S. S. Education Trust ‘s CET Code: E-175 (UG)/T-942 (PG)

S. G. BALEKUNDRI INSTITUTE OF TECHNOLOGY

An ISO 21001:2018 certified institution

Shivabasavanagar, Belagavi- 590 010, Karnataka- India

Department of Computer Science and Engineering

Accredited by NBA

Mobile Application Development

(18CSMP68)

Prepared By

Asst. Prof. Shaheen Mujawar

Assistant professor

Department of Computer Science and Engineering

SGBIT Belagavi

S. S. Education Trust's CET Code: E-175 (UG)/T-942 (PG)

S. G. BALEKUNDRI INSTITUTE OF TECHNOLOGY

Shivabasavanagar, Belagavi- 590 010, Karnataka- India

Office: 0831-2407172, 2554559 Fax: 0831-2407152 Website: www.sgbit.edu.in

● Approved by AICTE New Delhi ● Recognised by Govt. of Karnataka ●Affiliated to V T U, Belagavi

An ISO 21001:2018 certified institution

Department of Computer Science and Engineering

Accrediated by NBA

Email: hod-cs@sgbit.edu.in, Dept. Extn.: 518

Institute Vision

To impart Quality Education with Human values and emerge as one of the Nation's leading Institutions in the field of Technical Education and Research.

Institute Mission

Strive to encourage ideas, talents and value systems.
Guide students to be successful in their endeavour with moral and ethical values.
Build relation with Industries and National Laboratories to support in the field of Engineering and Technology.
Inculcate a thirst for knowledge in students and help them to achieve Academic Excellence and Placement.
Train and develop the faculty to achieve Professional, Organizational objectives, and excel in Research and Development.

Department Vision

To strive for excellence in imparting knowledge of Computer Science and Engineering to produce **IT professionals** committed to **human values.**

Department Mission

M1: Impart quality education in cutting edge technologies to achieve excellence in computer science and engineering to solve real-world problems.
M2: Imbibe **human values and ethical responsibilities** in professional Endeavors.

PEO's

PEO1: To prepare graduates to be successful in IT enabled professional career, higher studies and research by providing contextually relevant academic environment.
PEO2: To prepare graduates to be independent and adapt to the changing technologies by inculcating life-long learning ability, leadership qualities, and entrepreneurial skills.
PEO3: To prepare graduates to be committed citizens with social, ethical, and professional concerns.

PSO's

PSO1: To analyze and resolve the engineering problems related to Artificial Intelligence, Big Data analytics and Mobile Application Development for efficient design of computer based system of varying complexity.
PSO2: To design and arrive with optimal solution for a composite computer science and engineering problems with synthesized optimal hardware and software system.
PSO3: Apply reasoning informed by the contextual knowledge of computer science and engineering to resolve societal, health, safety and environmental problem.

A. LABORATORY OVERVIEW

Degree:	BE	Programme:	CS
Semester:	6	Academic Year:	2021-22
Laboratory Title:	**Mobile Application Development**	Laboratory Code:	18CSMP68
Number of Contact Hours/Week	0:0:2	Exam Hours:	3 Hrs
Total Contact Hours:	3 Hours/Week	Exam Marks:	60
Credits:	2	IA Marks:	40
Lab Manual Author:	Asst. Prof. Shaheen Mujawar	Sign:	Date :
Verified By:	Dr. B.S. Halakarnimath HOD,CSE	Sign:	Date :

B. DESCRIPTION

1. PREREQUISITES:

- Creative thinking, sound mathematical insight and programming skills.
- Computer Programming Laboratory
- Java Programming

2. BASE COURSE:

- Computer Programming Laboratory
- Java Programming

3. RELEVANCE OF THE COURSE:

- Project work ,Internship

4. COURSE OUTCOMES

At the end of the course, the students will be able to

C308.1: Create, test and debug Android application by setting up Android development environment.
C308.2: Implement adaptive, responsive user interfaces that work across a wide range of devices.
C308.3: Infer long running tasks and background work in Android applications.
C308.4: Demonstrate methods in storing, sharing and retrieving data in Android applications.
C308.5: Infer the role of permissions and security for Android applications.
C308.6: Know the methodologies and professional way of documentation, communication, team work and Project Management solve societal issues.

5. Mapping of CO-POs and PSOs:

Course Outcomes(COs) /POs & PSOs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
C308.1	2	2	2		2							2	2	2	
C308.2	2	2	2		2										
C308.3	2	2	2		2										
C308.4	2	2	2		2										
C308.5	2	2	2		2										
C308.6	2							2	2	2	2	2			2

MOBILE APPLICATION DEVELOPMENT

(Effective from the academic year 2018 -2019)

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

Laboratory Objectives: Thislaboratory (18CSMP68) will enable students to

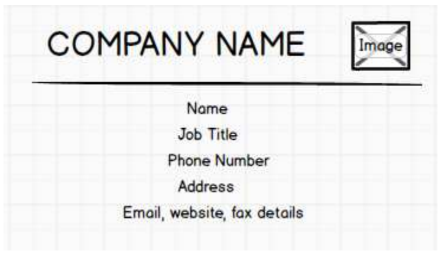
- Learn and acquire the art of Android Programming.
- ConfigureAndroid studio to run the applications.
- Understand and implement Android's User interface functions.
- Create, modify and query on SQlite database.
- Inspect different methods of sharing data using services.

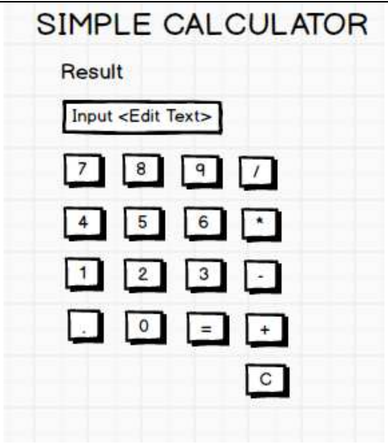
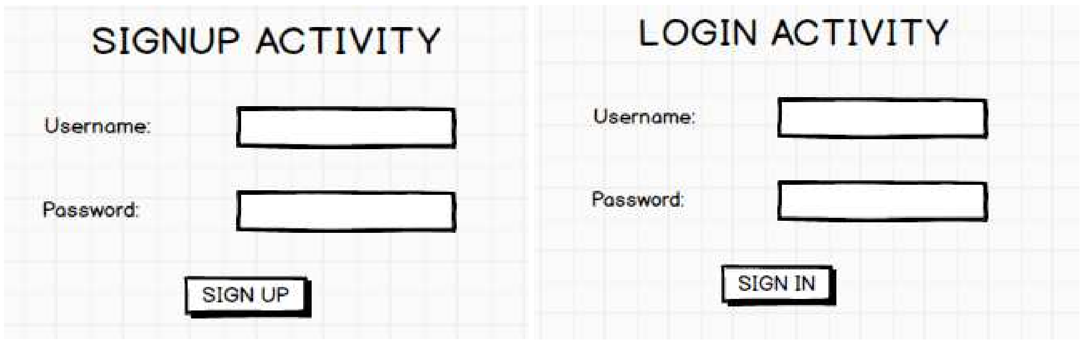
Descriptions (if any):

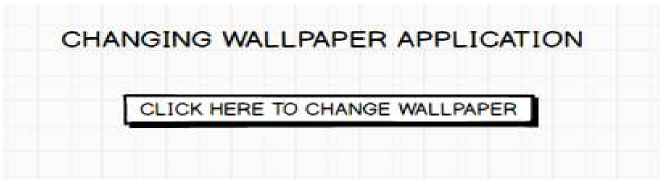
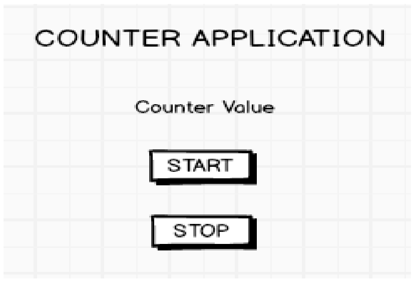
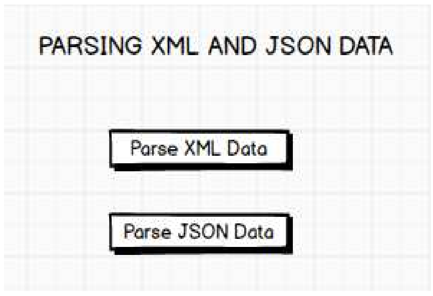
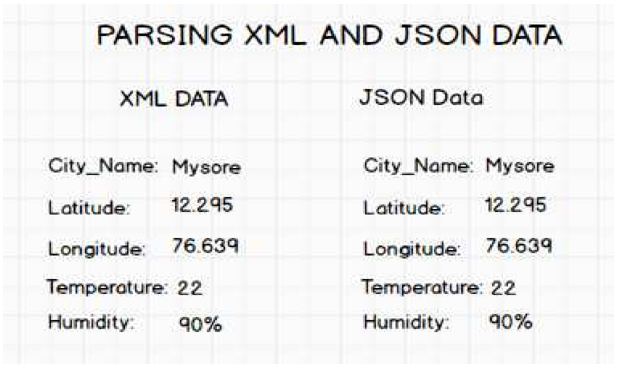
Installation procedure of the Android Studio/Java software must be demonstrated, carried out in groups.
Students should use the latest version of Android Studio/Java to execute these programs.
All of these diagrams are for representational purpose only.Students are expected to improvise on it.

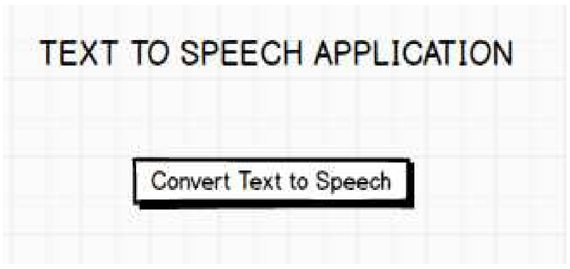
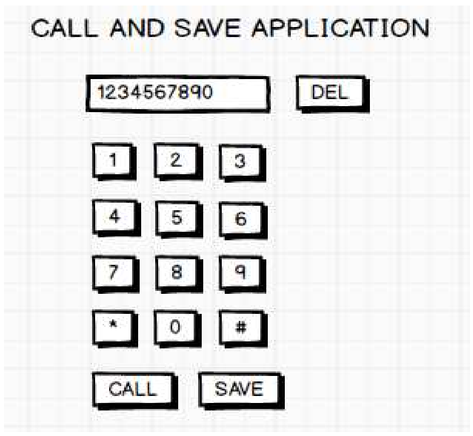
Programs List:

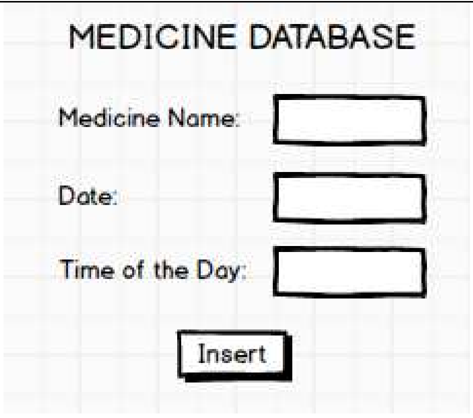
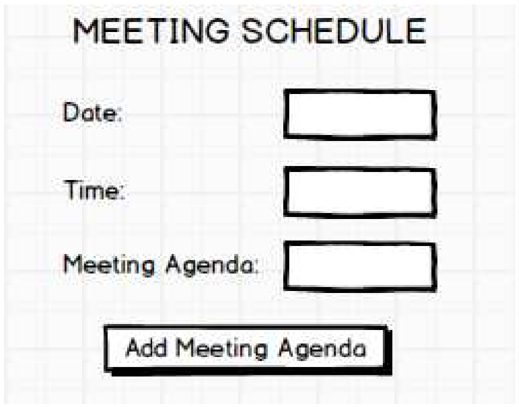
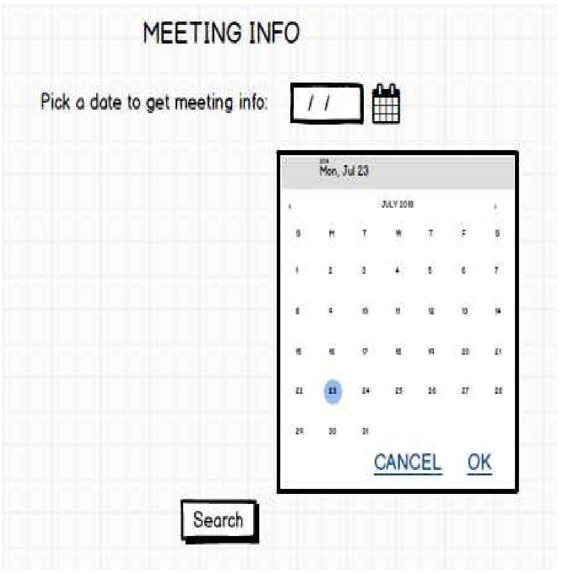
PART – A

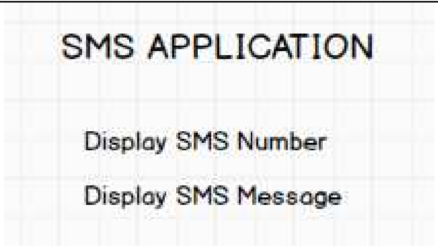
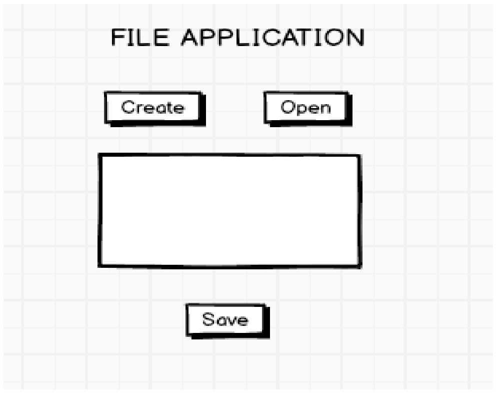
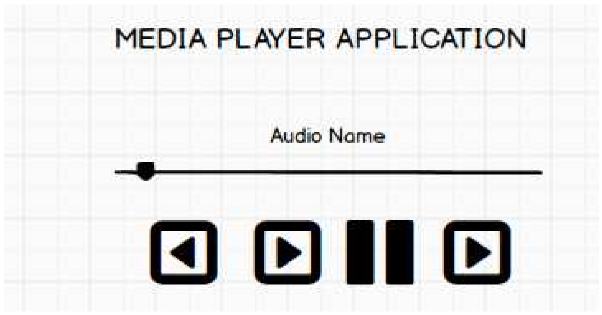
1	Create an application to design aVisiting Card. The Visiting card should havea companylogoatthe top right corner. The company name should be displayed in Capital letters, aligned to the center. Information like the name of the employee, job title, phone number, address, email, fax and the website address isto be displayed. Insert a horizontal line between the job title and the phone number.  COMPANY NAME Image Name Job Title Phone Number Address Email, website, fax details
2	Develop an Android application usingcontrols like Button, TextView, EditText for designing a calculatorhaving basic functionality like Addition, Subtraction, Multiplication,andDivision.

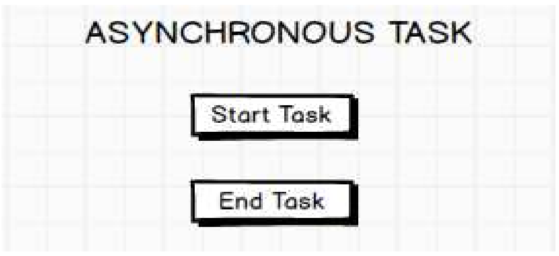
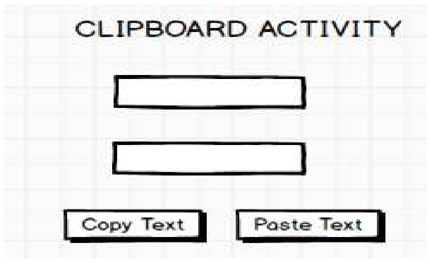
	 SIMPLE CALCULATOR Result Input <Edit Text> 7 8 9 / 4 5 6 * 1 2 3 - . 0 = + C
3	Create a SIGN Up activity with Username and Password. Validation of password should happen based on the following rules: • Password should contain uppercase and lowercase letters. • Password should contain letters and numbers. • 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 whichdisplays a message saying "Successful Login" or else display a toast message saying "Login Failed".The user is given only two attempts and after thatdisplay 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 Username: Password: SIGN UP LOGIN ACTIVITY Username: Password: SIGN IN

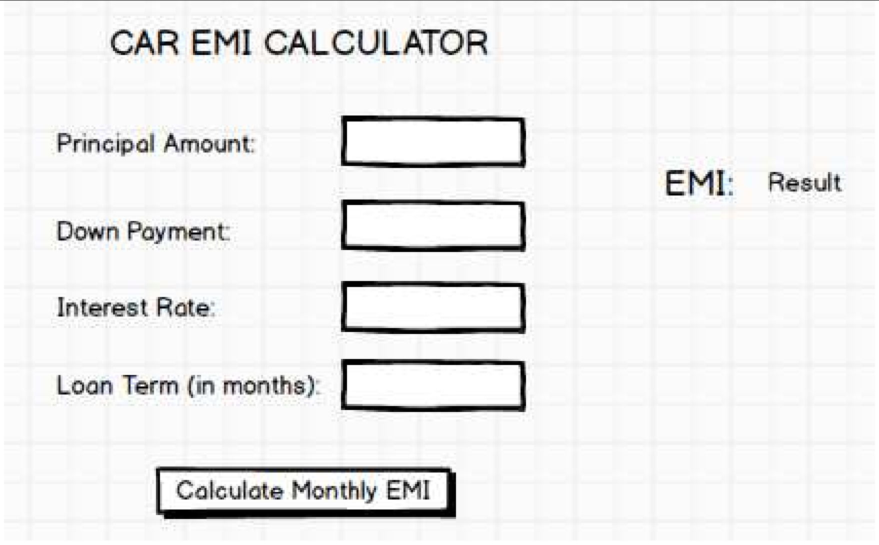
4	Develop an application to set an image as wallpaper. On click of a button, the wallpaper image should start to change randomly every 30 seconds.  CHANGING WALLPAPER APPLICATION CLICK HERE TO CHANGE WALLPAPER
5	Write a program to create an activity with two buttons START and STOP. On pressingoftheSTART 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 TextViewcontrol. 
6	Create two files of XML and JSON type with values for City_Name, Latitude, Longitude, Temperature,andHumidity. 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: City_Name: Mysore Latitude: 12.295 Longitude: 76.639 Temperature: 22 Humidity: 90% JSON Data: City_Name: Mysore Latitude: 12.295 Longitude: 76.639 Temperature: 22 Humidity: 90%

7	Develop a simple application withoneEditTextso 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 withCALLand 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.  CALL AND SAVE APPLICATION 1234567890 DEL 1 2 3 4 5 6 7 8 9 * 0 # CALL SAVE
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 Eveningor Night. Trigger an alarm based on the Date and Time of the Day and display the Medicine Name.

	 MEDICINE DATABASE Medicine Name: Date: Time of the Day: Insert
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".  MEETING SCHEDULE Date: Time: Meeting Agenda: Add Meeting Agenda  MEETING INFO Pick a date to get meeting info: / / Mon, Jul 23 JULY 2018 CANCEL OK Search
3	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.

	 SMS APPLICATION Display SMS Number Display SMS Message
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 playerthat 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 scrollfrom 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 $$\mathbf{E = P * (r(1+r)^n)/((1+r)^n-1)}$$ 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.

	 CAR EMI CALCULATOR Principal Amount: Down Payment: Interest Rate: Loan Term (in months): EMI: Result Calculate Monthly EMI

Laboratory Outcomes: After studying theselaboratory 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 oneexperiment from the lot with equal opportunity.
    - For laboratories having PART A and PART B: Students are allowed to pick oneexperiment from PART A and one experiment from PART B, with equalopportunity.

- Change of experiment is allowed only once and marks allotted for procedure to be made zero of the changed part only.

- Marks Distribution (Courseed 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",** 1st Edition, Wiley India Pvt Ltd, 2014. ISBN-13: 978-8126547197 2. Dawn Griffiths and David Griffiths, **"Head First Android Development",** 1st 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",** 3rd Edition, Big Nerd Ranch Guides, 2017. ISBN-13: 978-0134706054

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: S. G. Balekundri Institute of Technology
    - 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: Shaheen
    - 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: Shivbasav Nagar, Belagavi
    - Align: center

9) Add TextView component change the following properties:
    - Size: 20dp
    - Text: Email-shaheenm@sgbit.edu.in
    - Align: center

10) Add TextView component change the following properties:
    - Size: 20dp
    - Text: Phone-9742771004

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="S G Balekundri Institute of
        Technology"
        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" />
```

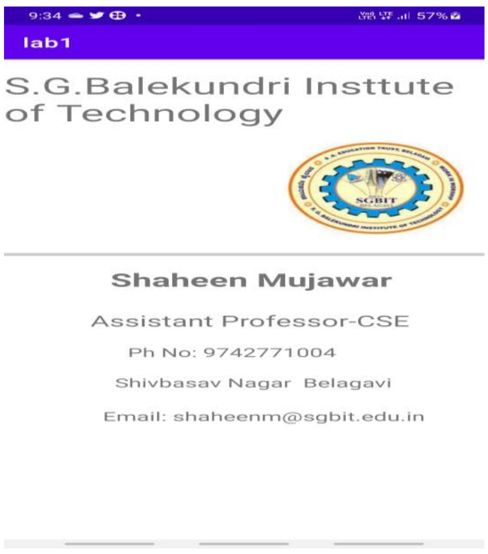
```
<TextView android:id="@+id/textView2"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_alignParentEnd="true"
    android:layout_alignParentRight="true"
    android:layout_alignParentBottom="true"
    android:layout_marginEnd="117dp"
    android:layout_marginRight="117dp"
    android:layout_marginBottom="394dp"
    android:text="Shaheen Mujawar"
    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="Assistant Professor-CSE"
    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: 9742771004"
    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="Shivabasav Nagar, Belagavi "
         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="Email:shaheenm@sgbit.edu.in
        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:



S.G.Balekundri Insttute of Technology

Shaheen Mujawar

Assistant Professor-CSE

Ph No: 9742771004

Shivbasav Nagar Belagavi

Email: shaheenm@sgbit.edu.in

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"
```

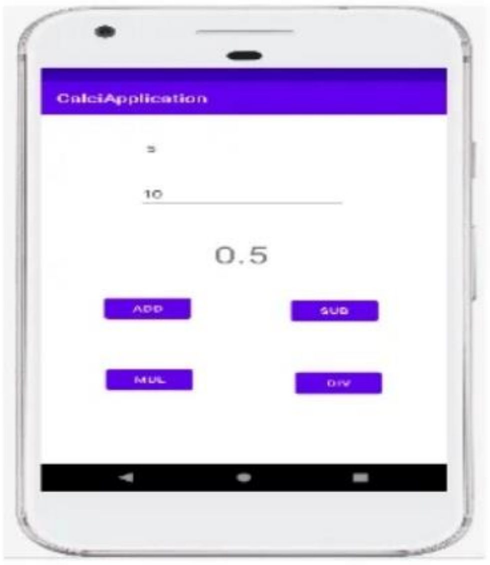
```
        android:layout_marginEnd="98dp"
        android:layout_marginBottom="653dp"
        android:text="SIMPLE CALCI"
        android:textSize="38dp"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintHorizontal_bias="0.498"
        app:layout_constraintLeft_toLeftOf="parent"
        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="ADD" />

    <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="SUB" />

    <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="MUL" />

    <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="DIV" />

</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 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.

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(getBaseContext(),"FAILED LOGIN
ATTEMPTS",Toast.LENGTH_LONG).show();
               loginBtn.setEnabled(false);
}
        });
    }
  }
```

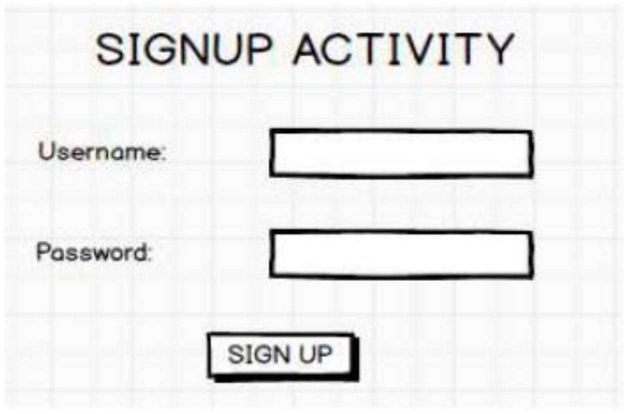
13) Right click on Java folder-> new-> activity->empty activity-> name it as "LoginSuccessful"
14) Go to xml code of design change the layout to "RelativeLayout"
15) Add TextView component & change the following properties:
    - Size: 38dp
    - Text: "Login Successful"
    - Center-Align

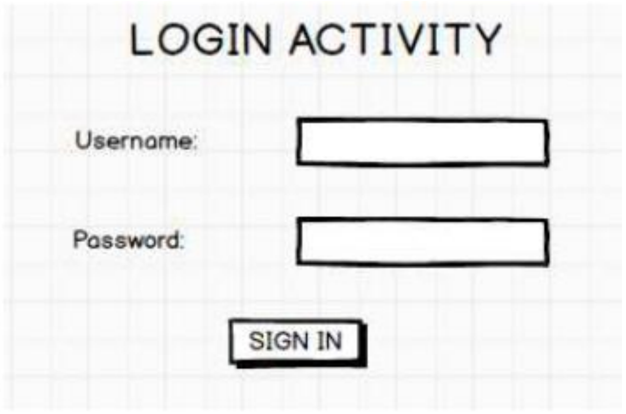
XML-CODE

```
<?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=".LoginSuccessActivity">

    <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 LoginSuccessActivity extends AppCompatActivity {
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_login_success);
```


SIGNUP ACTIVITY
Username:
Password:
SIGN UP


LOGIN ACTIVITY
Username:
Password:
SIGN IN

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.

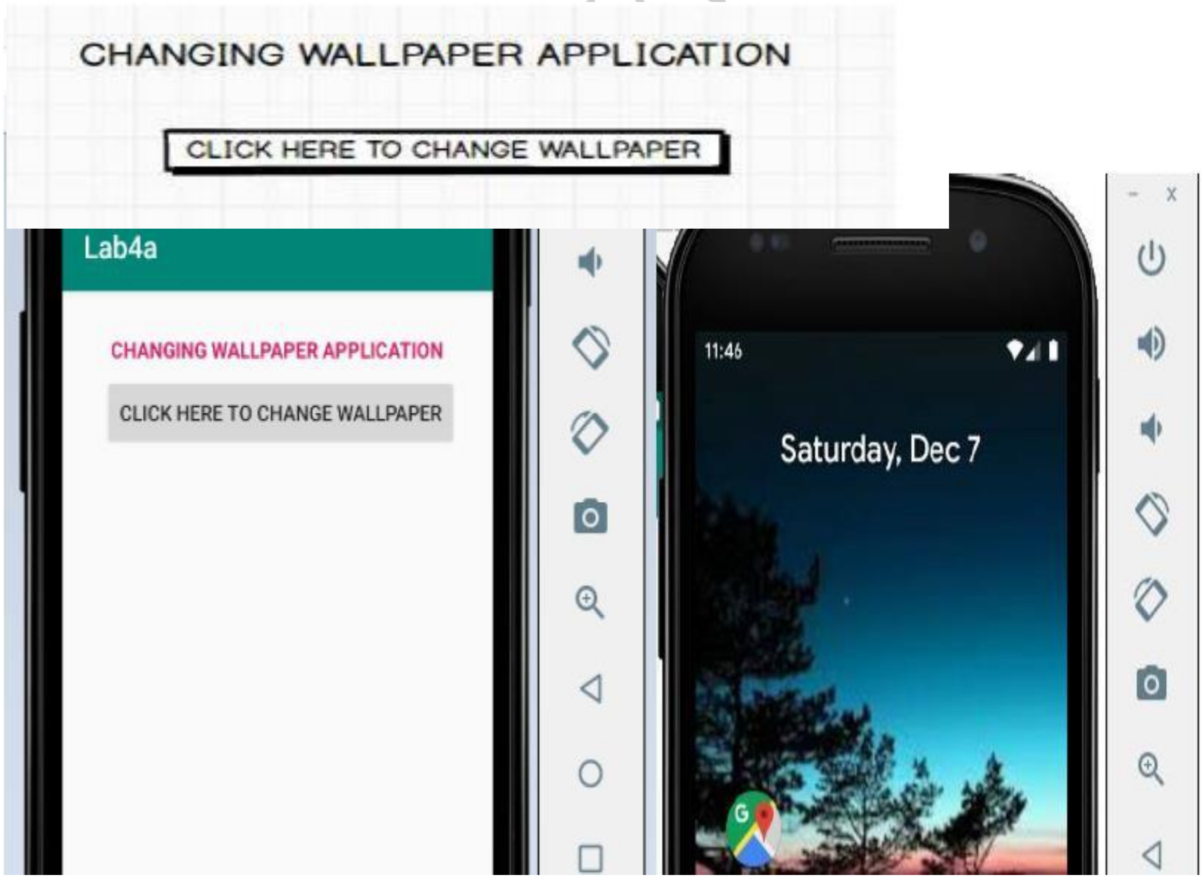
1) Firstly Create an Application by Name “WallpaperActivity”
2) Go to xml code of design change the layout to “RelativeLayout”
3) Add TextView component & change the following properties:
   - Size: 38dp
   - Text: Wall Paper Change Application
   - Center-Align
4) Add Button component & change the following 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); } }
```


CHANGING WALLPAPER APPLICATION
CLICK HERE TO CHANGE WALLPAPER
Lab4a
CHANGING WALLPAPER APPLICATION
CLICK HERE TO CHANGE WALLPAPER
11:46
Saturday, Dec 7

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 TextViewcontrol.

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"
```

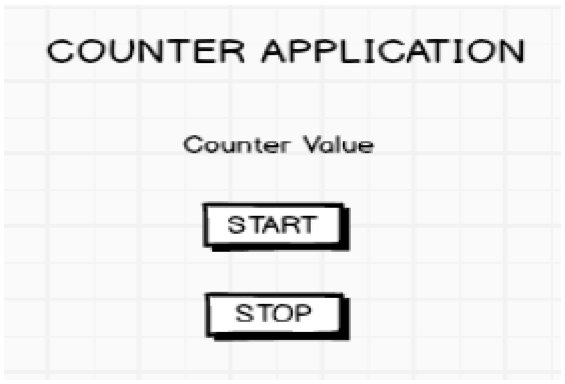
```
        android:layout_width="121dp"
        android:layout_height="32dp"
        android:layout_alignParentEnd="true"
        android:layout_alignParentBottom="true"
        android:layout_marginEnd="145dp"
        android:layout_marginBottom="478dp"
        android:text="Counter Value" />

    <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:



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->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:

```
<?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>
```

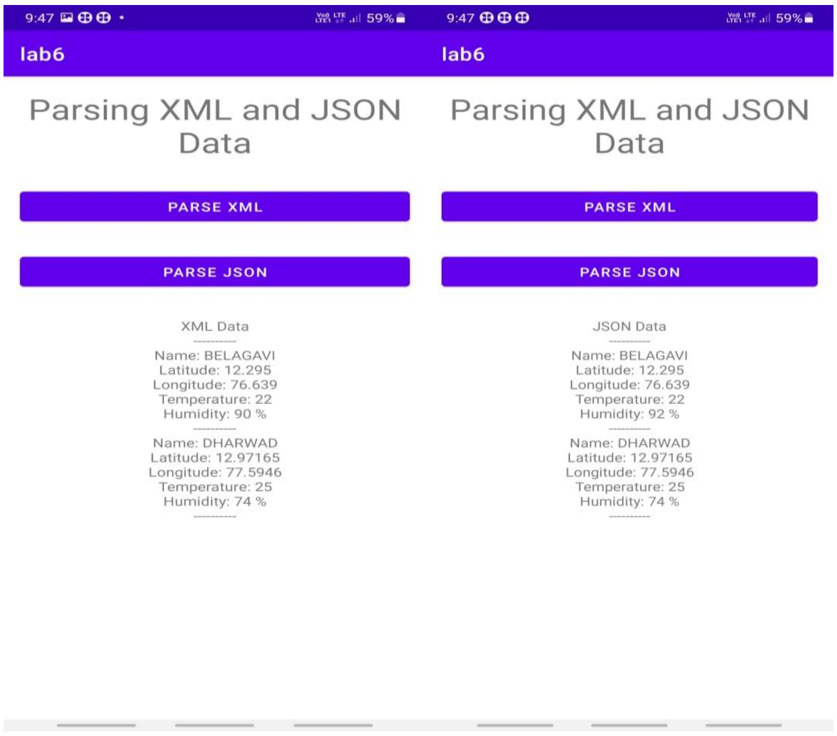
JAVA-CODE:

```
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:



```
XML Data
----------
Name: BELAGAVI
Latitude: 12.295
Longitude: 76.639
Temperature: 22
Humidity: 90 %
----------
Name: DHARWAD
Latitude: 12.97165
Longitude: 77.5946
Temperature: 25
Humidity: 74 %
----------
```

```
JSON Data
----------
Name: BELAGAVI
Latitude: 12.295
Longitude: 76.639
Temperature: 22
Humidity: 92 %
----------
Name: DHARWAD
Latitude: 12.97165
Longitude: 77.5946
Temperature: 25
Humidity: 74 %
----------
```

<u>Program-7</u>: 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"
```

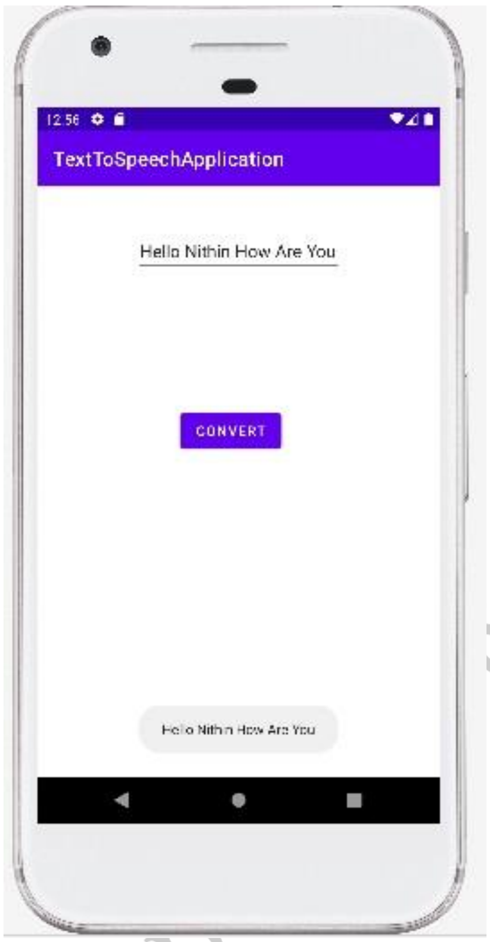
```
        android:layout_alignParentRight="true"
        android:layout_alignParentBottom="true"
        android:layout_marginEnd="101dp"
        android:layout_marginRight="101dp"
        android:layout_marginBottom="514dp"
        android:ems="10"
        android:hint="Enter the text to be converted"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

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" />
```

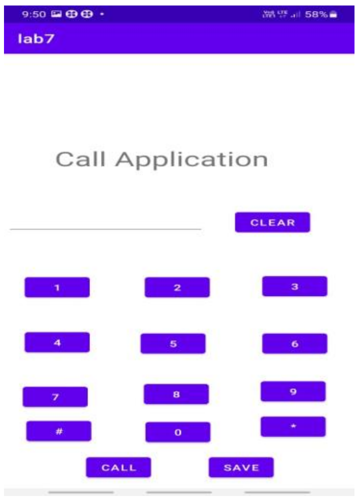
```
<Button
    android:id="@+id/button9"
    android:layout_width="76dp"
    android:layout_height="wrap_content"
    android:layout_alignParentEnd="true"
    android:layout_alignParentBottom="true"
    android:layout_marginEnd="171dp"
    android:onClick="inputNumber"
    android:layout_marginBottom="147dp"
    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:



Viva Questions

Q #1) What is Android?
Answer: Android is an open-source operating system and is mainly popular for Smartphones and Tablets.
This operating system is Linux Kernel-based. Using the Android operating system, the developer develops the functions or programs which can perform basic as well as the advanced type of operations on the Smartphone.

Q #2) What is the Android SDK?
Answer: To develop a mobile application, Android developers require some tools and this requirement is satisfied by "Android SDK" which is a set of tools that are used for developing or writing apps.
It has a Graphical User Interface that emulates the Android environment. This emulator acts like an actual mobile device on which the developers write their code and then debug/test the same code to check if anything is wrong.

Q #3) What is the difference between Mobile Application Testing and Mobile Testing?
Answer: Mobile app testing is the testing of applications on a device which mainly focuses on functions and features of the application.
And Mobile Testing is the testing of the actual mobile device and focuses on mobile features like Call, SMS, Contacts, Media Player, inbuilt browsers, etc.

Q #4) Name the languages supported for Android development.
Answer: Java is the widely used language for Android development. It also supports C/C++ and when used with Android SDK, it improves the performance speed too.

Q #5) What are the advantages of the Android Operating System?
Answer: It is open-source and platform-independent. It supports various technologies like Bluetooth, Wi-Fi, etc

Q #6) Define and explain the Android Framework.
Answer: Android framework is a set of API's using which the Android developers write code for the mobile apps. It contains the methods and classes to write the programming code.
Android framework includes a different set of tools to create image pane, text field, buttons, etc. It also includes "Activities" with which the user interacts and "Services", which are the programs that run in the background. It is a package of different components like Intents, Broadcast Receivers, Content Providers, etc.

Q #7) Which components are necessary for a New Android project?
Answer: Whenever a new Android project is created, the below components are required:
- **manifest:** It contains an **XML** file.
- **build/:** It contains build output.
- **src/:** It contains the code and resource files.
- **res/:** It contains bitmap images, UI Strings and XML Layout i.e. all non-code resources.
- **assets/:** It contains a file that should be compiled into a **.apk** file.

Q #8) Provide the important core components of Android.
Answer: The core components of Android operating systems are:
- Activity

- Intents
- Services
- Content Provider
- Fragment

Q #9) Explain briefly – what is meant by Activities?
Answer: Activities are the part of the mobile app which the user can see and interact with.
For Example, if you open an SMS app which has multiple activities like create new SMS, add a contact from the address book, write the content in the SMS body, send SMS to the selected contact, etc.
Activity keeps a track of the following:

- Keeps track of what a user is currently looking for in an app.
- Keeps a track of previously used processes, so that the user can switch between ongoing process and previous process.
- It helps to kill the processes so that the user can return to their previous state

An activity is implemented as a subclass of Activity class as shown below:

```
Public class MyActivity extends Activity
{
}
```

Q # 10) What is meant by Services?
Answer: Service is an Android component that runs in the background and acts independently. It does not provide any user interface.
Though the services are running behind the scene, a user can continue their work on different apps. Most of the time, the users are not aware of the services which are running in the background. These services allow the system to kill the process without interrupting the user's ongoing work.

A service is implemented as a subclass of Service class:

```
Public class MainService extends Service
{
}
```

Q #11) What is an Intent?
Answer: Android has an Intent class when the user has to navigate from one activity to another. Intent displays notifications from the device to the user and then the user can respond to the notification if required.
Given below are the two types:

- Implicit Intents
- Explicit Intents

Q #12) Explain Implicit and Explicit Intents.
Answer: Implicit Intent calls the system components while Explicit Intents invoke the Activity class.

Q #13) What is .apk extension in Android?
Answer: It is a default file format that is used by the Android Operating System. Application Package Kit (APK) is used for the installation of mobile apps. The .apk contains resource file, certificate, manifest file, and other code.
APK files are archive files in the zip format with .apk extension.

Q #14) What is the database used for the Android platform?
Answer: SQLite is the database that is used for the Android platform. It is an open-source, serverless database.

Q #15) What is ANR in Android?
Answer: ANR stands for Application Not Responding. It is a notification or pop-up displayed by the Android platform whenever the application is performing too many functions at a time and if it is suddenly not responding for a long time to the user action.

Q #16) Which are the dialog boxes supported by the Android platform?
Answer: Android supports four types of dialog boxes:

- **AlertDialog**: It has a maximum of 3 buttons and sometimes AlertDialog includes checkboxes and Radio buttons to select the element.
- **ProgressDialog**: It displays the progress bar or wheels.
- **TimePickerDialog**: Using this dialog box, a user selects the Time.
- **DatePickerDialog**: Using this dialog box, a user selects the Date
-

Q #17) What is ADB?
Answer: Android Debug Bridge (ADB) is a command-line tool that performs shell commands.
ADB is used for direct communication between the emulator ports. It gives direct control of the communication between the emulator instances to the developer.

Q #18) What is ActivityCreator?
Answer: ActivityCreator is a batch file and shell script which was used to create a new Android project. It is now replaced by the "Create New Project" in Android SDK.

Q #19) What is Orientation?
Answer: Orientation is the key feature in Smartphones nowadays. It has the ability to rotate the screen between Horizontal or Vertical mode.
Android supports two types of screen Orientations as mentioned below:

- **Portrait**: When your device is vertically aligned.
- **Landscape**: When your device is horizontally aligned.

setOrientation() is a method using which you can set a screen alignments. HORIZONTAL and VERTICAL are two values that can be set in the setOrientation() method. Whenever there is a change in the display orientation i.e. from Horizontal to Vertical or vice versa then onCreate() method of the Activity gets fired.

Basically, when the orientation of the Android mobile device gets changed then the current activity gets destroyed and then the same activity is recreated in the new display orientation. Android developers define the orientation in the AndroidManifest.xml file.

Q #20) What is AIDL?
Answer: In the Android platform, there are remote methods that facilitate the use of methods from one program to another. To create and implement the remote methods the first step is to define the communication interface in AIDL.
AIDL stands for Android Interface Definition Language. It facilitates communication between the client and the service. It also communicates the information through inter-process communication.

For communication between processes, the data is broken down into chunks which are easily understandable by the Android platform.

Q #21) What are the data types supported by AIDL?
Answer: Data Types supported by AIDL are as follows:

- String
- List

- Map
- charSequence
- Java data types such as INT, Long, Char, Boolean, etc
-

Q #22) Explain the AndroidManifest.xml file and why do you need this?
Answer: Every application must have an AndroidManifest.xml file in the root directory. It contains information about your app and provides the same to the Android system.
The information includes the package name, Android components such as Activity, Services, Broadcast Receivers, Content Providers, etc. Every Android system must have this information before running any app code.

AndroidManifest.xml file performs the following tasks:
- It provides a name to the Java package and this name is a unique identifier for the application.
- It describes the various components of the application which include Activity, Services, Content Providers, etc. Also, it defines the classes which implement these components.
- It is responsible to protect the application and it declares the permission for accessing the protected part of the app.
- It also declares the Android API which is going to be used by the application.
- It contains the library file details which are used and linked to the application.

Q #23) What all devices have you worked on?
Answer: There are many mobile devices available in the market with different operating systems.Specifically, I have worked on Android, Windows, Symbian, iPhone, etc

Q #24) Which tools are used for debugging on the Android platform?
Answer: To understand the cause of the failure or cause of any issue, debugging is important. On the Android platform **Android Monitor.bat** utility is used while on the iOS platform, iPhone Configuration utility is used for debugging purposes.
There are different tools for debugging which include: Android DDMS, Android Debug Bridge, iOS simulator, Debugging from Eclipse with ADT, Remote debugging on Android with Chrome, etc.

Q #25) Which scenario can test only on real devices but not on an emulator?
Answer: Emulators are used for performing similar kinds of testing which is performed on the real devices. Basically, emulators are used as a replacement for real devices as sometimes real devices are not available for testing, the use of real mobile devices for testing purposes is costlier at times.
But there are few scenarios that cannot be tested using emulator, these can be tested only using real devices. These scenarios are interrupted scenarios i.e. message, phone call interruption while using the app, low battery, Bluetooth, memory card mount and unmount, etc.

Q #26) Name the mobile automation tools that are available in the market.
Answer: There are quite a few mobile automation testing tools that are available in the market but these are used only if the project requires it and if the application supports the automation.
These tools are paid as well as free tools, hence analysis needs to be done within the project team and then the appropriate mobile automation tool needs to be selected. Silk Mobile, SeeTest, Ranorex are the paid mobile automation tool while Appium, KIF, Robotium, Calabash are few free tools.

Q #27) How do you troubleshoot the android application which is crashing frequently?

Answer: Given below are the few steps that we need to follow while troubleshooting the crashing issue:

- **Free up memory space**: There is only limited space available on mobile devices for mobile apps. To avoid crashing issues or memory-related issues, you need to first check the memory space.
- **Clear app data usage**: You can clear the app data using the Application Manager under "Settings". This will clear the cache memory and allow some free space to install another app or it will boost up your current app.
- **Memory Management**: Some apps run perfectly on one type of mobile device but the same app may not work on another type of device as for such devices the processing power, memory management, and CPU speed is different. For any app to run properly on any type of mobile device, you should manage the memory on the device.
- **Compatibility issue**: It is always not possible to test mobile apps on all mobile devices, browsers, operating systems, etc. So you need to test your mobile app on as many mobile devices as you can in order to avoid any compatibility issue.

Q #28) How do you find memory leaks in the mobile app on the Android platform?

Answer: Android Studio is using Android Device Manager (ADM), this ADM is used to detect the memory leaks in the Android platform.
When you open ADM in the Android Studio then on the left-hand side of the ADM, you will find your device or emulator in which a heap sign will be displayed. When you are running any mobile app then you will see the heap size, memory analysis and other statistics displayed on it.

Q #29) What is DDMS?

Answer: Android Studio has debugging tools known as DDMS i.e. Dalvik Debug Monitor Server.
It has wide debugging features which include:

- Port forwarding services.
- Screen capture on the device.
- Thread and Heap information.
- Incoming call and SMS spoofing.
- Logcat
- Radio state information.
- Location data spoofing.

DDMS is integrated with the Android studio. To launch the DDMS, you need to open the Android Device Monitor (ADM) first and then click on the DDMS menu button. Once DDMS is launched, then on the left-hand side the list of connected devices is displayed along with the processes which are running on each device.

With the help of DDMS, you can debug both on real devices and emulators.

Q #30) What are the different data storage options available on the Android platform?

Answer: Android platform provides a wide range of data storage options. These options must be used based on the need such as data is secure and used with permission only or can be accessed publicly.
Below is the list of data storage options on the Android platform:

- **SharedPreference**: It stores data in XML files. It is the simplest way to store private data in the key-value pair.
- **SQLite**: It stores structured data in the private database.
- **Internal Storage**: It stores data in the device file system and any other app cannot read this data.