li>• Configure Android studio to run the applications.</li> <li>• Understand and implement Android's User interface functions.</li> <li>• Create, modify and query on SQLite database.</li> <li>• Inspect different methods of sharing data using services.</li> </ul>			
<b>Descriptions (if any):</b>			
<ol style="list-style-type: none"> <li>1. The installation procedure of the Android Studio/Java software must be demonstrated and carried out in groups.</li> <li>2. Students should use the latest version of Android Studio/Java/ Kotlin to execute these programs. Diagrams given are for representational purposes only; students are expected to improvise on them.</li> <li>3. <b>Part B programs should be developed as an application and are to be demonstrated as a mini project in a group by adding extra features or the students can also develop their application and demonstrate it as a mini-project. (Projects/programs are not limited to the list given in Part B).</b></li> </ol>			
<b>Programs List:</b>			
PART – A			
1	Create an application to design a Visiting Card. The Visiting card should have a company logo at the top right corner. The company name should be displayed in Capital letters, aligned to the center. Information like the name of the employee, job title, phone number, address, email, fax and the website address is to be displayed. Insert a horizontal line between the job title and the phone number.		

2

Develop an Android application using controls like Button, TextView, EditText for designing a calculator having basic functionality like Addition, Subtraction, Multiplication, and Division.

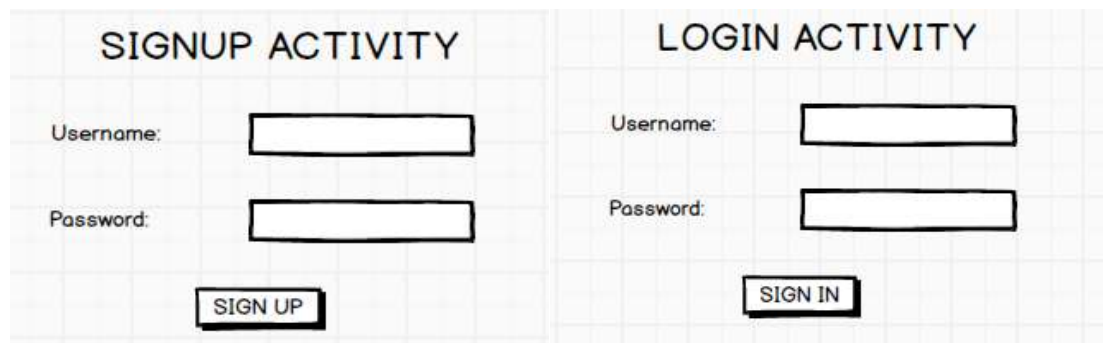


3

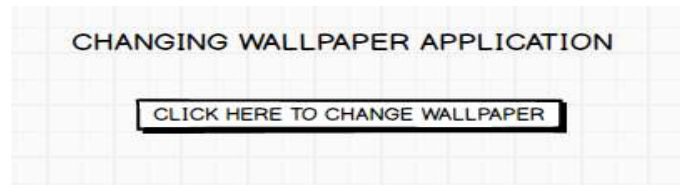
Create a SIGN Up activity with Username and Password. Validation of password should happen based on the following rules:

- Password should contain uppercase and lowercase letters.
- Password should contain letters and numbers.
- Password should contain special characters.
- Minimum length of the password (the default value is 8).

On successful **SIGN UP** proceed to the next Login activity. Here the user should **SIGN IN** using the Username and Password created during signup activity. If the Username and Password are matched then navigate to the next activity which displays a message saying "Successful Login" or else display a toast message saying "Login Failed". The user is given only two attempts and after that display a toast message saying "Failed Login Attempts" and disable the SIGN IN button. Use Bundle to transfer information from one activity to another.



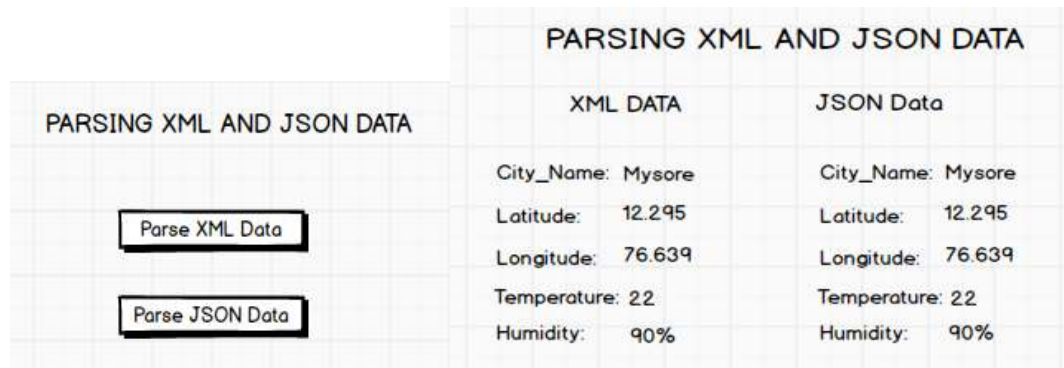
- 4 Develop an application to set an image as wallpaper. On click of a button, the wallpaper image should start to change randomly every 30 seconds.




- 5 Write a program to create an activity with two buttons START and STOP. On pressing of the START button, the activity must start the counter by displaying the numbers from One and the counter must keep on counting until the STOP button is pressed. Display the counter value in a TextView control.



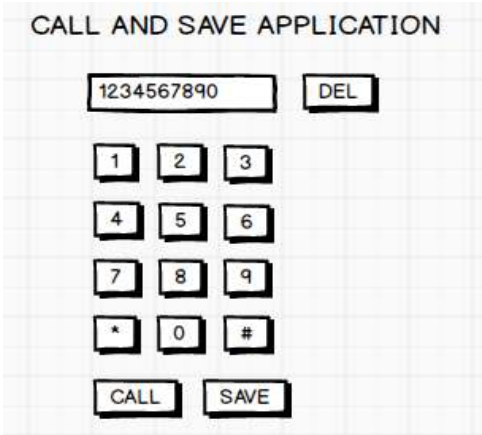
- 6 Create two files of XML and JSON type with values for City\_Name, Latitude, Longitude, Temperature, and Humidity. Develop an application to create an activity with two buttons to parse the XML and JSON files which when clicked should display the data in their respective layouts side by side.



7 Develop a simple application with one EditText so that the user can write some text in it. Create a button called “Convert Text to Speech” that converts the user input text into voice.

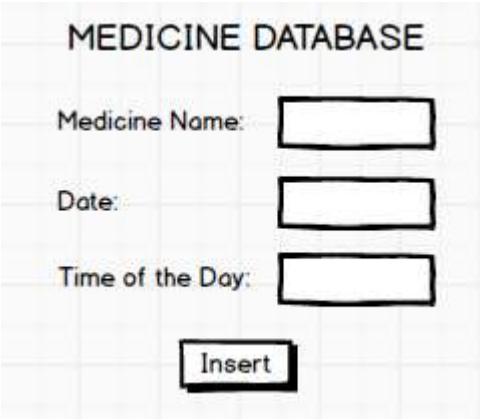


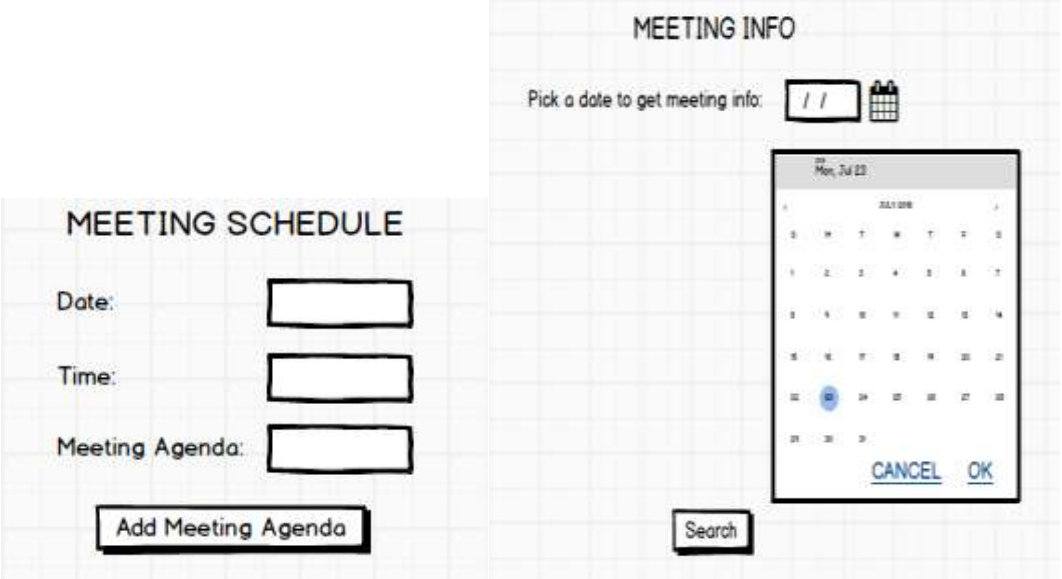

8 Create an activity like a phone dialer with CALL and SAVE buttons. On pressing the CALL button, it must call the phone number and on pressing the SAVE button it must save the number to the phonecontacts.

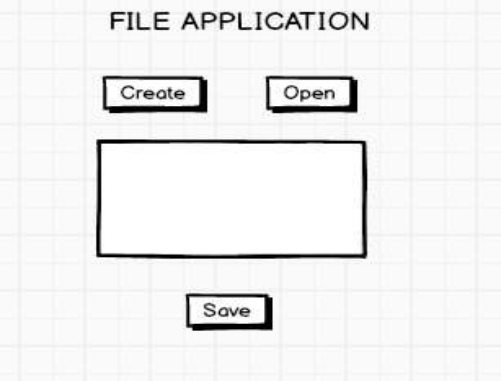
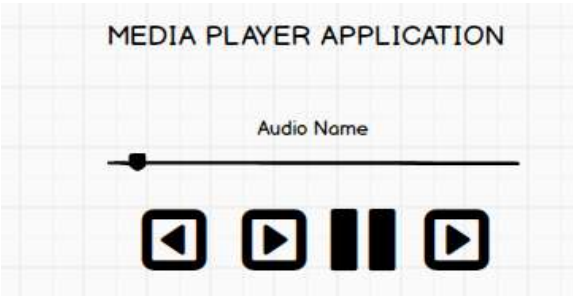
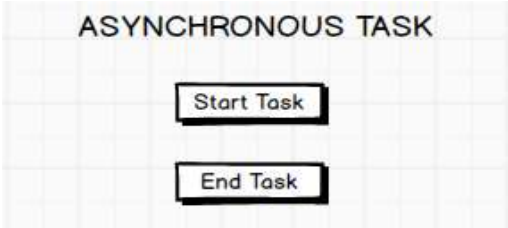


**PART - B**

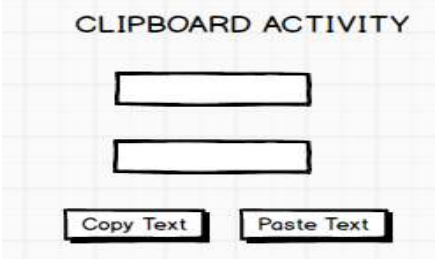
1 Write a program to enter Medicine Name, Date and Time of the Day as input from the user and store it in the SQLite database. Input for Time of the Day should be either Morning or Afternoon or Evening or Night. Trigger an alarm based on the Date and Time of the Day and display the Medicine Name.



<p>2</p>	<p>Develop a content provider application with an activity called “Meeting Schedule” which takes Date, Time and Meeting Agenda as input from the user and store this information into the SQLite database. Create another application with an activity called “Meeting Info” having DatePicker control, which on the selection of a date should display the Meeting Agenda information for that particular date, else it should display a toast message saying “No Meeting on this Date”.</p>  <p>The image shows two screenshots of an Android application. The left screenshot, titled "MEETING SCHEDULE", features three text input fields labeled "Date:", "Time:", and "Meeting Agenda:", each with a corresponding empty text box. Below these fields is a button labeled "Add Meeting Agenda". The right screenshot, titled "MEETING INFO", shows a date picker interface. It includes a text prompt "Pick a date to get meeting info:" followed by a date input field containing " / /" and a calendar icon. A calendar is displayed below, showing the month of July 2018 with the 22nd selected. At the bottom of the calendar are "CANCEL" and "OK" buttons. Below the calendar is a "Search" button.</p>
<p>3</p>	<p>Create an application to receive an incoming SMS which is notified to the user. On clicking this SMS notification, the message content and the number should be displayed on the screen. Use appropriate emulator control to send the SMS message to your application.</p>  <p>The image shows a screenshot of an application titled "SMS APPLICATION". It contains two text labels: "Display SMS Number" and "Display SMS Message", arranged vertically.</p>
<p>4</p>	<p>Write a program to create an activity having a Text box, and also Save Open and Create buttons. The user has to write some text in the Text box. On pressing the Create button the text should be saved as a text file in Mksdcard. On subsequent changes to the text, the Save button should be pressed to store the latest content to the same file. On pressing the Open button, it should display the contents from the previously stored files in the Text box. If the user tries to save the contents in the Textbox to a file without creating it, then a toast message has to be displayed saying “First Create aFile”.</p>

	
5	<p>Create an application to demonstrate a basic media player that allows the user to Forward, Backward, Play and Pause an audio. Also, make use of the indicator in the seek bar to move the audio forward or backward as required.</p> 
6	<p>Develop an application to demonstrate the use of Asynchronous tasks in android. The asynchronous task should implement the functionality of a simple moving banner. On pressing the <b>Start Task</b> button, the banner message should scroll from right to left. On pressing the <b>Stop Task</b> button, the banner message should stop. Let the banner message be “Demonstration of Asynchronous Task”.</p> 

7 Develop an application that makes use of the clipboard framework for copying and pasting of the text. The activity consists of two EditText controls and two Buttons to trigger the copy and paste functionality.



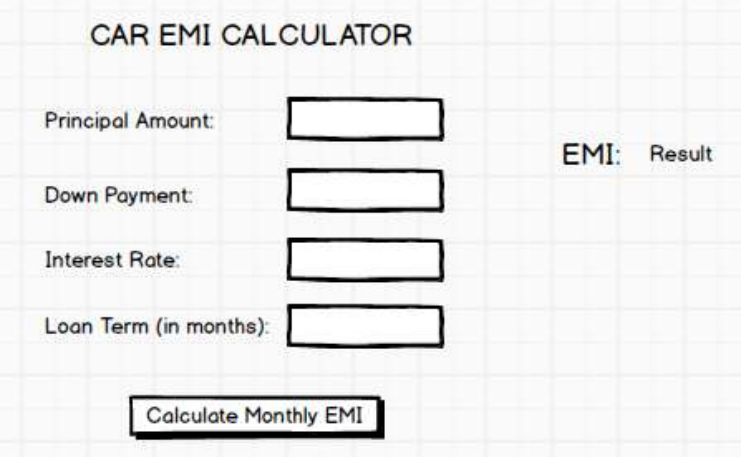
8 Create an AIDL service that calculates Car Loan EMI. The formula to calculate EMI is

$$E = P * (r(1+r)^n)/((1+r)^n - 1)$$

where

- E = The EMI payable on the car loan amount
- P = The Car loan Principal Amount
- r = The interest rate value computed on a monthly basis
- n = The loan tenure in the form of months

The down payment amount has to be deducted from the principal amount paid towards buying the Car. Develop an application that makes use of this AIDL service to calculate the EMI. This application should have four EditText to read the Principal Amount, Down Payment, Interest Rate, Loan Term (in months) and a button named as “Calculate Monthly EMI”. On click of this button, the result should be shown in a TextView. Also, calculate the EMI by varying the Loan Term and Interest Rate values.



--	--



**Laboratory Outcomes:** After studying these laboratory programs, students will be able to

- Create, test and debug Android application by setting up Android development environment.
- Implement adaptive, responsive user interfaces that work across a wide range of devices.
- Infer long running tasks and background work in Android applications.
- Demonstrate methods in storing, sharing and retrieving data in Android applications.
- Infer the role of permissions and security for Android applications.

**Procedure to Conduct Practical Examination**

- Experiment distribution
  - For laboratories having only one part: Students are allowed to pick one experiment from the lot with equal opportunity.
  - For laboratories having PART A and PART B: Students are allowed to pick one experiment from PART A and one experiment from PART B, with equal opportunity.
- Change of experiment is allowed only once and marks allotted for procedure to be made zero of the changed part only.
- Marks Distribution (Course to change in accordance with university regulations)
  - For laboratories having only one part – Procedure + Execution + Viva-Voce: 15+70+15= 100 Marks
  - For laboratories having PART A and PART B
    - i. Part A – Procedure + Execution + Viva = 6 + 28 + 6 = 40 Marks
    - ii. Part B – Procedure + Execution + Viva = 9 + 42 + 9 = 60 Marks

**Text Books:**

1. Google Developer Training, "**Android Developer Fundamentals Course – Concept Reference**", Google Developer Training Team, 2017.  
<https://www.gitbook.com/book/google-developer-training/android-developer-fundamentals-course-concepts/details>  
(Download pdf file from the above link)

**Reference Books:**

1. Erik Hellman, "**Android Programming – Pushing the Limits**", 1<sup>st</sup> Edition, Wiley India Pvt Ltd, 2014. ISBN-13:978-8126547197
2. Dawn Griffiths and David Griffiths, "**Head First Android Development**", 1<sup>st</sup> Edition, O'Reilly SPD Publishers, 2015. ISBN-13:978-9352131341
3. Bill Phillips, Chris Stewart and Kristin Marsicano, "**Android Programming: The Big Nerd Ranch Guide**", 3<sup>rd</sup> Edition, Big Nerd Ranch Guides, 2017. ISBN-13:978-0134706054

## **Install and Set up Android Studio in Windows**

**Android Studio** is the official **IDE (Integrated Development Environment)** for Android app development and it is based on **JetBrains' IntelliJ IDEA** software. Android Studio provides many excellent features that enhance productivity when building Android apps, such as:

- A blended environment where one can develop for all Android devices
- Apply Changes to push code and resource changes to the running app without restarting the app
- A flexible Gradle-based build system
- A fast and feature-rich emulator
- GitHub and Code template integration to assist you to develop common app features and import sample code
- Extensive testing tools and frameworks
- C++ and NDK support
- Built-in support for Google Cloud Platform, making it easy to integrate Google Cloud Messaging and App Engine, and many more.

### **System Requirements**

- Microsoft Windows 7/8/10 (32-bit or 64-bit)
- 4 GB RAM minimum, 8 GB RAM recommended (plus 1 GB for the Android Emulator)
- 2 GB of available disk space minimum, 4 GB recommended (500 MB for IDE plus 1.5 GB for Android SDK and emulator system image)
- 1280 x 800 minimum screen resolution

## **Installation Guide**

**Step 1:** Head over to <https://developer.android.com/studio/#downloads> to get the Android Studio executable or zip file.

**Step 2:** Click on the **Download Android Studio** Button. Click on the “I have read and agree with the above terms and conditions” checkbox followed by the download button.

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

**Step 4:** It will start the installation

**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’.

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

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

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

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

## Program-1

Create an application to design a Visiting Card. The Visiting card should have a company logo at the top right corner. The company name should be displayed in Capital letters, aligned to the center. Information like the name of the employee, job title, phone number, address, email, fax and the website address is to be displayed. Insert a horizontal line between the job title and the phone number.



### XML CODE

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

    <TextView
        android:layout_width="220dp"
        android:layout_height="wrap_content"
        android:layout_alignParentEnd="true"
        android:layout_alignParentBottom="true"
        android:layout_marginEnd="12dp"
        android:layout_marginBottom="632dp"
        android:text="CIT-GUBBI"
        android:textColor="#051449"
        android:textSize="40sp"
        android:textStyle="bold|italic" />

    <View
        android:id="@+id/view"
        android:layout_width="wrap_content"
        android:layout_height="12dp"
        android:layout_alignParentBottom="true"
        android:layout_marginBottom="584dp"
        android:background="@color/purple_700" />
```

**<ImageView**

```
    android:id="@+id/imageView2"  
    android:layout_width="wrap_content"  
    android:layout_height="88dp"  
    android:layout_alignParentEnd="true"  
    android:layout_alignParentBottom="true"  
    android:layout_marginEnd="249dp"  
    android:layout_marginBottom="611dp"  
    app:srcCompat="@drawable/logo" />
```

**<TextView**

```
    android:id="@+id/textView3"  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:layout_alignParentEnd="true"  
    android:layout_alignParentBottom="true"  
    android:layout_marginEnd="175dp"  
    android:layout_marginBottom="493dp"  
    android:text="SUHAS K C"  
    android:textColor="#021C1B"  
    android:textSize="36sp"  
    android:textStyle="bold|italic" />
```

**<TextView**

```
    android:id="@+id/textView4"  
    android:layout_width="318dp"  
    android:layout_height="56dp"  
    android:layout_alignParentEnd="true"  
    android:layout_alignParentBottom="true"  
    android:layout_marginEnd="38dp"  
    android:layout_marginBottom="430dp"  
    android:text="Assitant Professor"  
    android:textAlignment="viewStart"  
    android:textColor="#2BC5A6"  
    android:textSize="30sp"  
    android:textStyle="bold|italic" />
```

**<TextView**

```
    android:id="@+id/textView5"  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:layout_alignParentEnd="true"  
    android:layout_alignParentBottom="true"  
    android:layout_marginEnd="193dp"  
    android:layout_marginBottom="398dp"
```

```
android:text="Dept. of CSE"  
android:textColor="#3B9713"  
android:textSize="30sp"  
android:textStyle="bold|italic" />
```

```
<TextView  
android:id="@+id/textView6"  
android:layout_width="wrap_content"  
android:layout_height="wrap_content"  
android:layout_alignParentEnd="true"  
android:layout_alignParentBottom="true"  
android:layout_marginEnd="40dp"  
android:layout_marginBottom="354dp"  
android:text="#49, Ananda Nilaya,10th cross"  
android:textColor="#1E0877"  
android:textSize="24sp"  
android:textStyle="bold|italic" />
```

```
<TextView  
android:id="@+id/textView7"  
android:layout_width="336dp"  
android:layout_height="69dp"  
android:layout_alignParentEnd="true"  
android:layout_alignParentBottom="true"  
android:layout_marginEnd="25dp"  
android:layout_marginBottom="286dp"  
android:text="Sapthagiri Extension, Tumkur-572102"  
android:textColor="#0B0459"  
android:textSize="24sp"  
android:textStyle="bold|italic" />
```

```
<TextView  
android:id="@+id/textView8"  
android:layout_width="239dp"  
android:layout_height="48dp"  
android:layout_alignParentEnd="true"  
android:layout_alignParentBottom="true"  
android:layout_marginEnd="119dp"  
android:layout_marginBottom="239dp"  
android:text="Mob: 9844987877"  
android:textColor="#4E074A"  
android:textSize="24sp"  
android:textStyle="bold|italic" />
```

```
<TextView  
android:id="@+id/textView9"
```

```
android:layout_width="312dp"  
android:layout_height="48dp"  
android:layout_alignParentEnd="true"  
android:layout_alignParentBottom="true"  
android:layout_marginEnd="45dp"  
android:layout_marginBottom="206dp"  
android:text="Email: suhas@cittumkur.org"  
android:textColor="#49312B"  
android:textSize="24sp"  
android:textStyle="bold|italic" />  
</RelativeLayout>
```

**Java- Code:**

```
package com.example.visitingcard;  
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:**

## Program-2:

Develop an Android application using controls like Button, TextView, EditText for designing a calculator having basic functionality like Addition, Subtraction, Multiplication, and Division

### XML-CODE

```
<?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:background="@color/teal_200"
    android:backgroundTint="#4AA7CC"
    tools:context=".MainActivity">
```

#### <EditText

```
    android:id="@+id/ed_num1"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_alignParentEnd="true"
    android:layout_alignParentBottom="true"
    android:layout_marginEnd="75dp"
    android:layout_marginBottom="546dp"
    android:backgroundTint="#C62626"
    android:ems="10"
    android:hint="Enter number 1"
    android:inputType="textPersonName"
    android:textSize="24sp" />
```

#### <EditText

```
    android:id="@+id/ed_num2"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_alignParentEnd="true"
    android:layout_alignParentBottom="true"
    android:layout_marginEnd="74dp"
    android:layout_marginBottom="462dp"
    android:backgroundTint="#C32929"
    android:ems="10"
```



```
android:hint="Enter number 2"  
android:inputType="textPersonName"  
android:textSize="24sp" />
```

```
<TextView  
    android:id="@+id/res"  
    android:layout_width="191dp"  
    android:layout_height="wrap_content"  
    android:layout_alignParentEnd="true"  
    android:layout_alignParentBottom="true"  
    android:layout_marginEnd="125dp"  
    android:layout_marginBottom="356dp"  
    android:background="#FFFFFF"  
    android:backgroundTint="#F2F3F1"  
    android:text="0" android:textAlignment="center"  
    android:textColor="#44230202"  
    android:textSize="36sp"  
    android:textStyle="bold|italic" />
```

```
<Button  
    android:id="@+id/btn1"  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:layout_alignParentEnd="true"  
    android:layout_alignParentBottom="true"  
    android:layout_marginEnd="263dp"  
    android:layout_marginBottom="208dp"  
    android:backgroundTint="#EA5858"  
    android:text="Add"  
    android:onClick="Add"  
    android:textSize="24sp" />
```

```
<Button  
    android:id="@+id/btn2"  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:layout_alignParentEnd="true"  
    android:layout_alignParentBottom="true"  
    android:layout_marginEnd="108dp"  
    android:layout_marginBottom="209dp"
```

```
android:backgroundTint="#E85A5A"  
android:text="Sub"  
android:onClick="Sub"  
android:textSize="24sp" />
```

<Button

```
android:id="@+id/btn3"  
android:layout_width="wrap_content"  
android:layout_height="wrap_content"  
android:layout_alignParentEnd="true"  
android:layout_alignParentBottom="true"  
android:layout_marginEnd="260dp"  
android:layout_marginBottom="125dp"  
android:backgroundTint="#E85A5A"  
android:text="Mul"  
android:onClick="Mul"  
android:textSize="24sp" />
```

<Button

```
android:id="@+id/btn4"  
android:layout_width="wrap_content"  
android:layout_height="wrap_content"  
android:layout_alignParentEnd="true"  
android:layout_alignParentBottom="true"  
android:layout_marginEnd="107dp"  
android:layout_marginBottom="125dp"  
android:backgroundTint="#E85A5A"  
android:text="Div"  
android:onClick="Div"  
android:textSize="24sp" />
```

<TextView

```
android:id="@+id/textView2"  
android:layout_width="389dp"  
android:layout_height="45dp"  
android:layout_alignParentEnd="true"  
android:layout_alignParentBottom="true"  
android:layout_marginEnd="15dp"  
android:layout_marginBottom="681dp"  
android:background="#C61717"
```

```
        android:text="CalculatorApplication"  
        android:textAlignment="center"  
        android:textColor="#F3F4FA"  
        android:textSize="36sp"  
        android:textStyle="bold|italic" />  
</RelativeLayout>
```

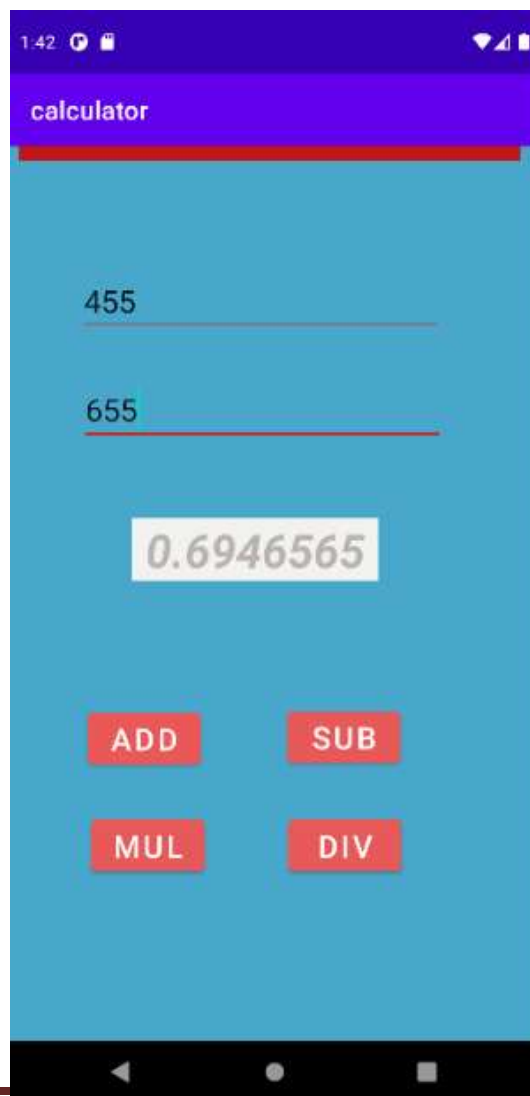
### JAVA-CODE

```
package com.example.calculator;  
import androidx.appcompat.app.AppCompatActivity;  
  
import android.os.Bundle;  
import android.view.View;  
import android.widget.EditText;  
import android.widget.TextView;  
  
public class MainActivity extends AppCompatActivity {  
    EditText e1, e2;  
    TextView result;  
    @Override  
    protected void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
        setContentView(R.layout.activity_main);  
        e1 = findViewById(R.id.ed_num1);  
        e2 = findViewById(R.id.ed_num2);  
        result = findViewById(R.id.res);  
    }  
  
    public void Add(View view) {  
        int a1 = Integer.parseInt(e1.getText().toString());  
        int a2 = Integer.parseInt(e2.getText().toString());  
        int res = a1 + a2;  
        result.setText(""+res);  
    }  
  
    public void Sub(View view) {  
        int a1 = Integer.parseInt(e1.getText().toString());  
        int a2 = Integer.parseInt(e2.getText().toString());  
        int res = a1 - a2;  
        result.setText(""+res);  
    }  
}
```

```
public void Mul(View view) {  
    int a1 = Integer.parseInt(e1.getText().toString());  
    int a2 = Integer.parseInt(e2.getText().toString());  
    int res = a1 * a2;  
    result.setText(""+res);  
}
```

```
public void Div(View view) {  
    float a1 = Integer.parseInt(e1.getText().toString());  
    float a2 = Integer.parseInt(e2.getText().toString());  
    float res = a1 /a2;  
    result.setText(""+res);  
}  
}
```

## Output:



### Program-3

3. Create a SIGN Up activity with Username and Password. Validation of password should happen based on the following rules:

- Password should contain uppercase and lowercase letters.
- Password should contain letters and numbers.
- Password should contain special characters.
- Minimum length of the password (the default value is 8).

On successful SIGN UP proceed to the next Login activity.

Here the user should SIGN IN using the Username and Password created during signup activity.

If the Username and Password are matched then navigate to the next activity which displays a message saying “Successful Login” or else display a toast message saying “Login Failed”.

The user is given only two attempts and after that display a toast message saying “Failed Login Attempts” and disable the SIGN IN button. Use Bundle to transfer information from one activity to another.

### Signup Activity.xml

#### activity\_main.xml

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

    <TextView
        android:id="@+id/textView2"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignParentEnd="true"
        android:layout_alignParentBottom="true"
        android:layout_marginEnd="94dp"
        android:layout_marginBottom="640dp"
        android:text="SIGNUP ACTIVITY"
        android:textAlignment="center"
        android:textColor="@color/purple_700"
        android:textSize="24sp"
        android:textStyle="bold" />
```

```
<EditText
    android:id="@+id/emaile"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_alignParentEnd="true"
    android:layout_alignParentBottom="true"
    android:layout_marginEnd="26dp"
    android:layout_marginBottom="525dp"
    android:ems="10"
    android:hint="Email"
    android:inputType="textEmailAddress"
    android:textAlignment="center"
    android:textColor="#800E0E"
    android:textSize="30dp"
    android:textStyle="bold" />
```

```
<EditText
    android:id="@+id/Passworde"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_alignParentEnd="true"
    android:layout_alignParentBottom="true"
    android:layout_marginEnd="21dp"
    android:layout_marginBottom="411dp"
    android:ems="10"
    android:hint="Password"
    android:inputType="textPassword"
    android:textAlignment="center"
    android:textSize="30dp"
    android:textStyle="bold" />
```

```
<Button
    android:id="@+id/signupb"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_alignParentEnd="true"
    android:layout_alignParentBottom="true"
    android:layout_marginEnd="125dp"
    android:layout_marginBottom="246dp"
    android:text="SignUp"
    android:textAlignment="center"
    android:textSize="30dp"
    android:textStyle="bold" />
```

```
</RelativeLayout>
```

## MainActivity.java

```

package com.example.loginactivity;

import androidx.appcompat.app.AppCompatActivity;

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

import java.util.regex.Pattern;

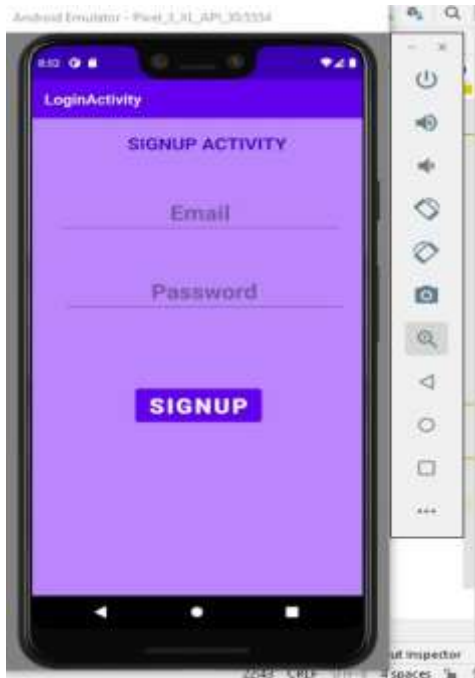
public class MainActivity extends AppCompatActivity {
    EditText emailE,passwordE;
    Button signupb;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        emailE=findViewById(R.id.emailE);
        passwordE=findViewById(R.id.PasswordE);
        signupb=findViewById(R.id.signupb);
        signupb.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                String email=emailE.getText().toString();
                String password=passwordE.getText().toString();
                if(!isValidpassword(password))
                {
                    Toast.makeText(MainActivity.this,"Password does not match the
rules",Toast.LENGTH_LONG).show();
                    return;
                }
                Intent intent = new Intent(MainActivity.this,LoginActivity.class);
                intent.putExtra("email",email);
                intent.putExtra("password",password);
                startActivity(intent);
            }
        });
    }
    Pattern lowercase=Pattern.compile("^.*[a-z].*$");
    Pattern uppercase= Pattern.compile("^.*[A-Z].*$");

```

```
Pattern number=Pattern.compile("^.*[0-9].*$");
Pattern specialcharacters=Pattern.compile("^.*[^a-zA-Z0-9].*$");
private Boolean isvalidpassword(String password){
    if(password.length(<8)
    {
        return false;
    }
    if(!lowercase.matcher(password).matches()){
        return false;
    }
    if(!uppercase.matcher(password).matches()){
        return false;
    }
    if(!number.matcher(password).matches()){
        return false;
    }
    if(!specialcharacters.matcher(password).matches()){
        return false;
    }
    return true;
}
}
```

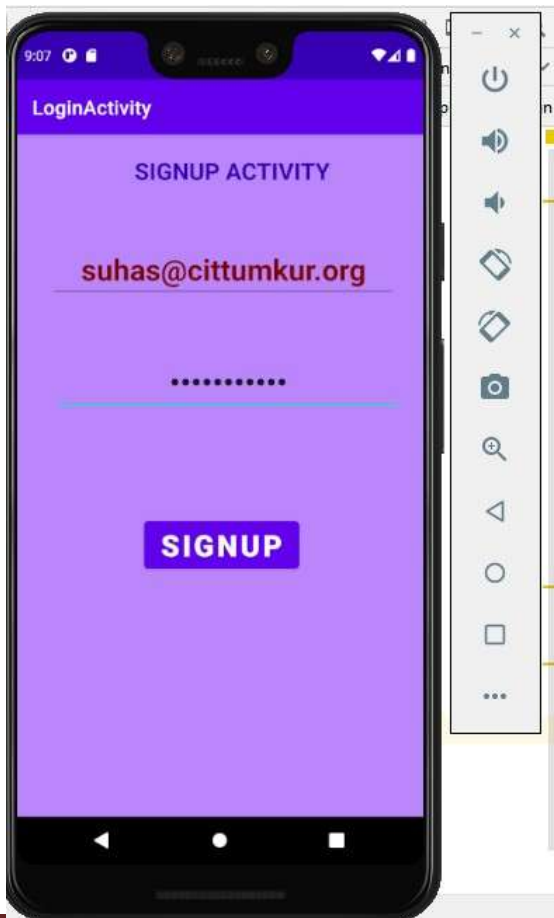




**Initial Signup Activity**



**Password Rule Incorrect**



**Correct Signup will take to login page**

## LOGIN ACTIVITY PAGE

### activity\_login.xml

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="@color/teal_200"
    tools:context=".LoginActivity">

    <TextView
        android:id="@+id/textView"
        android:layout_width="336dp"
        android:layout_height="wrap_content"
        android:layout_alignParentEnd="true"
        android:layout_alignParentBottom="true"
        android:layout_marginEnd="33dp"
        android:layout_marginBottom="648dp"
        android:text="LOGIN ACTIVITY"
        android:textAlignment="center"
        android:textColor="@color/purple_700"
        android:textSize="24sp"
        android:textStyle="bold|italic" />

    <EditText
        android:id="@+id/emailee"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignParentEnd="true"
        android:layout_alignParentBottom="true"
        android:layout_marginEnd="55dp"
        android:layout_marginBottom="544dp"
        android:ems="10"
        android:hint="Email"
        android:inputType="textEmailAddress"
        android:textAlignment="center"
        android:textColor="@color/black"
        android:textSize="30dp"
        android:textStyle="italic|bold" />
```

```
<EditText
    android:id="@+id/passwordee"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_alignParentEnd="true"
    android:layout_alignParentBottom="true"
    android:layout_marginEnd="57dp"
    android:layout_marginBottom="433dp"
    android:ems="10"
    android:hint="Password"
    android:inputType="textPassword"
    android:textAlignment="center"
    android:textColor="@color/black"
    android:textSize="30dp"
    android:textStyle="bold|italic" />
```

```
<Button
    android:id="@+id/loginb2"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_alignParentEnd="true"
    android:layout_alignParentBottom="true"
    android:layout_marginEnd="176dp"
    android:layout_marginBottom="241dp"
    android:textColor="@color/black"
    android:textSize="30dp"
    android:textStyle="bold|italic"
    android:textAlignment="center"
    android:text="LOGIN" />
```

```
</RelativeLayout>
```

### LoginActivity.java

```
package com.example.loginactivity;

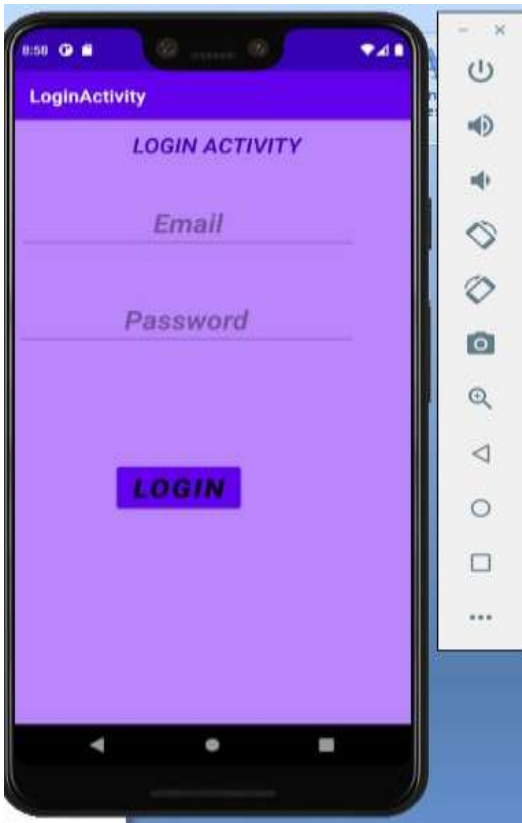
import androidx.activity.result.contract.ActivityResultContracts;
import androidx.appcompat.app.AppCompatActivity;

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 LoginActivity extends AppCompatActivity {
    EditText email1,password1;
    Button loginb1;
    int counter=2;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_login);
        email1= findViewById(R.id.email1);
        password1=findViewById(R.id.password1);
        loginb1=findViewById(R.id.loginb1);
        String registeredmail=getIntent().getStringExtra("email");
        String registeredpassword=getIntent().getStringExtra("password");
        loginb1.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                String email = email1.getText().toString();
                String password=password1.getText().toString();
                if(registeredmail.equals(email) && registeredpassword.equals(password)){
                    Intent intent= new Intent(LoginActivity.this,LoginSuccessful.class);
                    startActivity(intent);
                }
                else{
                    Toast.makeText(LoginActivity.this,"Invalid
Credential",Toast.LENGTH_LONG).show();
                }
                counter--;
                if(counter==0)
                {
                    Toast.makeText(getApplicationContext(),"Failed Login
Attempts",Toast.LENGTH_LONG).show();
                    loginb1.setEnabled(false);
                }
            }
        });
    }
}

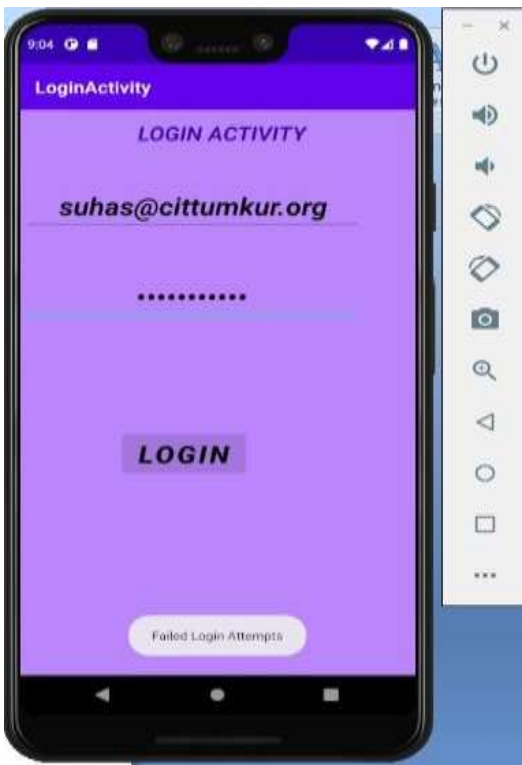
```



Initial Login Activity



Wrong Password



**Failed Login after 2 Attempts**

**Correct Login will take to Successful Page**

## SUCCESSFUL Activity PAGE

### activity\_login\_successful.xml

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="@color/teal_200"
    tools:context=".LoginSuccessful">

    <TextView
        android:id="@+id/textView3"
        android:layout_width="371dp"
        android:layout_height="68dp"
        android:layout_alignParentEnd="true"
        android:layout_alignParentBottom="true"
        android:layout_marginEnd="38dp"
        android:layout_marginBottom="433dp"
        android:text="Login Successful"
        android:textAlignment="center"
        android:textColor="@color/black"
        android:textSize="36dp"
        android:textStyle="italic|bold" />
</RelativeLayout>
```

### LoginSuccessful.java

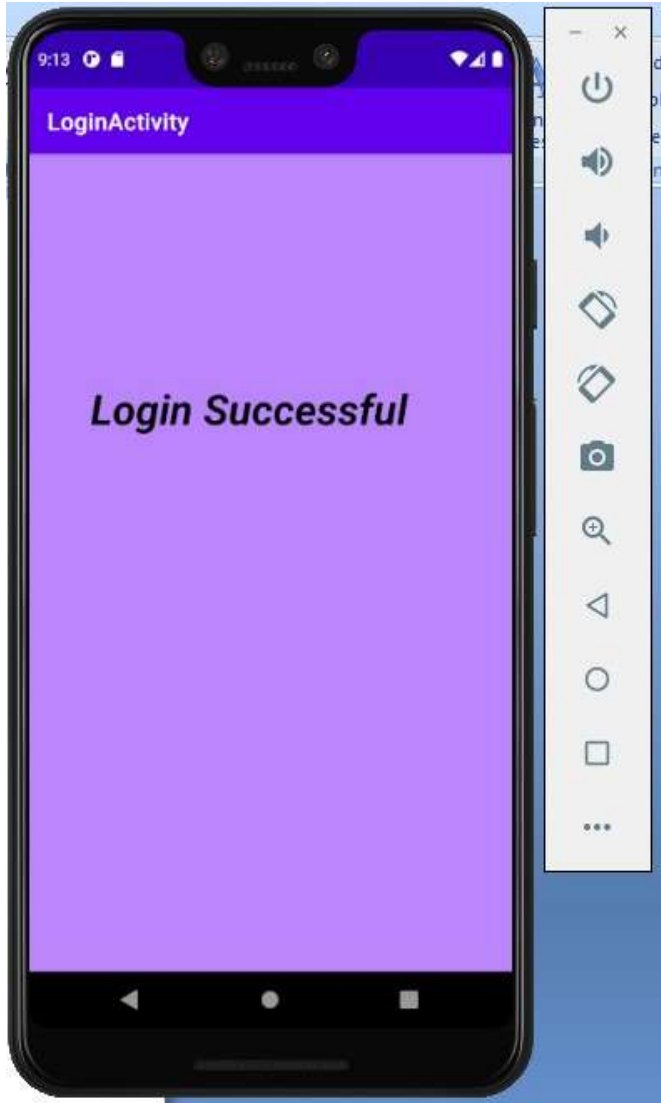
```
package com.example.loginactivity;

import androidx.appcompat.app.AppCompatActivity;

import android.os.Bundle;

public class LoginSuccessful extends AppCompatActivity {

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



**Successful Login Message**



## Program-4

**Develop an application to set an image as wallpaper. On click of a button, the wallpaper image should start to change randomly every 30 seconds.**

- 1) Firstly Create an Application by Name "WallpaperActivity"
- 2) Go to xml code of design change the layout to "RelativeLayout"
- 3) Add TextView component & change the following properties:
  - Size: 38dp
  - Text: Wall Paper Change Application
  - Center-Align
- 4) Add Button component & change the following properties:
  - Size: 38dp
  - Text: Click Here To Change Wall Paper
- 5) Save five images (jpg format) in the drawable folder.

In this example one.jpg, two.jpg, three.jpg, four.jpg and five.jpg images are saved in drawable folder.

### activity\_main.xml

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

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignParentEnd="true"
        android:layout_alignParentBottom="true"
        android:layout_marginEnd="13dp"
        android:layout_marginBottom="580dp"
```

```
android:text="Changing Wallpaper Application"  
android:textAlignment="center"  
android:textColor="@color/black"  
android:textSize="30dp"  
app:layout_constraintBottom_toBottomOf="parent"  
app:layout_constraintLeft_toLeftOf="parent"  
app:layout_constraintRight_toRightOf="parent"  
app:layout_constraintTop_toTopOf="parent" />
```

<Button

```
android:id="@+id/button"  
android:layout_width="wrap_content"  
android:layout_height="wrap_content"  
android:layout_alignParentEnd="true"  
android:layout_alignParentBottom="true"  
android:layout_marginEnd="20dp"  
android:layout_marginBottom="344dp"  
android:text="Click here to change Wallpaper"  
android:textAlignment="center"  
android:textColor="@color/black"  
android:textSize="30dp"  
android:textStyle="bold" />
```

</RelativeLayout>

### MainActivity.java

```
package com.example.wallpaper;  
  
import androidx.appcompat.app.AppCompatActivity;  
  
import android.annotation.SuppressLint;  
import android.app.WallpaperManager;  
import android.graphics.Bitmap;  
import android.graphics.drawable.BitmapDrawable;  
import android.graphics.drawable.Drawable;  
import android.os.Bundle;  
import android.view.View;  
import android.widget.Button;  
  
import java.io.IOException;  
import java.util.Timer;  
import java.util.TimerTask;
```

```

public class MainActivity extends AppCompatActivity {
    Button changewallpaper;
    Timer mytimer;
    Drawable drawable;
    WallpaperManager wpm;
    int prev=1;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        mytimer= new Timer();
        wpm=WallpaperManager.getInstance(this);
        changewallpaper=findViewById(R.id.button);
        changewallpaper.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                setWallpaper();
            }
        });
    }
    private void setWallpaper()
    {
        mytimer.schedule(new TimerTask() {

            @Override
            public void run() {
                if(prev==1)
                {
                    drawable=getResources().getDrawable(R.drawable.one);
                    prev=2;
                }
                else if(prev==2)
                {
                    drawable=getResources().getDrawable(R.drawable.two);
                    prev=3;
                }
                else if(prev==3)
                {
                    drawable=getResources().getDrawable(R.drawable.three);
                    prev=4;
                }
                else if(prev==4)
                {
                    drawable=getResources().getDrawable(R.drawable.four);
                    prev=5;
                }
            }
        });
    }
}

```

```
    }  
    else if(prev==5)  
    {  
        drawable=getResources().getDrawable(R.drawable.five);  
        prev=1;  
    }  
    Bitmap wallpaper = ((BitmapDrawable)drawable).getBitmap();  
    try {  
        wpm.setImageBitmap(wallpaper);  
    } catch (IOException e) {  
        e.printStackTrace();  
    }  
    }  
    },10,10000);  
    }  
}
```

### Output:



## Program-5

**Write a program to create an activity with two buttons START and STOP. On pressing of the START button, the activity must start the counter by displaying the numbers from one and the counter must keep on counting until the STOP button is pressed. Display the counter value in a Text View control.**

- 1) Firstly Create an Application by Name “CounterActivity”
- 2) Go to xml code of design change the layout to “RelativeLayout”
- 3) Add TextView component & change the following properties:
  - Size: 38dp
  - Text: “Counter Application”
  - Center-Align
- 4) Add TextView component & change the following properties:
  - Text: “Counter value”
  - id: “@+id/textcounter”
- 5) Add Button components & change the following properties:
  - Size: 38dp
  - Text: Start
  - id: “@+id/btnstart”
- 6) Add Button components & change the following properties:
  - Size: 38dp
  - Text: Stop
  - id: “@+id/btnstop”

### activity\_main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:background="@color/teal_200"
```

```
android:layout_height="match_parent"  
tools:context=".MainActivity">
```

```
<TextView
```

```
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:layout_alignParentEnd="true"  
    android:layout_alignParentBottom="true"  
    android:layout_marginEnd="74dp"  
    android:layout_marginBottom="581dp"  
    android:text="Counter Application!"  
    android:textColor="@color/black"  
    android:textSize="30dp"  
    android:textStyle="bold"  
    app:layout_constraintBottom_toBottomOf="parent"  
    app:layout_constraintLeft_toLeftOf="parent"  
    app:layout_constraintRight_toRightOf="parent"  
    app:layout_constraintTop_toTopOf="parent" />
```

```
<TextView
```

```
    android:id="@+id/textcounter"  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:layout_alignParentEnd="true"  
    android:layout_alignParentBottom="true"  
    android:layout_marginEnd="132dp"  
    android:layout_marginBottom="451dp"  
    android:hint="Counter Value"  
    android:text=""  
    android:textColor="@color/black"  
    android:textSize="30dp"  
    android:textStyle="bold" />
```

```
<Button
```

```
    android:id="@+id/btnstart"  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:layout_alignParentEnd="true"  
    android:layout_alignParentBottom="true"  
    android:layout_marginEnd="241dp"  
    android:layout_marginBottom="271dp"  
    android:text="START"  
    android:textAlignment="center"  
    android:textColor="@color/black"  
    android:textSize="30dp"  
    android:textStyle="bold" />
```

```

<Button
    android:id="@+id/btnstop"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_alignParentEnd="true"
    android:layout_alignParentBottom="true"
    android:layout_marginEnd="53dp"
    android:layout_marginBottom="268dp"
    android:text="STOP"
    android:textAlignment="center"
    android:textColor="@color/black"
    android:textSize="30dp"
    android:textStyle="bold" />

```

```
</RelativeLayout>
```

### Mainactivity.java

```

package com.example.program5;

import androidx.appcompat.app.AppCompatActivity;

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

public class MainActivity extends AppCompatActivity {
    Button start,stop;
    TextView tc;
    int i=1;
    Handler customhandler = new Handler();

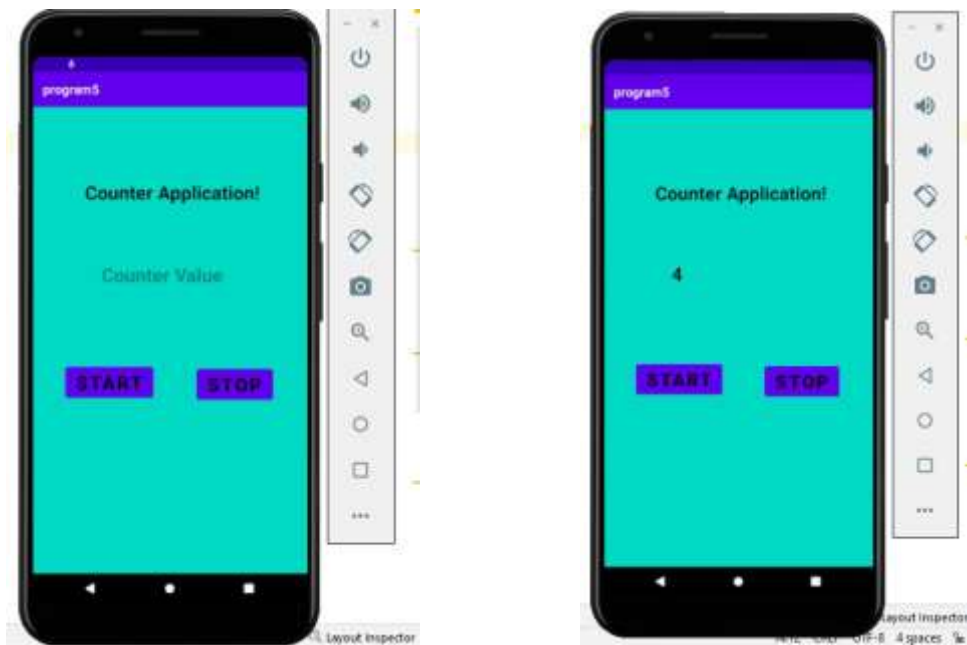
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        start=findViewById(R.id.btnstart);
        stop=findViewById(R.id.btnstop);
        tc=findViewById(R.id.textcounter);

        start.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {

```

```
        customhandler.postDelayed(updatetimerthread,0);
    }
});
stop.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        customhandler.removeCallbacks(updatetimerthread);
    }
});
}
private final Runnable updatetimerthread = new Runnable() {
    @Override
    public void run() {
        tc.setText(""+i);
        customhandler.postDelayed(this,1000);
        i++;
    }
};
}
```

## OUTPUT





## Program-6

**6. Create two files of XML and JSON type with values for City Name, Latitude, Longitude, Temperature, and Humidity. Develop an application to create an activity with two buttons to parse the XML and JSON files which when clicked should display the data in their respective layouts side by side.**

- 1) Firstly Create an Application by Name “JsonParser”
- 2) Go to xml code of design change the layout to “RelativeLayout”
- 3) Add TextView component & change the following properties:
  - a. Size: 38dp
  - b. Text: XML and JSON Parser
  - c. Center-Align
- 4) Add Two Buttons to Design & change the name “ParseXml” & “ParseJson” with following onclick functions:
  - ParseXml-Button:btn xmlparse
  - ParseJson-Button: btnjsonparse
- 5) Add TextView component & change the following properties:
  - Id: display
  - Text: “”
  - Align: Center
- 6) Add Assets folder by following the given hierarchy:

App->new->folder->Assests folder
- 7) Inside the assets folder create new files of xml and json using the following hierarchy:

new->file->city.xml

new->file->city.json

Once created place the following details inside the “city.xml” and “city.json”

city.xml	city.json
<pre> &lt;?xml version="1.0"?&gt; &lt;records&gt; &lt;place&gt;   &lt;name&gt;Mysore&lt;/name&gt;   &lt;lat&gt; 12.295 &lt;/lat&gt;   &lt;long&gt;76.639 &lt;/long&gt;   &lt;temperature&gt; 22 &lt;/temperature&gt;   &lt;humidity&gt; 90 % &lt;/humidity&gt; &lt;/place&gt;    &lt;place&gt;     &lt;name&gt;Bangalore&lt;/name&gt;     &lt;lat&gt; 12.97165 &lt;/lat&gt;     &lt;long&gt;77.5946 &lt;/long&gt;     &lt;temperature&gt; 25 &lt;/temperature&gt;     &lt;humidity&gt; 74 % &lt;/humidity&gt;   &lt;/place&gt; &lt;/records&gt; </pre>	<pre> [   {     "name": "Mysore ",     "lat": "14.295 ",     "long": "86.639 ",     "temperature": "20 ",     "humidity": "82 %"   },   {     "name": "Bangalore",     "lat": "15.97165 ",     "long": "87.5946 ",     "temperature": "21 ",     "humidity": "84 %"   } ] </pre>
<p>Note: 1. Add Assets folder by following the given hierarchy: App-&gt;new-&gt;folder-&gt;Assets folder  2. Inside the assets folder create new files of xml and json using the following hierarchy:</p> <ul style="list-style-type: none"> <li>• new-&gt;file-&gt;city.xml</li> <li>• new-&gt;file-&gt;city.json</li> </ul>	

### activity\_main.xml

```

<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="#C6BBC3"
    tools:context=".MainActivity">

    <TextView
        android:id="@+id/textView"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignParentEnd="true"
        android:layout_alignParentBottom="true"
        android:layout_marginEnd="81dp"
        android:layout_marginBottom="657dp"
        android:text="Parser Application"
        android:textColor="@color/black"
        android:textSize="30dp"
        android:textStyle="bold" />

```

```
<Button
    android:id="@+id/buttonxml"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_alignParentEnd="true"
    android:layout_alignParentBottom="true"
    android:layout_marginEnd="104dp"
    android:layout_marginBottom="546dp"
    android:text="XML parser"
    android:textColor="@color/black"
    android:textSize="26dp"
    android:textStyle="bold" />

<Button
    android:id="@+id/buttonjson"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_alignParentEnd="true"
    android:layout_alignParentBottom="true"
    android:layout_marginEnd="88dp"
    android:layout_marginBottom="602dp"
    android:text="JSON Parser"
    android:textColor="@color/black"
    android:textSize="26dp"
    android:textStyle="bold" />

<TextView
    android:id="@+id/display_result"
    android:layout_width="308dp"
    android:layout_height="376dp"
    android:layout_alignParentEnd="true"
    android:layout_alignParentBottom="true"
    android:layout_marginEnd="41dp"
    android:layout_marginBottom="101dp"
    android:hint="Result"
    android:textStyle="bold"
    android:text=""
    android:textAlignment="center"
    android:textColor="@color/black"
    android:textSize="20dp" />

</RelativeLayout>
```

**MainActivity.java**

```

package com.example.parse;
import androidx.appcompat.app.AppCompatActivity;
import android.os.Bundle;
import android.view.View;
import android.widget.TextView;
import android.widget.Button;
import android.widget.Toast;
import org.json.JSONArray;
import org.json.JSONObject;
import org.w3c.dom.Document;
import org.w3c.dom.Element;
import org.w3c.dom.Node;
import org.w3c.dom.NodeList;
import java.io.InputStream;
import java.nio.charset.StandardCharsets;
import javax.xml.parsers.DocumentBuilder;
import javax.xml.parsers.DocumentBuilderFactory;
public class MainActivity extends AppCompatActivity {
    TextView display;
    Button btnxmlparse, btnjsonparse;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        display = (TextView)findViewById(R.id.display_result);
        btnxmlparse=findViewById(R.id.buttonxml);
        btnjsonparse=findViewById(R.id.buttonjson);
        btnxmlparse.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                try {
                    InputStream is = getAssets().open("city.xml");
                    DocumentBuilderFactory documentBuilderFactory =
DocumentBuilderFactory.newInstance();
                    DocumentBuilder documentBuilder =
documentBuilderFactory.newDocumentBuilder();
                    Document document = documentBuilder.parse(is);
                    StringBuilder stringBuilder = new StringBuilder();
                    stringBuilder.append("XML DATA");
                    stringBuilder.append("\n -----");
                    NodeList nodeList = document.getElementsByTagName("place");
                    for (int i = 0; i < nodeList.getLength(); i++)
                    {
                        Node node = nodeList.item(i);
                        if (node.getNodeType() == Node.ELEMENT_NODE) {

```

```

        Element element = (Element) node;
        stringBuilder.append("\nName: ").append(getValue("name", element));
        stringBuilder.append("\nLatitude: ").append(getValue("lat", element));
        stringBuilder.append("\nLongitude: ").append(getValue("long", element));
        stringBuilder.append("\nTemperature: ").append(getValue("temperature",
element));
        stringBuilder.append("\nHumidity: ").append(getValue("humidity",
element));
        stringBuilder.append("\n -----");
    }
}
display.setText(stringBuilder.toString());
}
catch (Exception e)
{
    e.printStackTrace();
    Toast.makeText(MainActivity.this, "Error Parsing
XML", Toast.LENGTH_LONG).show();
}
}
});
btnjsonp.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        String json;
        StringBuilder stringBuilder = new StringBuilder();
        try {
            InputStream is = getAssets().open("city.json");
            int size = is.available();
            byte[] buffer = new byte[size];
            is.read(buffer);
            json = new String(buffer, StandardCharsets.UTF_8);
            JSONArray jsonArray = new JSONArray(json);
            stringBuilder.append("JSON DATA");
            stringBuilder.append("\n -----");
            for (int i = 0; i < jsonArray.length(); i++) {
                JSONObject jsonObject = jsonArray.getJSONObject(i);
                stringBuilder.append("\nName: ").append(jsonObject.getString("name"));
                stringBuilder.append("\nLatitude: ").append(jsonObject.getString("lat"));
                stringBuilder.append("\nLongitude: ").append(jsonObject.getString("long"));
                stringBuilder.append("\nTemperature:
").append(jsonObject.getString("temperature"));
                stringBuilder.append("\nHumidity:
").append(jsonObject.getString("humidity"));
                stringBuilder.append("\n -----");
            }

```

```

        display.setText(stringBuilder.toString());
        is.close();
    }
    catch (Exception e)
    {
        e.printStackTrace();
        Toast.makeText(MainActivity.this, "Error
inreading", Toast.LENGTH_LONG).show();
    }
}
});
}
private String getValue(String tag, Element element)
{
    return
element.getElementsByTagName(tag).item(0).getChildNodes().item(0).getNodeValue();
}
}

```

**OUTPUT:**

## Program-7

**Develop a simple application with one EditText so that the user can write some text in it. Create a button called “Convert Text to Speech” that converts the user input text into voice.**

- 1) Create an Application by Name “TextToSpeech”
- 2) Go to xml code of design change the layout to “RelativeLayout”
- 3) Add TextView component & change the following properties:
  - Size: 38dp
  - Text: Text2Speech App
  - Center-Align
- 4) Add PlainText(EditText) component & change the following properties in XML Code:
  - Text: “”
  - Hint: “Enter the text to be converted”
  - id: “@+id/editText”
- 5) Add Button component & change the following properties in XML Code:
  - Text: “Text To Speech convert”
  - Id:”@+/button “

### **Output: To listen the speech in your Mobile**

Extracting an APK file from Android Studio

Why APK?

While developing an Android app, you would usually run it on a physical device or an emulator. If you want to share it with someone for their feedback, you would share an APK that can easily be installed on any Android device.

How you can extract an APK file using Android Studio?

1. In the Android menu, go to **Build > Build Bundle(s) / APK (s) > Build APK(s)**.
2. Android Studio will start building the APK for you. Once done, a pop-up on the bottom right will notify you of its completion. Click the ‘**locate**’ button in this dialog.
3. The ‘**locate**’ button should open File Explorer with the **debug folder** open that contains a file called “**app-debug.apk**”.
4. Install in your mobile and check output

## Activity\_main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="#F2F2F2"
    tools:context=".MainActivity">

    <TextView
        android:id="@+id/textView"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignParentEnd="true"
        android:layout_alignParentBottom="true"
        android:layout_marginEnd="18dp"
        android:layout_marginBottom="634dp"
        android:text="TextToSpeech Application"
        android:textAlignment="center"
        android:textColor="@color/black"
        android:textSize="30dp"
        android:textStyle="bold" />

    <EditText
        android:id="@+id/editText"
        android:layout_width="297dp"
        android:layout_height="142dp"
        android:layout_alignParentEnd="true"
        android:layout_alignParentBottom="true"
        android:layout_marginEnd="61dp"
        android:layout_marginBottom="435dp"
        android:ems="10"
        android:hint="Enter the Text"
        android:inputType="textPersonName"
        android:text=""
        android:textAlignment="center"
        android:textColor="@color/black"
        android:textSize="30dp"
        android:textStyle="bold|italic" />

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



```

    android:layout_height="wrap_content"
    android:layout_alignParentEnd="true"
    android:layout_alignParentBottom="true"
    android:layout_marginEnd="38dp"
    android:layout_marginBottom="246dp"
    android:text="Text To Speech Converter"
    android:textAlignment="center"
    android:textColor="@color/black"
    android:textSize="30dp"
    android:textStyle="bold" />
</RelativeLayout>

```

### MainActivity.java

```

package com.example.text;

import androidx.appcompat.app.AppCompatActivity;

import android.os.Bundle;
import android.speech.tts.TextToSpeech;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.Toast;
import java.util.Locale;

public class MainActivity extends AppCompatActivity {
    TextToSpeech t1;
    EditText e1;
    Button b1;

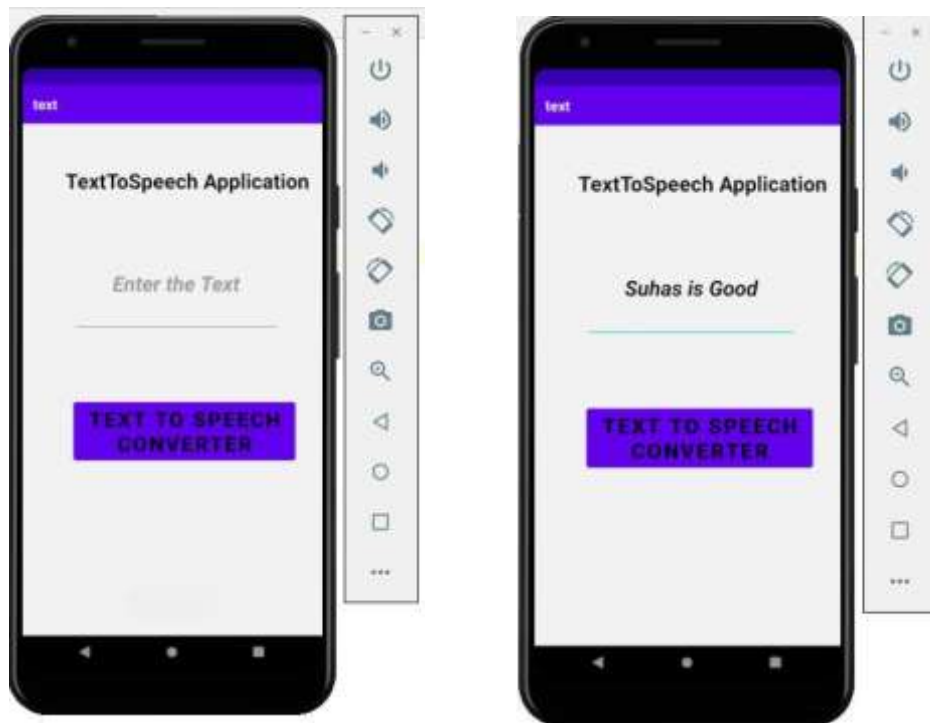
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        e1 = findViewById(R.id.editText);
        b1 = findViewById(R.id.button);
        t1 = new TextToSpeech(getApplicationContext(), new TextToSpeech.OnInitListener() {
            @Override
            public void onInit(int status) {
                if (status != TextToSpeech.ERROR) {
                    t1.setLanguage(Locale.UK);
                }
            }
        });
    }
}

```

```
b1.setOnClickListener(new View.OnClickListener() {  
    @Override  
    public void onClick(View v) {  
        String tospeak = e1.getText().toString();  
        Toast.makeText(getApplicationContext(),tospeak, Toast.LENGTH_SHORT).show();  
        t1.speak(tospeak, TextToSpeech.QUEUE_FLUSH, null);  
    }  
});  
}
```

## Output



## Program-8

**Create an activity like a phone dialer with CALL and SAVE buttons. On pressing the CALL button, it must call the phone number and on pressing the SAVE button it must save the number to the phone contacts**

- 1) Firstly Create an Application by Name "CallActivity"
- 2) Go to xml code of design change the layout to "RelativeLayout"
- 3) Add TextView component & change the following properties:
  - Size: 38dp
  - Text: Call Activity
  - Center-Align
- 4) Add Plaintext(EditText) component & change the following properties in XML Code:
  - id: "@+id/phoneNumberEditText"
- 5) Add three buttons to the design & change the text of the Buttons to "Clear", "Call", "Save" and change the id as follows:
  - id:"@+id/clearBtn"
  - id:"@+id/callBtn"
  - id:"@+id/saveBtn"
- 6) Add twelve buttons to the design & change the text of the Buttons as 1,2,3,4,5,6,7,8,9,0,\*,#

### XML-CODE

```
?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:background="#C1D6D1"
    tools:context=".MainActivity">

    <TextView
        android:layout_width="298dp"
        android:layout_height="wrap_content"
        android:layout_alignParentEnd="true"
        android:layout_alignParentBottom="true"
        android:layout_marginEnd="54dp"
        android:layout_marginBottom="575dp"
        android:text="Call Application"
        android:textColor="@color/black"
        android:textSize="40dp"
        android:textStyle="bold">
```

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

<EditText

```
android:id="@+id/phoneNumberEditText"  
android:layout_width="wrap_content"  
android:layout_height="wrap_content"  
android:layout_alignParentEnd="true"  
android:layout_alignParentBottom="true"  
android:layout_marginEnd="176dp"  
android:layout_marginBottom="462dp"  
android:ems="10"  
android:inputType="phone" />
```

<Button

```
android:id="@+id/clearBtn"  
android:layout_width="wrap_content"  
android:layout_height="wrap_content"  
android:layout_alignParentEnd="true"  
android:layout_alignParentBottom="true"  
android:layout_marginEnd="52dp"  
android:layout_marginBottom="459dp"  
android:text="Clear" />
```

<Button

```
android:id="@+id/button2"  
android:layout_width="76dp"  
android:layout_height="wrap_content"  
android:layout_alignParentEnd="true"  
android:layout_alignParentBottom="true"  
android:layout_marginEnd="282dp"  
android:layout_marginBottom="361dp"  
android:onClick="inputNumber"  
android:text="1" />
```

<Button

```
android:id="@+id/button3"  
android:layout_width="76dp"  
android:layout_height="wrap_content"  
android:layout_alignParentEnd="true"  
android:layout_alignParentBottom="true"  
android:layout_marginEnd="186dp"  
android:layout_marginBottom="363dp"  
android:onClick="inputNumber"  
android:text="2" />
```

<Button

```
android:id="@+id/button4"  
android:layout_width="76dp"  
android:layout_height="wrap_content"  
android:layout_alignParentEnd="true"  
android:layout_alignParentBottom="true"  
android:layout_marginEnd="95dp"
```

```
    android:layout_marginBottom="362dp"
    android:onClick="inputNumber"
    android:text="3" />
```

<Button

```
    android:id="@+id/button5"
    android:layout_width="76dp"
    android:layout_height="wrap_content"
    android:layout_alignParentEnd="true"
    android:layout_alignParentBottom="true"
    android:layout_marginEnd="283dp"
    android:layout_marginBottom="294dp"
    android:onClick="inputNumber"
    android:text="4" />
```

<Button

```
    android:id="@+id/button6"
    android:layout_width="76dp"
    android:layout_height="wrap_content"
    android:layout_alignParentEnd="true"
    android:layout_alignParentBottom="true"
    android:layout_marginEnd="187dp"
    android:layout_marginBottom="293dp"
    android:onClick="inputNumber"
    android:text="5" />
```

<Button

```
    android:id="@+id/button7"
    android:layout_width="76dp"
    android:layout_height="wrap_content"
    android:layout_alignParentEnd="true"
    android:layout_alignParentBottom="true"
    android:layout_marginEnd="97dp"
    android:layout_marginBottom="293dp"
    android:onClick="inputNumber"
    android:text="6" />
```

<Button

```
    android:id="@+id/button8"
    android:layout_width="76dp"
    android:layout_height="wrap_content"
    android:layout_alignParentEnd="true"
    android:layout_alignParentBottom="true"
    android:layout_marginEnd="282dp"
    android:layout_marginBottom="222dp"
    android:onClick="inputNumber"
    android:text="7" />
```

<Button

```
    android:id="@+id/button9"
    android:layout_width="76dp"
    android:layout_height="wrap_content"
    android:layout_alignParentEnd="true"
    android:layout_alignParentBottom="true"
    android:layout_marginEnd="189dp"
```

```
    android:layout_marginBottom="219dp"
    android:onClick="inputNumber"
    android:text="8" />
```

<Button

```
    android:id="@+id/button10"
    android:layout_width="76dp"
    android:layout_height="wrap_content"
    android:layout_alignParentEnd="true"
    android:layout_alignParentBottom="true"
    android:layout_marginEnd="95dp"
    android:layout_marginBottom="221dp"
    android:onClick="inputNumber"
    android:text="9" />
```

<Button

```
    android:id="@+id/button11"
    android:layout_width="76dp"
    android:layout_height="wrap_content"
    android:layout_alignParentEnd="true"
    android:layout_alignParentBottom="true"
    android:layout_marginEnd="284dp"
    android:layout_marginBottom="157dp"
    android:onClick="inputNumber"
    android:text="#" />
```

<Button

```
    android:id="@+id/button12"
    android:layout_width="76dp"
    android:layout_height="wrap_content"
    android:layout_alignParentEnd="true"
    android:layout_alignParentBottom="true"
    android:layout_marginEnd="187dp"
    android:layout_marginBottom="158dp"
    android:onClick="inputNumber"
    android:text="0" />
```

<Button

```
    android:id="@+id/button13"
    android:layout_width="76dp"
    android:layout_height="wrap_content"
    android:layout_alignParentEnd="true"
    android:layout_alignParentBottom="true"
    android:layout_marginEnd="94dp"
    android:layout_marginBottom="160dp"
    android:onClick="inputNumber"
    android:text="*" />
```

<Button

```
    android:id="@+id/callBtn"
    android:layout_width="76dp"
    android:layout_height="wrap_content"
    android:layout_alignParentEnd="true"
    android:layout_alignParentBottom="true"
    android:layout_marginEnd="255dp"
```

```

        android:layout_marginBottom="88dp"
        android:text="Call" />

<Button
    android:id="@+id/saveBtn"
    android:layout_width="76dp"
    android:layout_height="wrap_content"
    android:layout_alignParentEnd="true"
    android:layout_alignParentBottom="true"
    android:layout_marginEnd="136dp"
    android:layout_marginBottom="90dp"
    android:text="Save" />
</RelativeLayout>

```

#### JAVA-CODE

```

package com.example.callapplication;
import androidx.appcompat.app.AppCompatActivity;
import android.content.Intent;
import android.net.Uri;
import android.os.Bundle;
import android.provider.ContactsContract;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
public class MainActivity extends AppCompatActivity {
    EditText phoneNumberEditText;
    Button clearBtn,callBtn,saveBtn;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        phoneNumberEditText=findViewById(R.id.phoneNumberEditText);
        callBtn=findViewById(R.id.callBtn);
        saveBtn=findViewById(R.id.saveBtn);
        clearBtn=findViewById(R.id.clearBtn);

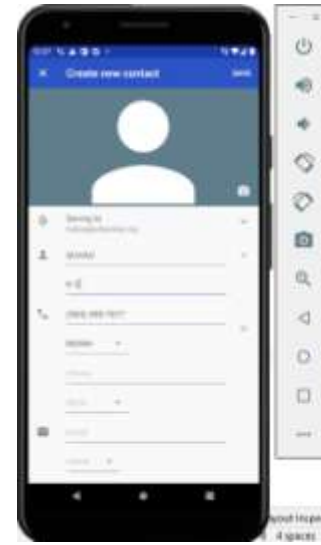
        clearBtn.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v)
            {
                phoneNumberEditText.setText("");
            }
        });
        callBtn.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                String phoneNumber=phoneNumberEditText.getText().toString();
                Intent intent=new Intent(Intent.ACTION_DIAL);
                intent.setData(Uri.parse("tel:"+phoneNumber));
                startActivity(intent);
            }
        });
        saveBtn.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {

```

```

String phoneNumber=phoneNumberEditText.getText().toString();
Intent intent=new Intent(Intent.ACTION_INSERT);
intent.setType(ContactsContract.Contacts.CONTENT_TYPE);
intent.putExtra(ContactsContract.Intents.Insert.PHONE,phoneNumber);
startActivity(intent);
    }
});
}
public void inputNumber(View V){
    Button btn=(Button)V;
    String digit=btn.getText().toString();
    String phoneNumber=phoneNumberEditText.getText().toString();
    phoneNumberEditText.setText(phoneNumber +digit);
}
}

```

**OUTPUT****Design****Call****Save****Contact Saved**