

BMS INSTITUTE OF TECHNOLOGY AND MANAGEMENT
DEPARTMENT OF CSE



MOBILE APPLICATION DEVELOPMENT
(Effective from the academic year 2018 -2019)

SEMESTER – VI

Course Code : 18CSMP68

Laboratory Objectives: This laboratory (18CSMP68) will enable students to -

- Learn and acquire the art of Android Programming.
- ConfigureAndroid studio to run the applications.
- Understand and implement Android's User interface functions.
- Create, modify and query on SQLite database.
- Inspect different methods of sharing data using services.

List of Applications

Part-A

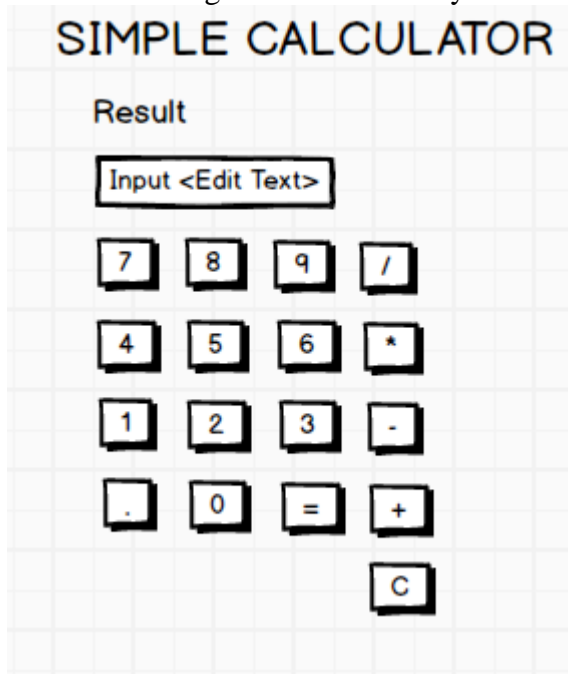
- Visiting Card

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.



- Calculator

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



- LoginSignUp

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.

The image shows two side-by-side UI designs on a light gray grid background. The left design is titled "SIGNUP ACTIVITY" and contains labels "Username:" and "Password:" followed by empty text input fields. Below the fields is a button labeled "SIGN UP". The right design is titled "LOGIN ACTIVITY" and contains similar labels "Username:" and "Password:" with empty text input fields. Below the fields is a button labeled "SIGN IN".

- Wallpaper

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.

The image shows a UI design titled "CHANGING WALLPAPER APPLICATION" on a light gray grid background. It features a single button in the center with the text "CLICK HERE TO CHANGE WALLPAPER".

- Counter

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.

The image shows a UI design titled "COUNTER APPLICATION" on a light gray grid background. It features a label "Counter Value" at the top. Below the label are two buttons: "START" and "STOP", arranged vertically.

- Parsing XML and JSON

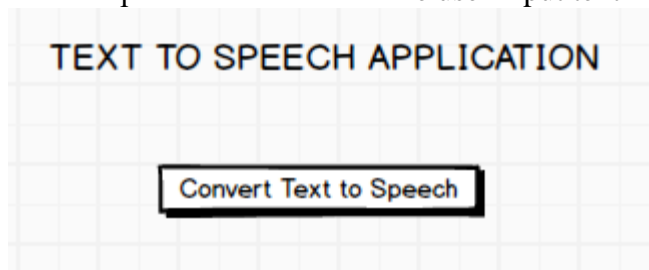
Create two files of XML and JSON type with values for City_Name, Latitude, Longitude, Temperature, and Humidity. Develop an application to create an activity with two buttons to parse

the XML and JSON files which when clicked should display the data in their respective layouts side by side.

PARSING XML AND JSON DATA	
XML DATA	JSON Data
City_Name: Mysore	City_Name: Mysore
Latitude: 12.295	Latitude: 12.295
Longitude: 76.639	Longitude: 76.639
Temperature: 22	Temperature: 22
Humidity: 90%	Humidity: 90%

- Text To Speech

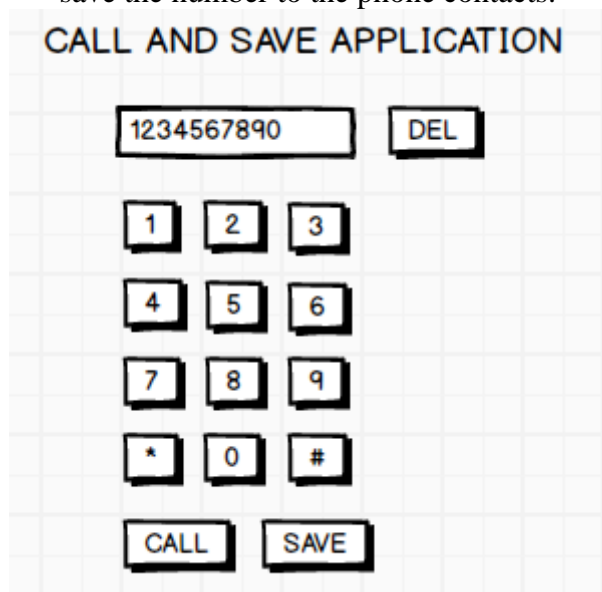
Develop a simple application with one EditText so that the user can write some text in it. Create a button called “Convert Text to Speech” that converts the user input text into voice.



- Phone Dialer

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.



PART - B

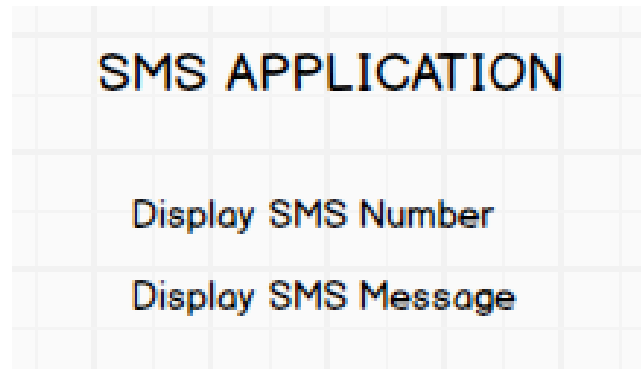
1. Write a program to enter Medicine Name, Date and Time of the Day as input from the user and store it in the SQLite database. Input for Time of the Day should be either Morning or Afternoon or Evening or Night. Trigger an alarm based on the Date and Time of the Day and display the Medicine Name.

The screenshot shows a mobile application interface titled "MEDICINE DATABASE". It features three input fields: "Medicine Name:", "Date:", and "Time of the Day:". Below these fields is a button labeled "Insert". The background has a light gray grid pattern.

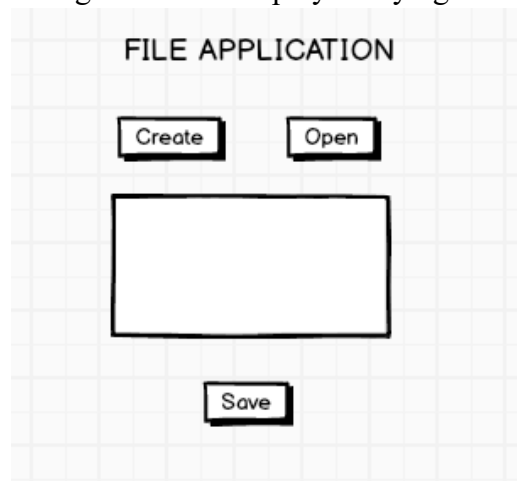
2. Develop a content provider application with an activity called "Meeting Schedule" which takes Date, Time and Meeting Agenda as input from the user and store this information into the SQLite database. Create another application with an activity called "Meeting Info" having DatePicker control, which 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".

The image shows two screenshots of mobile applications. The left screenshot is titled "MEETING SCHEDULE" and has three input fields: "Date:", "Time:", and "Meeting Agenda:". Below these fields is a button labeled "Add Meeting Agenda". The right screenshot is titled "MEETING INFO" and features a text prompt "Pick a date to get meeting info:" followed by a date input field and a calendar icon. Below this is a calendar widget showing the month of July 2018, with the 15th selected. At the bottom of the calendar are "CANCEL" and "OK" buttons. Below the calendar is a "Search" button. Both screens have a light gray grid background.

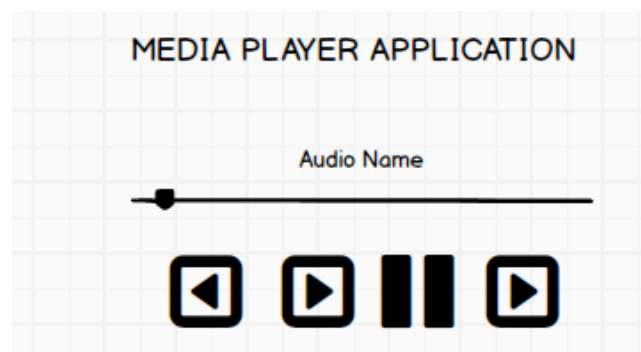
3. Create an application to receive an incoming SMS that is notified to the user. On clicking this SMS notification, the message content and the number should be displayed on the screen. Use appropriate emulator control to send the SMS message to your application.



4. 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 a File”.

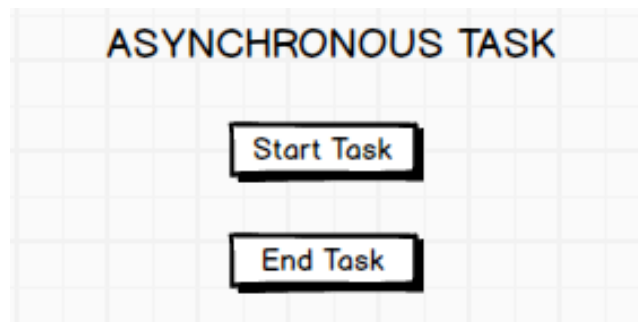


5. Create an application to demonstrate a basic media player that allows the user to Forward, Backward, Play and Pause an audio. Also, make use of the indicator in the seek bar to move the audio forward or backward as required.

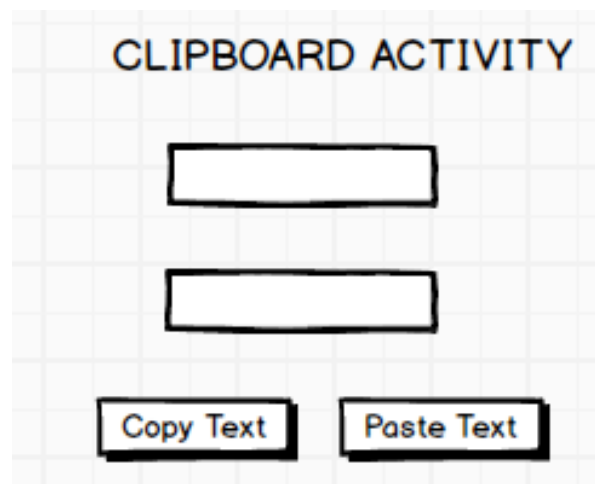


6. Develop an application to demonstrate the use of Asynchronous tasks in android. The asynchronous task should implement the functionality of a simple moving banner. On pressing the Start Task button,

the banner message should scroll from right to left. On pressing the Stop Task button, the banner message should stop. Let the banner message be “Demonstration of Asynchronous Task”.



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 PrincipalAmount, 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.

CAR EMI CALCULATOR

Principal Amount:

Down Payment:

Interest Rate:

Loan Term (in months):

EMI: Result

Calculate Monthly EMI

Guide to Install and Set up Android Studio

Android Studio is the official integrated development environment for Google's Android operating system, built on JetBrains' IntelliJ IDEA software and designed specifically for Android development.

Operating system: Windows, macOS, Linux

Stable release: 3.1.3 (June 2018; 1 month ago)

Preview release: 3.2 Beta 5 (July 30, 2018; 1 day ago)

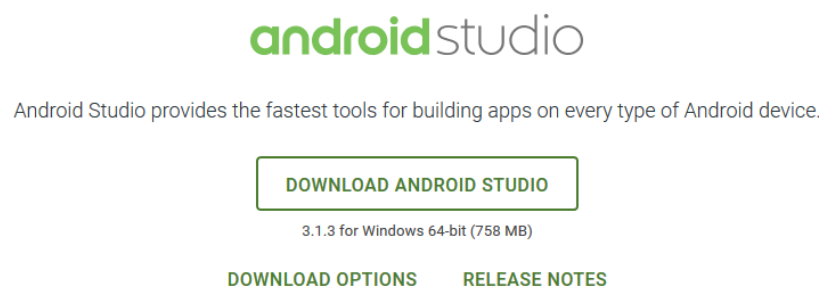
License: Freeware +Source code

Size: 854 MB compressed

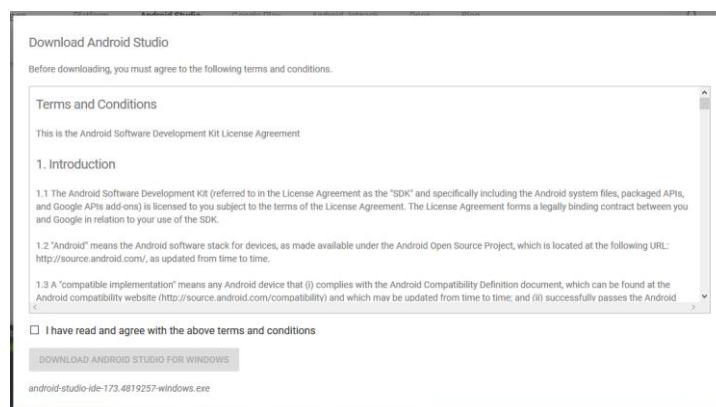
Developed by: Google, JetBrains

Installation guide:

- **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 .



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



Click on 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 dialogue box .



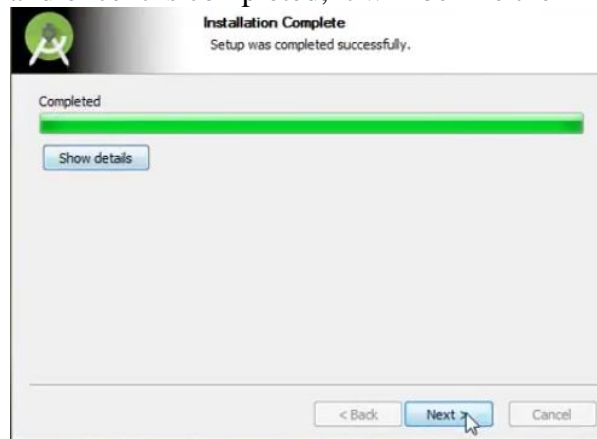
Click on next .

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

Note : The installation path should have the required minimum space.

- **Step – 4 :**

It will start the installation, and once it is completed, it will be like the image shown below .



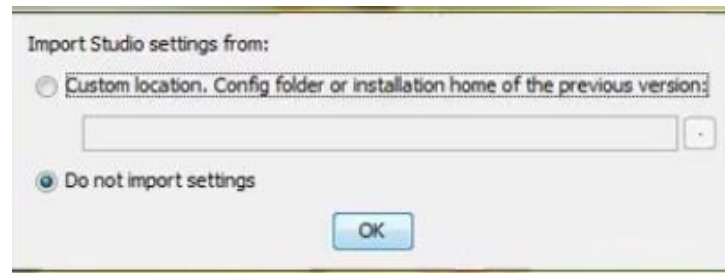
Click on next .



- **Step – 5 :**

Once “Finish” is clicked, it will ask whether the previous settings needs to be imported [if android studio had been installed earlier], or not.

It is better to choose the ‘Don’t import Settings option’ .

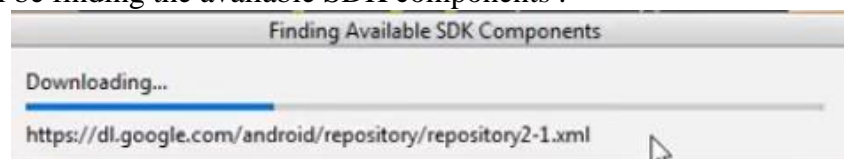


Click the OK button.

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



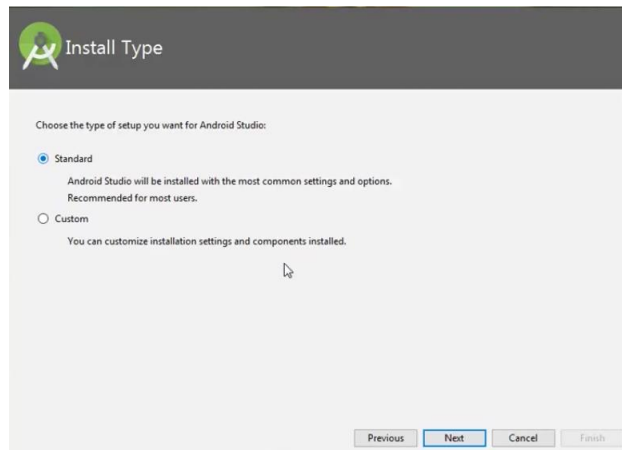
Meanwhile it will be finding the available SDK components .



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



Click on next .

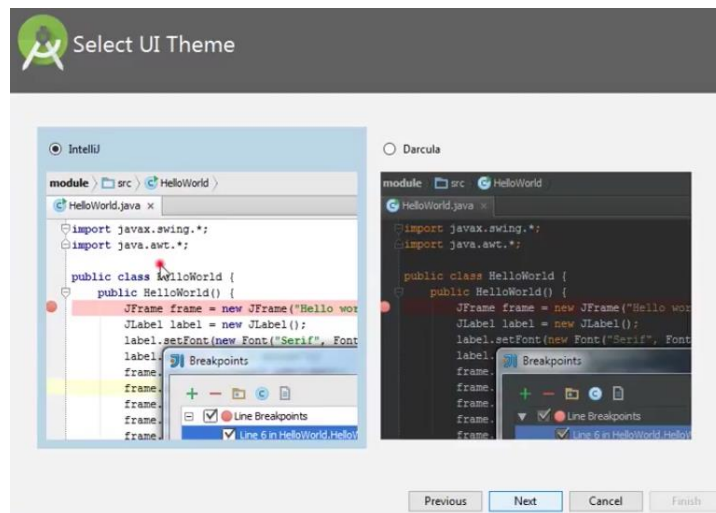


Choose Standard and click on Next.

Now choose the theme, whether Light theme or the Dark one .

The light one is called the IntelliJ theme whereas the dark theme is called Darcula .

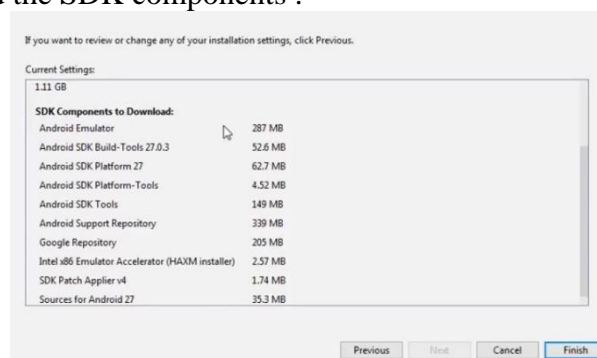
Choose as required.



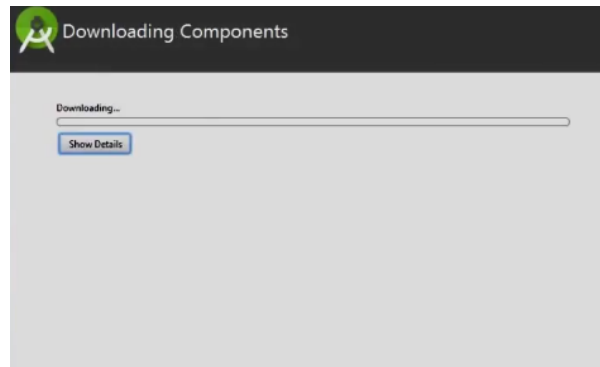
Click on the Next button

• Step – 8 :

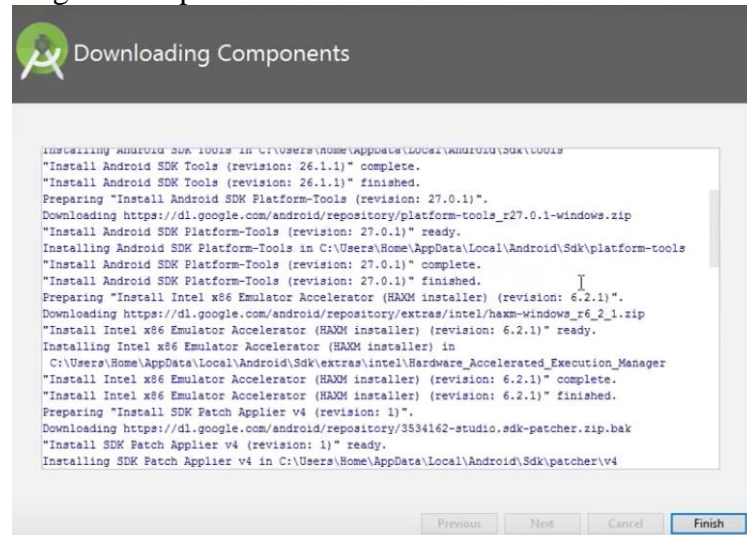
Now it is time to download the SDK components .



Click on Finish .



It has started downloading the components



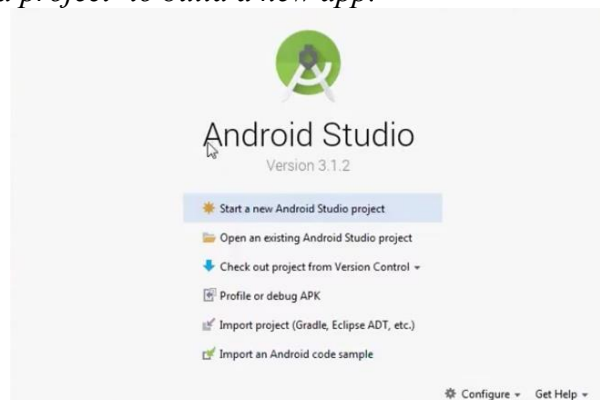
The Android Studio has been successfully configured.

Now it's time to launch and build apps.

Click on the Finish button to launch it.

- **Step – 9 :**

Click on 'Start new android 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.

- 1) Firstly Create an Application by Name “VisitingCardApp”
- 2) Go to xml code of design change the layout to “RelativeLayout”
- 3) Add TextView component change the following properties:
 - Size: 38dp
 - Text: VVCE
 - Align left top
- 4) Add ImageView to design and in type choose “IC_LAUNCHER_FOREGROUND”
 - Download the logo & copy the same in res->drawable folder
 - In xml code of imageview change srcCompat=”@drawable/logo”
 - Align right top
- 5) Add View component & change the following properties:
 - Height: 4dp
 - Background: “#4444” (black color)
- 6) Add TextView component change the following properties:
 - Size: 20dp
 - Text: Nithin Kumar
 - Style: Bold
 - Align center
- 7) Add TextView component change the following properties:
 - Size: 20dp
 - Text: Assistant Professor-CSE
 - Align center
- 8) Add TextView component change the following properties:
 - Size: 20dp
 - Text: Address-Kannada Sahithya Parishath Road, Mysuru-02
 - Align: center
- 9) Add TextView component change the following properties:
 - Size: 20dp
 - Text: Email-nithingowda021@vvce.ac.in
 - Align: center
- 10) Add TextView component change the following properties:
 - Size: 20dp
 - Text: Phone-8050462225

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:id="@+id/textView"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignParentEnd="true"
        android:layout_alignParentRight="true"
        android:layout_alignParentBottom="true"
        android:layout_marginStart="17dp"
        android:layout_marginLeft="17dp"
        android:layout_marginTop="17dp"
        android:layout_marginEnd="244dp"
        android:layout_marginRight="244dp"
        android:layout_marginBottom="486dp"
        android:text="VVCE" android:textSize="38dp" />

    <ImageView android:id="@+id/imageView"
        android:layout_width="231dp"
        android:layout_height="174dp" android:layout_alignParentEnd="true"
        android:layout_alignParentRight="true"
        android:layout_alignParentBottom="true"
        android:layout_marginEnd="-14dp"
        android:layout_marginRight="-14dp"
        android:layout_marginBottom="481dp"
        app:srcCompat="@drawable/logo" />

    <View
        android:id="@+id/view"
        android:layout_width="wrap_content"
        android:layout_height="4dp"
        android:layout_alignParentBottom="true"
        android:background="#4444"
        android:layout_marginBottom="466dp" />
```

```
<TextView android:id="@+id/textView2"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_alignParentEnd="true"
    android:layout_alignParentRight="true"
    android:layout_alignParentBottom="true"
    android:layout_marginEnd="117dp"
    android:layout_marginRight="117dp"
    android:layout_marginBottom="394dp"
    android:text="SWATHI SANDHYA"
    android:textSize="30dp"
    android:textStyle="bold" />
```

```
<TextView android:id="@+id/textView3"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_alignParentEnd="true"
    android:layout_alignParentRight="true"
    android:layout_alignParentBottom="true"
    android:layout_marginEnd="64dp"
    android:layout_marginRight="64dp"
    android:layout_marginBottom="343dp"
    android:text="BMSIT"
    android:textSize="25dp" />
```

```
<TextView android:id="@+id/textView4"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_alignParentEnd="true"
    android:layout_alignParentRight="true"
    android:layout_alignParentBottom="true"
    android:layout_marginEnd="127dp"
    android:layout_marginRight="127dp"
    android:layout_marginBottom="294dp"
    android:text="Ph No: 8050462225"
    android:textSize="20dp" />
```

```
<TextView android:id="@+id/textView5"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_alignParentEnd="true"
    android:layout_alignParentRight="true"
```



```

        android:layout_alignParentBottom="true"
        android:layout_marginEnd="10dp"
        android:layout_marginRight="10dp"
        android:layout_marginBottom="229dp"
        android:text="820987565"
        android:textSize="20dp" />

```

```

<TextView android:id="@+id/textView6"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_alignParentEnd="true"
    android:layout_alignParentRight="true"
    android:layout_alignParentBottom="true"
    android:layout_marginEnd="44dp"
    android:layout_marginRight="44dp"
    android:layout_marginBottom="189dp"
    android:text="#45 Viewpoint, GreenValley"
    android:textSize="20dp" />

```

```

</RelativeLayout>

```

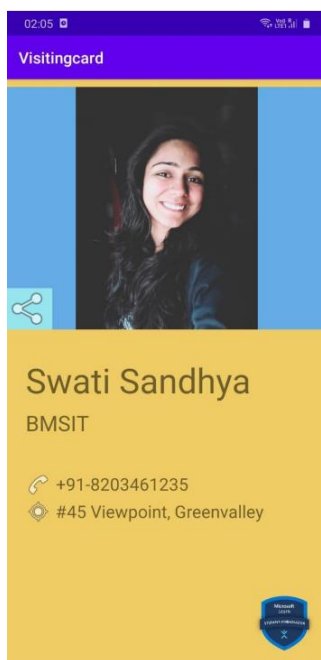
JAVA-CODE

```

import androidx.appcompat.app.AppCompatActivity;import
android.os.Bundle;
public class MainActivity extends AppCompatActivity { 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.

- 1) Firstly Create an Application by Name "SimpleCalci"
- 2) Go to xml code of design change the layout to "RelativeLayout"
- 3) Add TextView component & change the following properties:
 - Size: 38dp
 - Text: Simple Calci
 - Center-Align
- 4) Add PlainText(EditText) component & change the following properties in XML Code:
 - Text: ""
 - Hint: "Enter the first number"
 - id: "@+id/editText1"
- 5) Add PlainText(EditText) component & change the following properties in XML Code:
 - Text: ""
 - Hint: "Enter the second number"
 - id: "@+id/editText2"
- 6) Add TextView component to display result & change the following properties:
 - Size: 40dp
 - Text: "0"
 - Center-Align
 - id: "@+id/textView1"
- 7) Add 4 Buttons & rename the four buttons "Add", "Sub", "Mul" and "div" with following addition:
 - Onclick: "doAdd"(Add Button)
 - Onclick: "doSub"(Sub Button)
 - Onclick: "doMul"(Mul Button)
 - Onclick: "doDiv"(Div Button)

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="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignParentEnd="true"
        android:layout_alignParentBottom="true"
```

```
android:layout_marginEnd="98dp"
android:layout_marginBottom="653dp"
android:text="SIMPLE CALCI"

android:textSize="38dp"
app:layout_constraintBottom_toBottomOf="parent"
app:layout_constraintHorizontal_bias="0.498"
app:layout_constraintLeft_toLeftOf="parent"
app:layout_constraintRight_toRightOf="parent"
app:layout_constraintTop_toTopOf="parent"
app:layout_constraintVertical_bias="0.042" />
```

```
<EditText android:id="@+id/editText1"
```

```
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_alignParentEnd="true"
    android:layout_alignParentBottom="true"
    android:layout_marginEnd="115dp"
    android:layout_marginBottom="547dp"
    android:ems="10"
```

```
    android:hint="Enter the First Number"
```

```
    android:inputType="textPersonName"
```

```
    android:text="" />
```

```
<EditText android:id="@+id/editText2"
```

```
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_alignParentEnd="true"
    android:layout_alignParentBottom="true"
    android:layout_marginEnd="111dp"
    android:layout_marginBottom="455dp"
```

```
    android:ems="10"
```

```
    android:inputType="textPersonName"
```

```
    android:hint="Enter the Second Number" android:text="" />
```

```
<TextView android:id="@+id/textView1"
```

```
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_alignParentEnd="true"
```

```
    android:layout_alignParentBottom="true"
```

```
    android:layout_marginEnd="203dp"
```

```
    android:layout_marginBottom="350dp"
```

```
    android:text="0" android:textSize="40dp" />
```

```
<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="274dp"
    android:layout_marginBottom="237dp"
    android:onClick="doAdd" android:text="+" />
```

```
<Button android:id="@+id/button2"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_alignParentEnd="true"
    android:layout_alignParentBottom="true"
    android:layout_marginEnd="68dp"
    android:layout_marginBottom="233dp"
    android:onClick="doSub" android:text="-" />
```

```
<Button android:id="@+id/button3"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_alignParentEnd="true"
    android:layout_alignParentBottom="true"
    android:layout_marginEnd="277dp"
    android:layout_marginBottom="115dp"
    android:onClick="doMul" android:text="*" />
```

```
<Button android:id="@+id/button4"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_alignParentEnd="true"
    android:layout_alignParentBottom="true"
    android:layout_marginEnd="63dp"
    android:layout_marginBottom="104dp"
    android:onClick="doDiv" android:text="/" />
```

```
</RelativeLayout>
```

JAVA-CODE:

```
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 tv1; @Override

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

        e1 = (EditText)findViewById(R.id.editText1);
        e2 = (EditText)findViewById(R.id.editText2);
        tv1 = (TextView)findViewById(R.id.textView1);
    }

    public void doAdd(View V){
int a1 = Integer.parseInt(e1.getText().toString());
int a2 = Integer.parseInt(e2.getText().toString());
        int result= a1+a2;
        tv1.setText(""+result);
    }

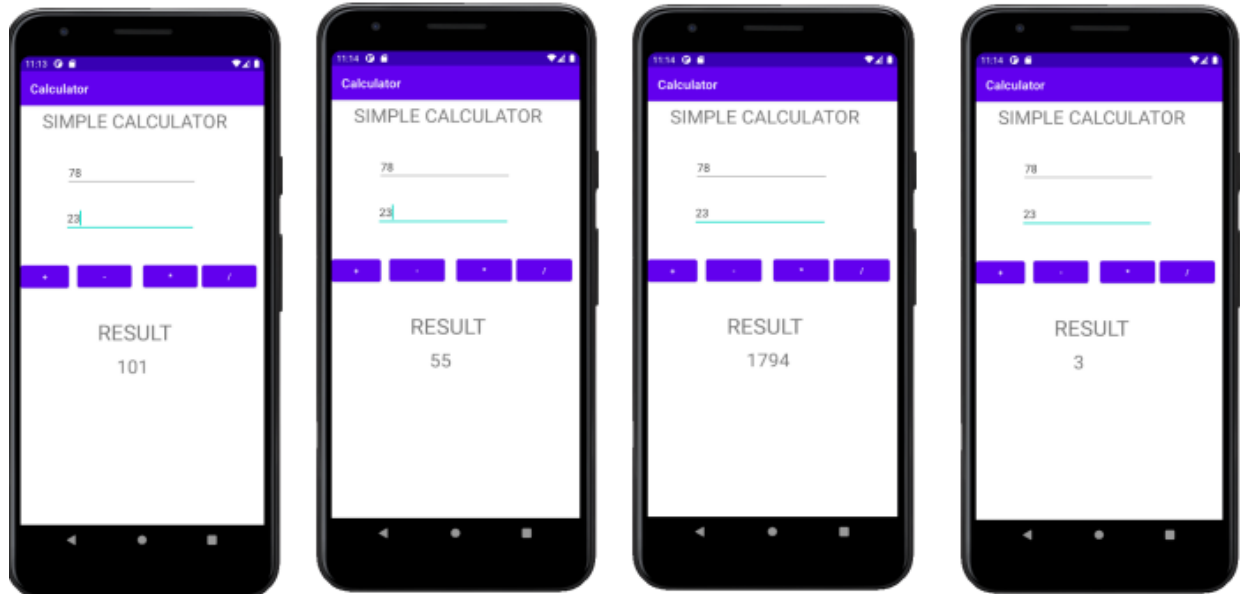
    public void doSub(View V)
    {
int a1 = Integer.parseInt(e1.getText().toString());
int a2 = Integer.parseInt(e2.getText().toString());
        int result= a1-a2;
        tv1.setText(""+result);
    }

    public void doMul(View V){
int a1 = Integer.parseInt(e1.getText().toString());
int a2 = Integer.parseInt(e2.getText().toString());
        int result= a1*a2;
        tv1.setText(""+result);
    }

    public void doDiv(View V){
int a1 = Integer.parseInt(e1.getText().toString());
```

```
int a2 = Integer.parseInt(e2.getText().toString());  
float result= a1/a2;  
tv1.setText(""+result);  
}  
}
```

OUTPUT:



Program-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 SIGNIN 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.

- 1) Firstly Create an Application by Name “SignUpActivity”
- 2) Go to xml code of design change the layout to “RelativeLayout”
- 3) Add TextView component & change the following properties:
 - Size: 38dp
 - Text: “Sign Up”
 - Center-Align
- 4) Add Email (EditText) component & change the following properties in XML Code:
 - Hint: “Email ID”
 - id: “@+id/emailEditText”
- 5) Add Password (EditText) component & change the following properties in XML Code:
 - Hint: “Password”
 - id: “@+id/passwordEditText”
- 6) Add Button component & change the following properties in XML
 - Id: “@+id/signBtn”
 - Text: “Sign Up”

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="160dp"
        android:layout_height="42dp"
        android:layout_alignParentEnd="true"
        android:layout_alignParentBottom="true"
        android:layout_marginEnd="112dp"
```

```
android:layout_marginBottom="573dp"
android:text="Sign Up" android:textSize="28dp"
app:layout_constraintBottom_toBottomOf="parent"
app:layout_constraintLeft_toLeftOf="parent"
app:layout_constraintRight_toRightOf="parent"
app:layout_constraintTop_toTopOf="parent" />
```

```
<EditText android:id="@+id/emailEditText"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_alignParentEnd="true"
    android:layout_alignParentBottom="true"
    android:layout_marginEnd="29dp"
    android:layout_marginBottom="431dp"
    android:ems="10"
    android:hint="Email ID"
    android:inputType="textEmailAddress"
    android:textSize="28dp" />
```

```
<EditText android:id="@+id/passwordEditText"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_alignParentEnd="true"
    android:layout_alignParentBottom="true"
    android:layout_marginEnd="34dp"
    android:layout_marginBottom="345dp"
    android:ems="10" android:hint="Password"
    android:inputType="textPassword"
    android:textSize="28dp" />
```

```
<Button android:id="@+id/signUpBtn"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_alignParentEnd="true"
    android:layout_alignParentBottom="true"
    android:layout_marginEnd="106dp"
    android:layout_marginBottom="226dp"
    android:text="Sign Up" android:textSize="28dp" />
```

```
</RelativeLayout>
```


JAVA-CODE

```
package com.example.loginapplication;

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 emailEditText, passwordEditText;

    Button signUpBtn;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        emailEditText = findViewById(R.id.emailEditText);
        passwordEditText = findViewById(R.id.passwordEditText);
        signUpBtn = findViewById(R.id.signUpBtn);

        signUpBtn.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                String email = emailEditText.getText().toString();
                String password = passwordEditText.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].*$");
    Pattern specialCharacter = Pattern.compile("^[^a-zA-Z0-9].*$");
```

```

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 (!specialCharacter.matcher(password).matches()) {
        return false;
    }
    return true;
}
}

```

- 7) Right click on Java folder-> new-> activity->empty activity-> name it as "LoginActivity"
- 8) Go to xml code of design change the layout to "RelativeLayout"
- 9) Add TextView component & change the following properties:
 - Size: 38dp
 - Text: "Login"
 - Center-Align
- 10) Add Email (EditText) component & change the following properties in XML Code:
 - Hint: "Email ID"
 - id: "@+id/emailEditText"
- 11) Add Password (EditText) component & change the following properties in XML Code:
 - Hint: "Password"
 - id: "@+id/passwordEditText"
- 12) Add Button component & change the following properties in XML
 - Id: "@+id/loginBtn"
 - Text: "Login"

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=".LoginActivity">
```

```
<TextView android:id="@+id/textView"  
    android:layout_width="210dp"  
    android:layout_height="54dp"  
    android:layout_alignParentEnd="true"  
    android:layout_alignParentBottom="true"  
    android:layout_marginEnd="120dp"  
    android:layout_marginBottom="576dp"  
    android:text="Login Activity"  
    android:textSize="28dp" />
```

```
<EditText android:id="@+id/emailEditText"  
    android:layout_width="222dp"  
    android:layout_height="80dp"  
    android:layout_alignParentEnd="true"  
    android:layout_alignParentBottom="true"  
    android:layout_marginEnd="108dp"  
    android:layout_marginBottom="424dp"  
    android:ems="10"  
    android:hint="Email ID"  
    android:inputType="textEmailAddress"  
    android:textSize="28dp" />
```

```
<EditText android:id="@+id/passwordEditText"  
    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="299dp"  
    android:ems="10" android:hint="Password"  
    android:inputType="textPassword"  
    android:textSize="28dp" />
```

```
<Button android:id="@+id/loginBtn"  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:layout_alignParentEnd="true"  
    android:layout_alignParentBottom="true"  
    android:layout_marginEnd="173dp"  
    android:layout_marginBottom="189dp"  
    android:text="login" android:textSize="26dp" />
```

</RelativeLayout>

JAVA-CODE

```
package com.example.loginapplication;

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 emailEditText, passwordEditText;
    Button loginBtn;
    int counter=2;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_login);
        emailEditText=findViewById(R.id.emailEditText);
        passwordEditText=findViewById(R.id.passwordEditText);
        loginBtn=findViewById(R.id.loginBtn);
        String registeredEmail=getIntent().getStringExtra("email");
        String registeredPassword=getIntent().getStringExtra("password");
        loginBtn.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                String email=emailEditText.getText().toString();
                String password=passwordEditText.getText().toString();
                if(registeredEmail.equals(email)&& registeredPassword.equals(password))
                {
                    Intent intent=new Intent (LoginActivity.this,LoginSuccessActivity.class);
                    startActivity(intent);
                }
                else{
                    Toast.makeText(LoginActivity.this,"Invalid
Credentials",Toast.LENGTH_LONG).show();
                }
                counter--;
                if (counter==0)
                {
                    Toast.makeText(getApplicationContext(),"FAILED LOGIN
ATTEMPTS",Toast.LENGTH_LONG).show();
                    loginBtn.setEnabled(false);
                }
            }
        });
    }
}
```

```

    }
    });
}
}

```

13) Right click on Java folder-> new-> activity->empty activity-> name it as

“LoginSuccessful”

14) Go to xml code of design change the layout to “RelativeLayout”

15) Add TextView component & change the following properties:

- Size: 38dp
- Text: “Login Successful”
- Center-Align

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=".LoginSuccessful">

    <TextView android:id="@+id/textView2"
        android:layout_width="297dp"
        android:layout_height="190dp"
        android:layout_alignParentEnd="true"
        android:layout_alignParentBottom="true"
        android:layout_marginEnd="42dp"
        android:layout_marginBottom="400dp"
        android:text="Login Successful"
        android:textSize="38dp" />
</RelativeLayout>

```

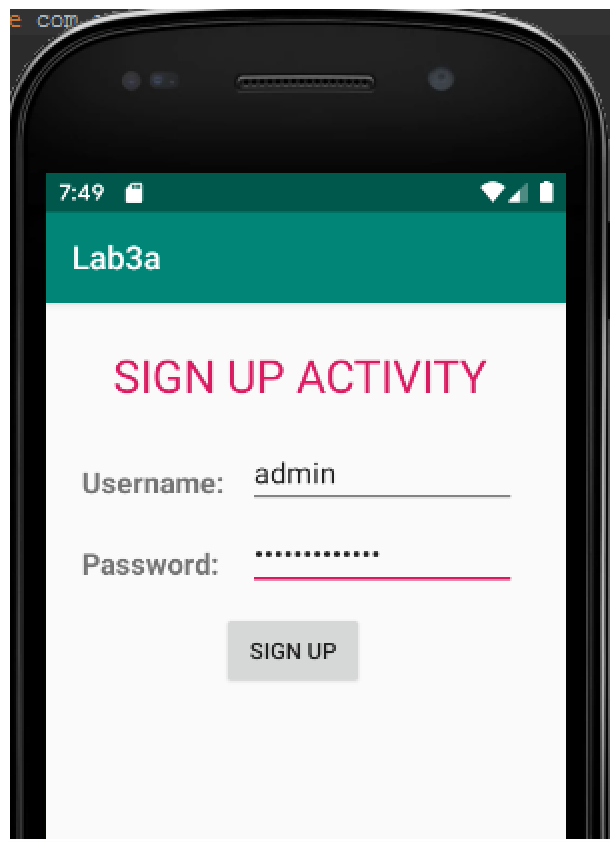
JAVA-CODE

```

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

```

OUT-PUT



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.

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:id="@+id/textView"
        android:layout_width="210dp"
        android:layout_height="54dp"
        android:layout_alignParentEnd="true"
        android:layout_alignParentBottom="true"
        android:layout_marginEnd="120dp"
        android:layout_marginBottom="576dp"
        android:text="Wall Paper Change Application"
        android:textSize="28dp" />

    <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="173dp"
        android:layout_marginBottom="189dp"
        android:text="Click Here To Change Wall Paper"
        android:textSize="26dp" />

</RelativeLayout>
```

JAVA-CODE

```
import androidx.appcompat.app.AppCompatActivity;
import android.app.WallpaperManager;
import android.graphics.Bitmap;
import android.graphics.BitmapFactory;
import android.graphics.drawable.AnimationDrawable;
import android.graphics.drawable.BitmapDrawable;
import android.graphics.drawable.Drawable;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.Toast;
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 view) {
                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;
                }
            }
        });
    }
}
```

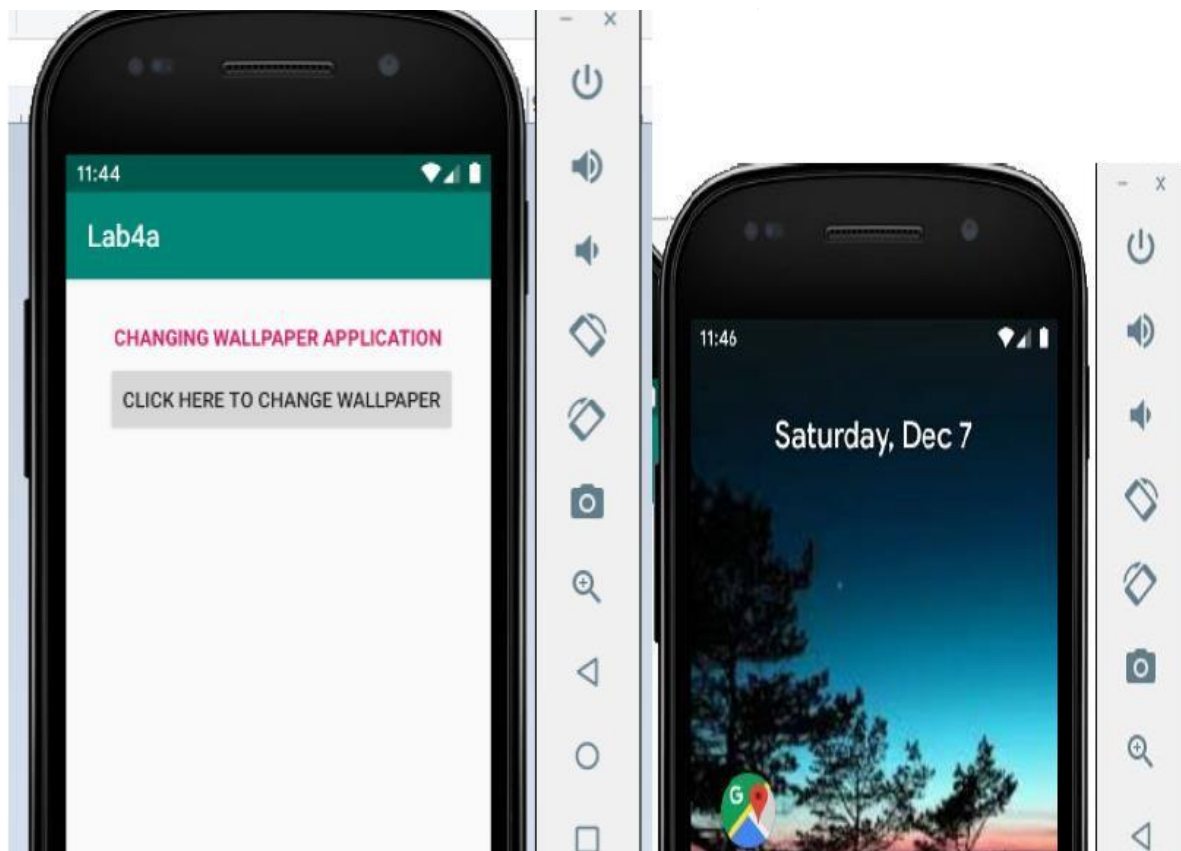


```

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.setBitmap(wallpaper);
} catch (IOException e) {
e.printStackTrace();
}
},0,30000); } }

```

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 TextView 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”
- 5) Add Button components & change the following properties:
 - Size: 38dp
 - Text: Start
 - id: “@+id/btn_start”
- 6) Add Button components & change the following properties:
 - Size: 38dp
 - Text: Stop
 - id: “@+id/btn_stop”

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="378dp"
        android:layout_height="68dp"
        android:layout_alignParentEnd="true"
        android:layout_alignParentBottom="true"
        android:layout_marginEnd="18dp"
        android:layout_marginBottom="602dp"
        android:text="Counter Application"
        android:textSize="38dp"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintLeft_toLeftOf="parent"
        app:layout_constraintRight_toRightOf="parent"
        app:layout_constraintTop_toTopOf="parent" />

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

```
android:layout_width="121dp"
android:layout_height="32dp"
android:layout_alignParentEnd="true"
android:layout_alignParentBottom="true"
android:layout_marginEnd="145dp"
android:layout_marginBottom="478dp"
android:text="Counter Value" />
```

```
<Button android:id="@+id/btn_start"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_alignParentEnd="true"
    android:layout_alignParentBottom="true"
    android:layout_marginEnd="297dp"
    android:layout_marginBottom="295dp"
    android:text="Start" />
```

```
<Button android:id="@+id/btn_stop"
    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="292dp"
    android:text="Stop" />
```

```
</RelativeLayout>
```

JAVA-CODE

```
package com.example.counterapplication;

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 btnstart, btnstop;
    TextView txtcounter;int i=1;
    Handler customHandler=new Handler();

    @Override
```

```

protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);
    btnstart=findViewById(R.id.btn_start);
    btnstop=findViewById(R.id.btn_stop);
    txtcounter=findViewById(R.id.textView);
    btnstart.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            customHandler.postDelayed(updateTimerThread,0);
        }
    });
    btnstop.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            customHandler.removeCallbacks(updateTimerThread);
        }
    });
}
private final Runnable updateTimerThread=new Runnable() {
    @Override
    public void run() { txtcounter.setText(""+i);
        customHandler.postDelayed(this,1000);i++;
    }
};
}

```

Output:

OUTPUT



Program-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:
 - 1) Size: 38dp
 - 2) Text: XML and JSON Parser
 - 3) Center-Align
- 4) Add Two Buttons to Design & change the name “ParseXml” & “ParseJson” with following onclick functions:
 - ParseXml-Button: parsexml
 - ParseJson-Button: parsejson
- 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->Assets 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:

```
<?xml version="1.0"?>
<records>
  <place>
    <name>Mysore</name>
    <lat>12.295</lat>
    <long>76.639</long>
    <temperature>22</temperature>
    <humidity>90 %</humidity>
  </place>
  <place>
    <name>Bangalore</name>
    <lat>12.97165</lat>
```

```
<long>77.5946</long>
<temperature>25</temperature>
<humidity>74 %</humidity>
</place>
</records>
```

city.json:

```
[
{
  "name": "HASSAN",
  "lat": "12.295",
  "long": "76.639",
  "temperature": "22",
  "humidity": "92 %"
},
{
  "name": "MANDYA",
  "lat": "12.97165",
  "long": "77.5946",
  "temperature": "25",
  "humidity": "74 %"
}
]
```

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:id="@+id/textView"

    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_alignParentEnd="true"
    android:layout_alignParentBottom="true"
    android:layout_marginEnd="10dp"
    android:layout_marginBottom="634dp"
    android:text="Parsing XML and JSON"
    android:textSize="36sp" />
```

```
<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="153dp"
    android:layout_marginBottom="484dp"
    android:onClick="parsexml"
    android:text="ParseXML" />
```

```
<Button

    android:id="@+id/button2"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_alignParentEnd="true"
    android:layout_alignParentBottom="true"
    android:layout_marginEnd="151dp"
    android:layout_marginBottom="364dp"
    android:onClick="parsejson"
    android:text="ParseJSON" />
```

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

    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_alignParentEnd="true"
    android:layout_alignParentBottom="true"
    android:layout_marginEnd="3dp"
    android:layout_marginBottom="68dp"
    android:text=""

    android:textAlignment="center" />
```

```
</RelativeLayout>
```

JAVA-CODE:

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

import android.os.Bundle;

import android.text.style.TabStopSpan;

import android.view.View;

import android.widget.TextView;

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;

    @Override

    protected void onCreate(Bundle savedInstanceState) {

        super.onCreate(savedInstanceState);

        setContentView(R.layout.activity_main);

        display = (TextView)findViewById(R.id.display);

    }

    public void parsexml(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();
}
}

public void parsejson(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 in reading",Toast.LENGTH_LONG).show();

}

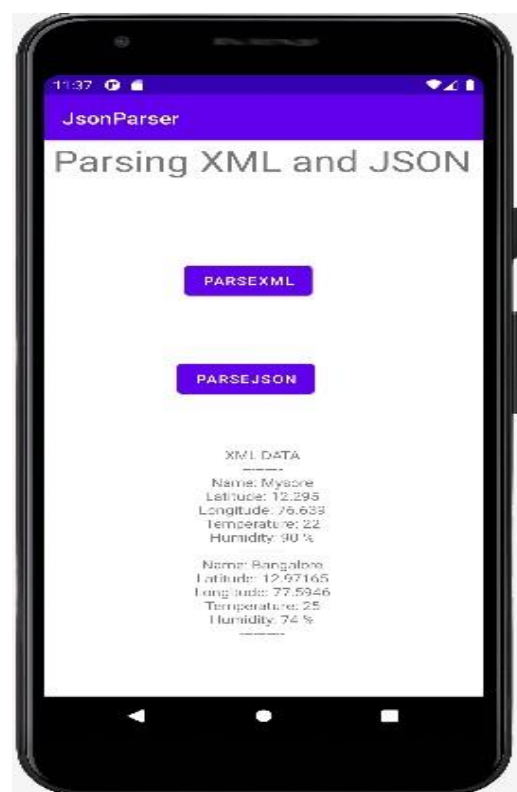
}

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) Firstly 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:
- 4) Size: 38dp
- 5) Text: Text2Speech App
- 6) Center-Align
- 7) Add PlainText(EditText) component & change the following properties in XML Code:
 - Text: “”
 - Hint: “Enter the text to be converted”
 - id: “@+id/editText”
- 8) Add Button component & change the following properties in XML Code:
 - Name: Convert
 - onClick: convert

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:id="@+id/textView"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignParentEnd="true"
        android:layout_alignParentRight="true"
        android:layout_alignParentBottom="true"
        android:layout_marginEnd="59dp"
        android:layout_marginRight="59dp"
        android:layout_marginBottom="649dp"
        android:text="Text2SpeechApp"
        android:textSize="40dp" />

    <EditText android:id="@+id/editText"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignParentEnd="true"
```

```

        android:layout_alignParentRight="true"
        android:layout_alignParentBottom="true"
        android:layout_marginEnd="101dp"
        android:layout_marginRight="101dp"
        android:layout_marginBottom="514dp"
        android:ems="10"
        android:hint="Enter the text to be converted"
        android:inputType="textPersonName"
        android:text="" />

<Button android:id="@+id/button"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignParentEnd="true"
        android:layout_alignParentRight="true"
        android:layout_alignParentBottom="true"
        android:layout_marginEnd="162dp"
        android:onClick="convert"
        android:layout_marginRight="162dp"
        android:layout_marginBottom="329dp"
        android:text="Convert" />
</RelativeLayout>

```

JAVA-CODE:

```

import androidx.appcompat.app.AppCompatActivity;
import android.os.Bundle;
import android.speech.tts.TextToSpeech;
import android.view.View;
import android.widget.EditText;
import android.widget.Toast;

import java.util.Locale;

public class MainActivity extends AppCompatActivity {
    TextToSpeech t1;
    EditText e1;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        e1 = findViewById(R.id.editText);
        t1 = new TextToSpeech(getApplicationContext(), new
        TextToSpeech.OnInitListener() {
            @Override
            public void onInit(int status) {
                if (status!=TextToSpeech.ERROR){
                    t1.setLanguage(Locale.UK);
                }
            }
        });
    }
}

```

```
        }  
    });  
}  
public void convert(View view){  
    String tospeak = e1.getText().toString();  
    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 EditText component & change the following properties in XML Code:
 - id: "@+id/phoneNumberEditText"
- 5) Add PlainText(EditText) component & change the following properties in XML Code:
 - Text: ""
 - Hint: "Copied Text"
 - id: "@+id/editText2"
- 6) 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"
- 7) 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"
    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:textSize="36dp"
        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="311dp"
    android:onClick="inputNumber"
    android:layout_marginBottom="341dp"
    android:text="1" />
```

```
<Button android:id="@+id/button3"
    android:layout_width="76dp"
    android:layout_height="wrap_content"
    android:layout_alignParentEnd="true"
    android:onClick="inputNumber"
    android:layout_alignParentBottom="true"
    android:layout_marginEnd="170dp"
    android:layout_marginBottom="341dp"
    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="32dp"
android:onClick="inputNumber"
android:layout_marginBottom="343dp"
android:text="3" />
```

```
<Button android:id="@+id/button5"
    android:layout_width="76dp"
    android:layout_height="wrap_content"
    android:layout_alignParentEnd="true"
    android:onClick="inputNumber"
    android:layout_alignParentBottom="true"
    android:layout_marginEnd="311dp"
    android:layout_marginBottom="241dp"
    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="175dp"
    android:onClick="inputNumber"
    android:layout_marginBottom="239dp"
    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="32dp"
    android:onClick="inputNumber"
    android:layout_marginBottom="239dp"
    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="313dp"
    android:onClick="inputNumber"
    android:layout_marginBottom="142dp"
    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="171dp"
    android:onClick="inputNumber"
    android:layout_marginBottom="147dp"
    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="34dp"
    android:onClick="inputNumber"
    android:layout_marginBottom="152dp"
    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="309dp"
    android:onClick="inputNumber"

    android:layout_marginBottom="80dp"
    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="169dp"
    android:onClick="inputNumber"
    android:layout_marginBottom="78dp"
    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="34dp"
        android:onClick="inputNumber"
        android:layout_marginBottom="88dp"
        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="284dp"
        android:layout_marginBottom="17dp"
        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="60dp"
        android:layout_marginBottom="17dp"
        android:text="Save" />
</RelativeLayout>

```

JAVA-CODE

```

package com.example.callingapplication;
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:

