BANGALORE INSTITUTE OF TECHNOLOGY

K.R. ROAD, V.V PURAM, BANGALORE – 560 004

DEPARTMENT OF
INFORMATION SCIENCE AND ENGINEERING

(AFFILIATED TO VTU, BELAGAVI)

SUBJECT CODE: 18CSMP68

MOBILE APPLICATION DEVELOPMENT LABORATORY

As per Choice Based Credit System Scheme (CBCS)

FOR VI SEMESTER CSE/ISE AS PRESCRIBED BY VTU

Academic year 2020-2021

Prepared By:
Prof. Mohan Babu G
Assistant Professor
Dept. of ISE, BIT

BANGALORE INSTITUTE OF TECHNOLOGY

VISION

To establish and develop the Institute as the center of higher learning, ever abreast with expanding horizon of knowledge in the field of Engineering and Technology with entrepreneurial thinking, leadership excellence for life-long success and solve societal problems.

MISSION

- Provide high quality education in the Engineering disciplines from the undergraduate through doctoral levels with creative academic and professional programs.
- Develop the Institute as a leader in Science, Engineering, Technology, Management and Research and apply knowledge for the benefit of society.
- Establish mutual beneficial partnerships with Industry, Alumni, Local, State and Central Governments by Public Service Assistance and Collaborative Research.
- Inculcate personality development through sports, cultural and extracurricular activities and engage in the social, economic and professional challenges.

QUALITY POLICY

- We strive to provide world-class education by anticipating and satisfying the changing requirement of technical, engineering knowledge and managerial skills of industry and institutions in specific and society in general.

Bangalore Institute of Technology

K R Road, VV Pura, Bengaluru - 560004

Department of Information Science and Engineering

VISION

Empower every student to be innovative, creative and productive in the field of Information Technology by imparting quality technical education, developing skills and inculcating human values.

MISSION

- To evolve continually as a centre of excellence in offering quality Information Technology Education.
- To nurture the students to meet the global competency in industry for Employment.
- To promote collaboration with industry and academia for constructive interaction to empower Entrepreneurship.
- To provide reliable, contemporary and integrated technology to support and facilitate Teaching, Learning, Research and Service

PROGRAM EDUCATIONAL OBJECTIVES

- Uplift the students through Information Technology **Education.**
- Provide exposure to emerging technologies and train them to **Employable** in Multi-disciplinary industries.
- Motivate them to become good professional Engineers and **Entrepreneur.**
- Inspire them to prepare for **Higher Learning and Research.**

PROGRAMME SPECIFIC OUTCOMES (PSOs)

- To provide our graduates with **Core Competence** in **Information Processing and Management.**
- To provide our graduates with higher learning in **Computing Skills.**

PROGRAM OUTCOMES (POs)

Engineering Graduates will be able to:

1. **Engineering knowledge**: Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.

2. **Problem analysis**: Identify, formulate, review research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences.

3. **Design/development of solutions**: Design solutions for complex engineering problems a n d design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations.

4. **Conduct investigations of complex problems**: Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.

5. **Modern tool usage**: Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modeling to complex engineering activities with an understanding of the limitations.

6. **The engineer and society**: Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice.

7. **Environment and sustainability**: Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development.

8. **Ethics**: Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.

9. **Individual and team work**: Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings.

10. **Communication**: Communicate effectively on complex engineering activities with the engineering community and with society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions.

11. **Project management and finance**: Demonstrate knowledge and understanding of t h e engineering and management principles and apply these to one's own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.

12. **Life-long learning**: Recognize the need for, and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change.

Bangalore Institute of Technology
K.R. Road, V.V.Pura, Bengaluru - 560004
Department of Information Science and Engineering

Pre-requisites:

- Knowledge about XML programming Languages.
- Knowledge about JAVA programming Languages.
- Basic concepts of Android Operating System.

Course Objectives:

This laboratory (18CSMP68) will enable students to:

- ❖ Learn and acquire the art of Android Programming.
- ❖ Configure Android 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.

Course Outcomes:

The students should be able to:

1. **Apply** to Implement adaptive, responsive user interfaces that work across a wide range of devices
2. **Analyze** the different methods in storing, sharing and retrieving data in Android applications.
3. **Evaluate** the role of permissions and security for Android applications.
4. **Create**, test and debug Android application by setting up Android development environment
5. **Apply** the concepts of android programming to **develop** mobile application for **experiment**. **Present** the results to the Lab handling faculties individually or to the batch mates using presentation slides, hence raising the **oral and written communication skills**.

Resources Required:

- Hardware resources
  - Desktop PC
  - Windows Operating System
- **Software resources**
  - Android studio

Mapping of COs-POs and COs-PSOs

Mobile Application Development (18CSMP68)

Year of Study: 2020 -2021 (EVEN)

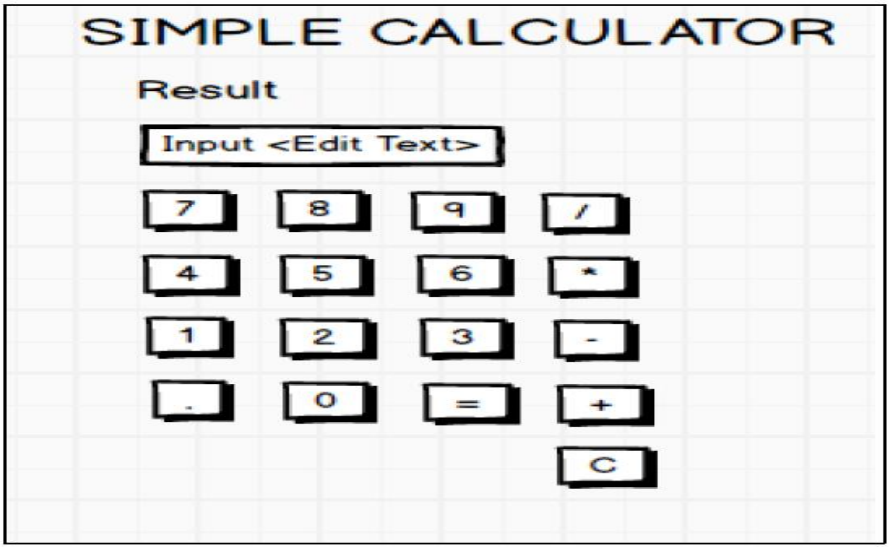
		PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
18CSMP68	CO1	2	2	2						1	1		1
	CO2	3	3	2		2	1			1	1		1
	CO3	3	3	3		2	1			1	1		1
	CO4	3	3	2		2	1		2	1	1		1
	CO5	3	3	2		3				3	3		2

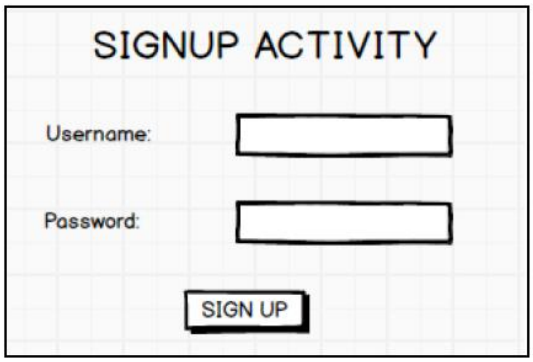
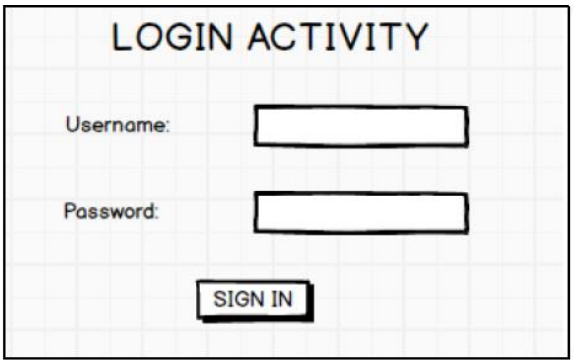
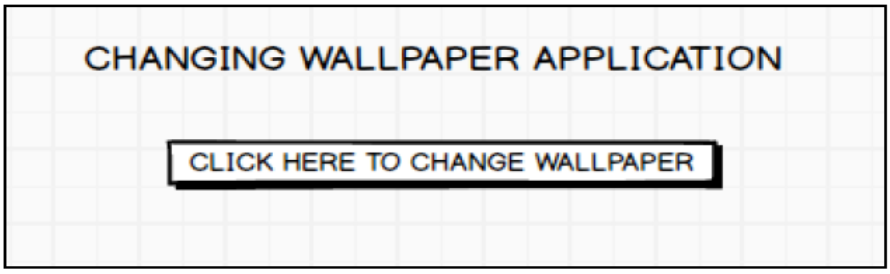
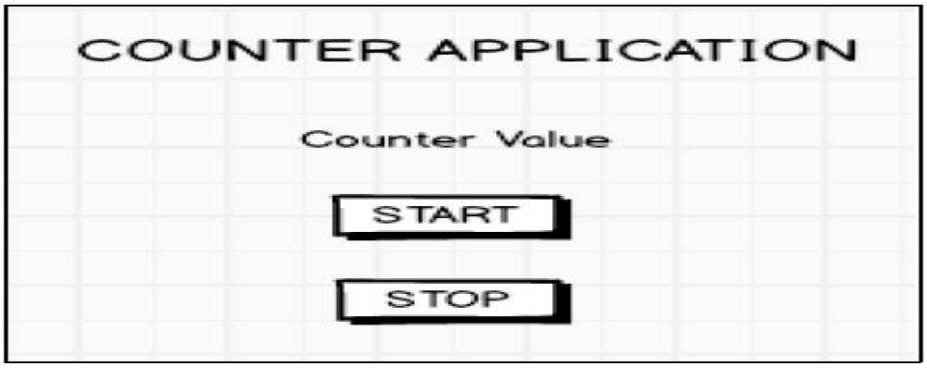
		PSO1	PSO2
Mobile Application Development (18CSMP68)	CO1	2	2
	CO2	3	3
	CO3	3	3
	CO4	2	2
	CO5	3	3

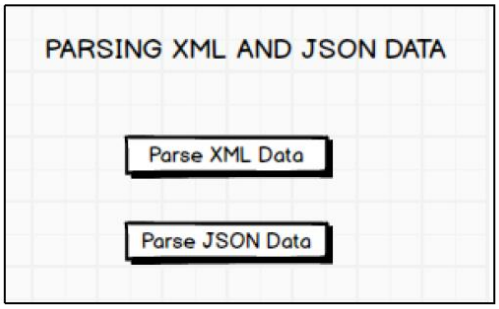
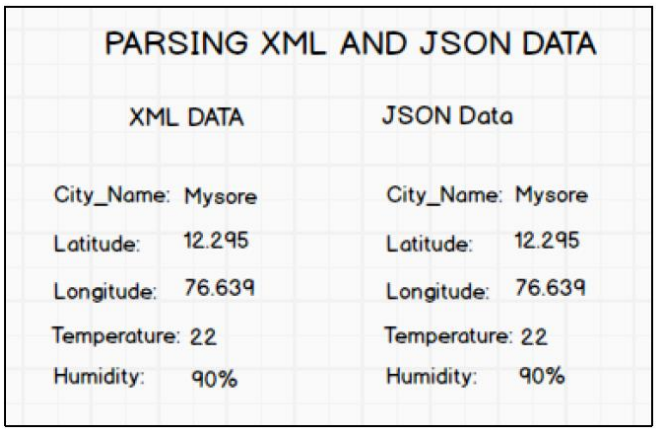
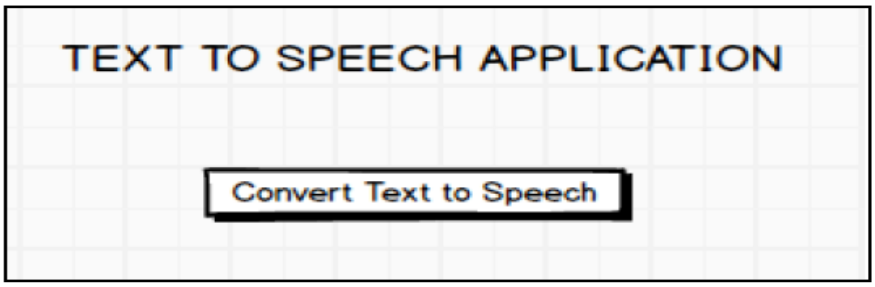
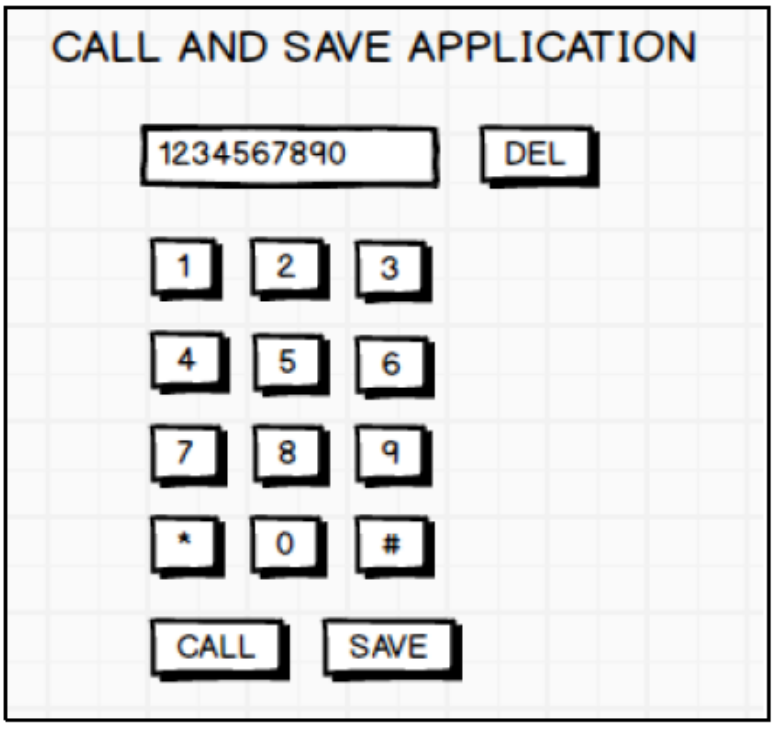
Mobile Application Development Laboratory

Subject Code: 18CSMP68 **IA Marks : 40**
Hours/Week: 03 **Exam Hours : 3**
Total Hours: 40 **Exam Marks : 60**

List of Programs

Sl. No	Name of Experiment
	PART A
1	Create an application to design a Visiting Card. The Visiting card should have a company logo at the top right corner. The company name should be displayed in Capital letters, aligned to the center. Information like the name of the employee, job title, phone number, address, email, fax and the website address is to be displayed. Insert a horizontal line between the job title and the phone number. 
2	Develop an Android application using controls like Button, TextView, EditText for designing a Calculator having basic functionality like Addition, Subtraction, Multiplication, and Division. 

3	Create a SIGN Up activity with Username and Password. Validation of password should happen based on the following rules: • • Password should contain uppercase and lowercase letters. • • Password should contain letters and numbers. • • Password should contain special characters. • • Minimum length of the password (the default value is 8). On successful **SIGN UP** proceed to the next Login activity. Here the user should **SIGN IN** using the Username and Password created during signup activity. If the Username and Password are matched then navigate to the next activity which displays a message saying "Successful Login" or else display a toast message saying "Login Failed". The user is given only two attempts and after that display a toast message saying "Failed Login Attempts" and disable the SIGN IN button. Use Bundle to transfer information from one activity to another.  
4	Develop an application to set an image as wallpaper. On click of a button, the wallpaper image should start to change randomly every 30 seconds. 
5	Write a program to create an activity with two buttons START and STOP. On Pressing of the START button, the activity must start the counter by displaying the numbers from One and the counter must keep on counting until the STOP button is pressed. Display the counter value in a TextView control. 

6	Create two files of XML and JSON type with values for City_Name, Latitude, Longitude, Temperature, and Humidity. Develop an application to create an activity with two buttons to parse the XML and JSON files which when clicked should display the data in their respective layouts side by side.  
7	Develop a simple application with one Edit Text so that the user can write some text in it. Create a button called “Convert Text to Speech” that converts the user input text into voice. 
8	Create an activity like a phone dialer with CALL and SAVE buttons. On pressing the CALL button, it must call the phone number and on pressing the SAVE button it must save the number to the phone contacts. 

Mobile Application Development Laboratory Evaluation Rubrics

Subject Code: 18CSMP68 **IA Marks : 40**
Hours/Week: 03 **Exam Hours : 03**
Total Hours: 40 **Exam Marks : 60**

Evaluation Criteria

	Daily conduction (15 Marks)			
	Task	**Excellent**	**Good**	**Fair**
a.	**Understanding of problem and approach to solve. (5 Marks)**	Demonstrate good knowledge of design concepts of given problem and implementation in Java and XML. (5)	Moderate understanding of the android application and its implementation in Java. (3)	Incomplete knowledge of requirement. (1)
b.	**Execution and Testing (5 Marks)**	Program has all possible conditions while execution with proper results. (5)	Average conditions are defined and verified during execution. (3)	Many functions are not working (1)
c.	**Results and Documentation (5 Marks)**	Meticulous documentation of results obtained once emulator is run (5)	Acceptable documentation shown in emulator (3)	Working for only one Activity/Button (1)

	Test Conduction (10 Marks)			
	Task	**Excellent**	**Good**	**Fair**
a.	**Program Write up (3 Marks)**	Demonstrate detailed knowledge of requirements of application and coding (Both XML and Java) to build android application for the given activity (3)	Moderate knowledge of requirements of application and coding (Both XML and Java) to build android application for the given activity (2)	No requirement analysis. (1)
b.	**Implementation and Execution (3 Marks)**	Application works for all possible conditions while execution of the program with satisfying results. (3)	Average conditions are defined and verified in the code. (2)	No Execution. (1)
c.	**Evaluation (2 Marks)**	Evaluate all working of all the activities in application possible results (2)	Partial Evaluation working of all the activities in application possible results (1)	No proper evaluation results (0)
d.	**Viva-voce (2 Marks)**	Excellent Knowledge overall android programming concepts and design requirements of an application. (2)	Good Knowledge overall android programming concepts and design requirements of an application (1)	Poor Knowledge overall android programming concepts and design requirements of an application (0)

	Mini Project (Max: 10 Marks)			
	Task	**Excellent**	**Good**	**Fair**
a.	**Application Development (6 Marks)**	Develop the application for all the features as per requirement specification and design. (6 Marks)	Moderate development of application as per requirement specification and design. (5 Marks)	Partial development of application as per requirement specification and design. (3 Mark)
b.	**Demonstration and Report preparation (4 Marks)**	Complete presentation of the application as per the requirement specification and report preparation (4 Marks)	Average Presentation of the application as per the requirement specification and report preparation (3 Marks)	Partial Presentation of the application as per the requirement specification and report preparation (2 Mark)

Mobile Application Development Laboratory

Subject Code: 18CSMP68 **IA Marks : 40**
Hours/Week: 03 **Exam Hours : 03**
Total Hours: 40 **Exam Marks : 60**

Lesson Planning / Schedule of Experiments

Sl. No	Name of the Experiment	To be completed
1	Pre-requisites for the lab and discussion of COs, POs, PEOs and PSOs. Discussion of Vision, Mission of department and college	Week 1
2	Program 1 : Create an application to design a Visiting Card	Week 2
3	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	Week 3
4	Program 3: Create a SIGN Up activity with Username and Password, SIGN IN using the Username and Password created during signup activity	Week 4
5	Program 4: Develop an application to set an image as wallpaper	Week 5
6	Discussion of mini project topic selection	Week 6
7	Lab Test-I	Week 7
8	Mini Project Presentation	Week 8
9	Program 5: Write a program to create an activity with two buttons START and STOP	Week 9
10	Program 6: Create two files of XML and JSON type with values for City_Name, Latitude, Longitude, Temperature, and Humidity	Week 10
11	Program 7: Develop a simple application with one Edit Text so that the user can write some text in it.	Week 11
12	Program 8: Create an activity like a phone dialer with CALL and SAVE buttons.	Week 12
13	Lab Test-II	Week 13
14	Mini Project discussion	Week 14
15	Demonstration of mini project	Week 15

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: This laboratory (18CSMP68) will enable students to

- Learn and acquire the art of Android Programming.
- Configure Android 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.

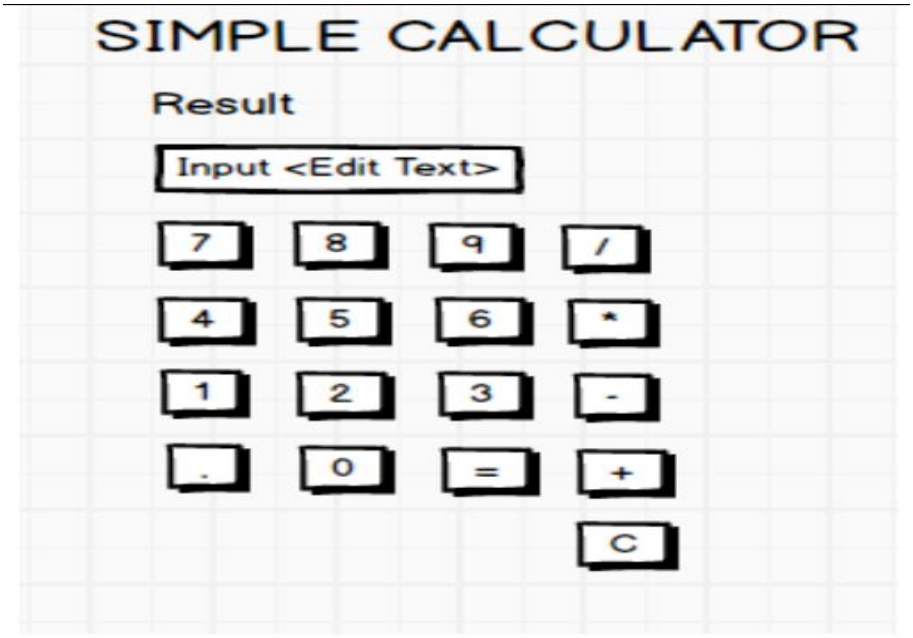
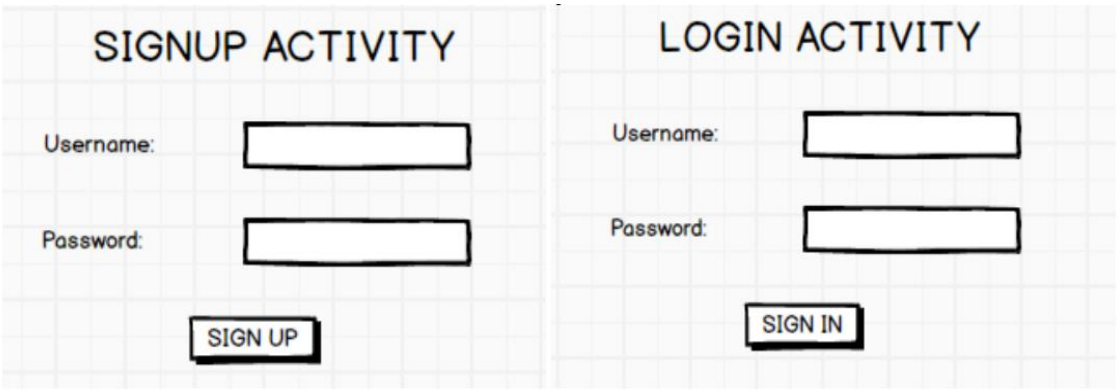
Descriptions (if any):

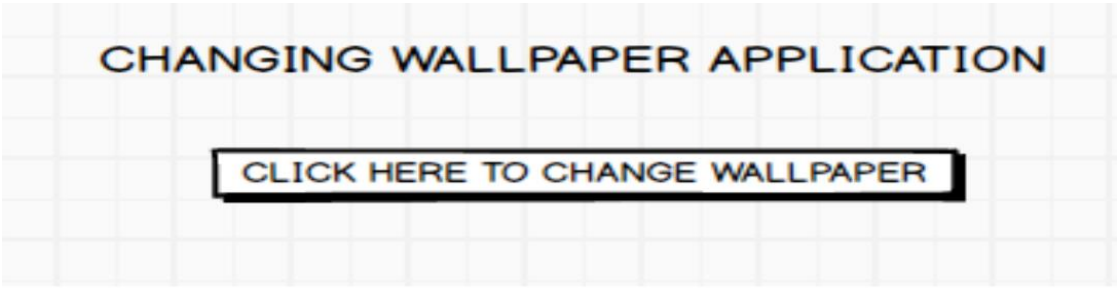
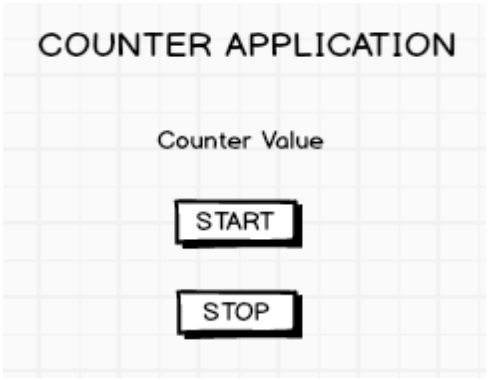
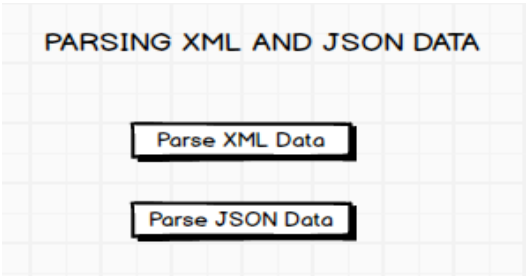
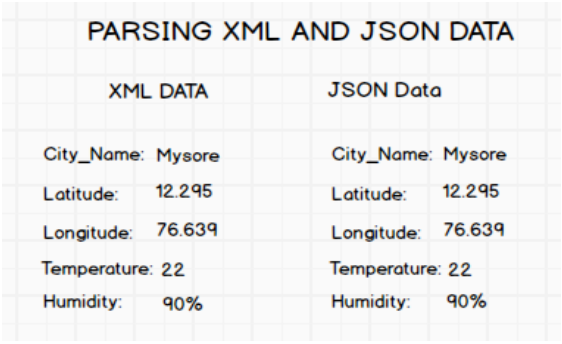
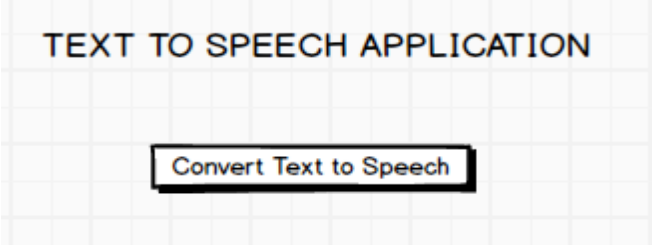
1. The installation procedure of the Android Studio/Java software must be demonstrated and carried out in groups.
2. Students should use the latest version of Android Studio/Java/Kotlin to execute these programs. Diagrams given are for representational purposes only, students are expected to improvise on them.
3. **Part B programs should be developed as an application and are to be demonstrated as a mini project in a group by adding extra features or the students can also develop their application and demonstrate it as a mini-project. (Projects/programs are not limited to the list given in Part B).**

Programs List:

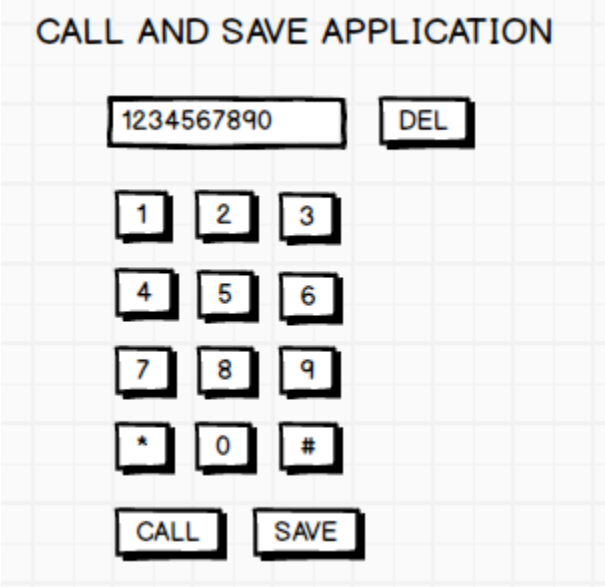
PART – A

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

2	Develop an Android application using controls like Button, TextView, EditText for designing a calculator having basic functionality like Addition, Subtraction, Multiplication, and Division. 
3	Create a SIGN Up activity with Username and Password. Validation of password should happen based on the following rules: • Password should contain uppercase and lowercase letters. • Password should contain letters and numbers. • Password should contain special characters. • Minimum length of the password (the default value is 8). On successful SIGN UP proceed to the next Login activity. Here the user should SIGN IN using the Username and Password created during signup activity. If the Username and Password are matched then navigate to the next activity which displays a message saying "Successful Login" or else display a toast message saying "Login Failed". The user is given only two attempts and after that display a toast message saying "Failed Login Attempts" and disable the SIGN IN button. Use Bundle to transfer information from one activity to another. 

4	Develop an application to set an image as wallpaper. On click of a button, the wallpaper image should start to change randomly every 30 seconds. 
5	Write a program to create an activity with two buttons START and STOP. On pressing of the START button, the activity must start the counter by displaying the numbers from One and the counter must keep on counting until the STOP button is pressed. Display the counter value in a TextView control. 
6	Create two files of XML and JSON type with values for City_Name, Latitude, Longitude, Temperature, and Humidity. Develop an application to create an activity with two buttons to parse the XML and JSON files which when clicked should display the data in their respective layouts side by side.  
7	Develop a simple application with one EditText so that the user can write some text in it. Create a button called “Convert Text to Speech” that converts the user input text into voice. 

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



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

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

Procedure to Conduct Practical Examination

- **Experiment distribution**
  - For laboratories having only one part: Students are allowed to pick one experiment from the lot with equal opportunity.
  - For laboratories having PART A and PART B: Students are allowed to pick one experiment from PART A and one experiment from PART B, with equal opportunity.

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

- Marks Distribution (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

PART A

Program No. 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.

Procedure:

1. **Create an Application by name "Labpgm1"**
2. **Go to xml code of design change the layout to** "RelativeLayout" and set width and height as "match_parent"

3. **Add TextView component and change the following properties:**

   - Text: "BANGALORE INSTITUTE OF TECHNOLOGY"
   - Align Center
   - Size: 24sp
   - Text Color "#D34848"

4. **Add ImageView to design and in type choose "IC_LAUNCHER_FOREGROUND"**

   - Download the logo & copy the same under res->drawable folder
   - In xml code of ImageView change app:srcCompat="@drawable/bitlogo"
   - Align right top

5. **Add View component & change the following properties:**
   - Height: 4dp
   - Background: "#D34848"

6. **Add TextView component change the following properties:**
   - Size: 20dp
   - Text: Vijaya Kumar N L
   - Style: Bold
   - Align center

7. **Add TextView component change the following properties:**

- Size: 16sp
- Text: Programmer
- Align center

8. **Add TextView component change the following properties:**
   - Size: 16sp
   - Text: 9663208741
   - Align: center

9. **Add TextView component change the following properties:**
   - Size: 16sp
   - Text: Dept. of Information Science and Engineering
   - Align: center

10. **Add TextView component change the following properties:**
    - Size: 20dp
    - Text: Email : vijayakumarnl@bit-bangalore.edu.in
    - Align: center

XML Code:

Activity_main.xml

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


    <TextView
        android:id="@+id/textView"
        android:layout_width="326dp"
        android:layout_height="85dp"
        android:layout_alignParentEnd="true"
        android:layout_alignParentRight="true"
        android:layout_alignParentBottom="true"
        android:layout_marginEnd="86dp"
        android:layout_marginRight="86dp"
        android:layout_marginBottom="550dp"
        android:text="BANGALORE INSTITUTE OF TECHNOLOGY"
        android:textAlignment="center"
        android:textColor="#D34848"
        android:textSize="24sp" />

    <View
```

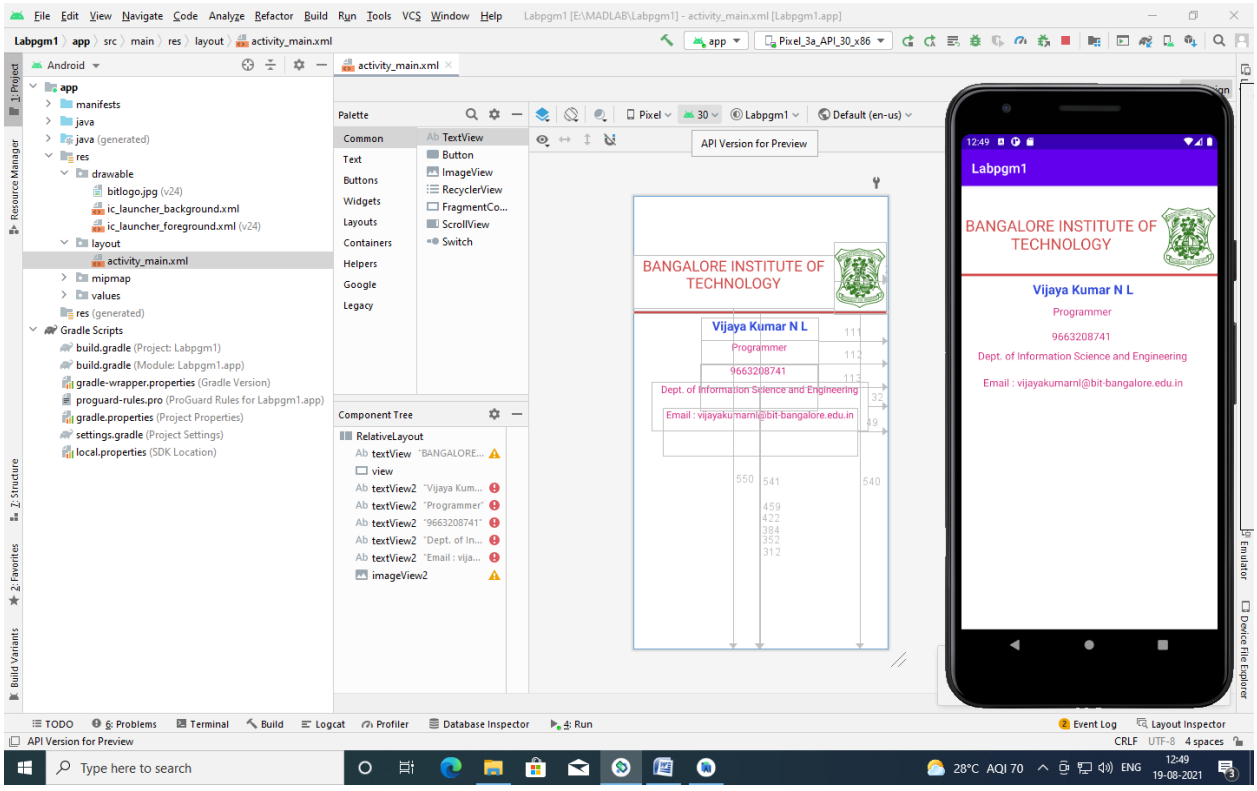
```
        android:id="@+id/view"
        android:layout_width="match_parent"
        android:layout_height="4dp"
        android:layout_alignParentBottom="true"
        android:layout_marginBottom="541dp"
        android:background="#D34848" />



    <TextView
        android:id="@+id/textView2"
        android:layout_width="190dp"
        android:layout_height="76dp"
        android:layout_alignParentEnd="true"
        android:layout_alignParentRight="true"
        android:layout_alignParentBottom="true"
        android:layout_marginEnd="111dp"
        android:layout_marginRight="111dp"
        android:layout_marginBottom="459dp"
        android:text="Vijaya Kumar N L"
        android:textStyle="bold"
        android:textAlignment="center"
        android:textColor="#314DDC"
        android:textSize="20sp" />

    <TextView
        android:id="@+id/textView2"
        android:layout_width="190dp"
        android:layout_height="76dp"
        android:layout_alignParentEnd="true"
        android:layout_alignParentRight="true"
        android:layout_alignParentBottom="true"
        android:layout_marginEnd="112dp"

        android:layout_marginRight="112dp"
        android:layout_marginBottom="422dp"
        android:text="Programmer"
        android:textAlignment="center"
        android:textColor="#D32782"
        android:textSize="16sp" />

    <TextView
        android:id="@+id/textView2"
        android:layout_width="190dp"
        android:layout_height="76dp"
        android:layout_alignParentEnd="true"
        android:layout_alignParentRight="true"
        android:layout_alignParentBottom="true"
        android:layout_marginEnd="113dp"
        android:layout_marginRight="113dp"
        android:layout_marginBottom="384dp"
        android:text="9663208741"
        android:textAlignment="center"
        android:textColor="#D32782"
```

```
        android:textSize="16sp" />

    <TextView
        android:id="@+id/textView2"
        android:layout_width="349dp"
        android:layout_height="78dp"
        android:layout_alignParentEnd="true"
        android:layout_alignParentRight="true"
        android:layout_alignParentBottom="true"
        android:layout_marginEnd="32dp"
        android:layout_marginRight="32dp"
        android:layout_marginBottom="352dp"
        android:text="Dept. of Information Science and Engineering"
        android:textAlignment="center"
        android:textColor="#D32782"
        android:textSize="16sp" />

    <TextView
        android:id="@+id/textView2"
        android:layout_width="314dp"
        android:layout_height="77dp"
        android:layout_alignParentEnd="true"
        android:layout_alignParentRight="true"
        android:layout_alignParentBottom="true"
        android:layout_marginEnd="49dp"
        android:layout_marginRight="49dp"
        android:layout_marginBottom="312dp"
        android:text="Email : vijayakumarnl@bit-bangalore.edu.in"
        android:textAlignment="center"
        android:textColor="#D32782"
        android:textSize="16sp" />

    <ImageView
        android:id="@+id/imageView2"
        android:layout_width="84dp"
        android:layout_height="116dp"
        android:layout_alignParentEnd="true"
        android:layout_alignParentRight="true"
        android:layout_alignParentBottom="true"
        android:layout_marginEnd="2dp"
        android:layout_marginRight="2dp"
        android:layout_marginBottom="540dp"
        app:srcCompat="@drawable/bitlogo" />


</RelativeLayout>
```

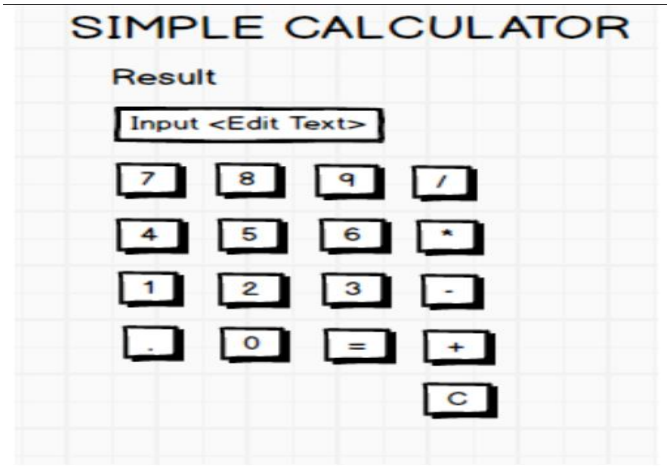
JAVA Code:

```
package com.example.labpgm1;

import androidx.appcompat.app.AppCompatActivity;

import android.os.Bundle;

public class MainActivity extends AppCompatActivity {

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

Output:



Program No. 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 “LinearLayout”
3. Add “EditText” component & change the following properties:
    a. Android hint “Enter First Number”
    b. Assign id: “@+id/num1”
    c. TextSize as “20sp”
    d. Center-Align

4. Add “EditText” component & change the following properties:
    a. Android hint “Enter Second Number”
    b. Assign id: “@+id/num2”
    c. TextSize as “20sp”
    d. Center-Align

5. Add “TextView” component & change the following properties:
    a. Android hint “Result”
    b. Assign id: “@+id/result”
    c. TextSize as “20sp”
    d. Center-Align

6. Create a LinearLayout under the main layout & change the following properties:
    a. assign width and height as “wrap_content”
    b. Layout_gravity “center”
    c. Android orientation “horizontal”

7. Add “Button” component & change the following properties:
    a. android id: “@+id/seven”
    b. android text “7”
    c. android onclick “seven”
    d. Center-Align

8. Repeat the above step for the numbers “8”, “9” and symbol “/”:
9. Repeat the steps 6 and 7 for rest of the numbers and symbols:

XML Code:

Activity_main.xml

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

    <EditText
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:hint="Enter First Number"
        android:id="@+id/num1"
        android:textSize="20sp"
        android:textAlignment="center"
        android:textColor="@color/design_default_color_secondary"          />

    <EditText
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:hint="Enter Second Number"
        android:id="@+id/num2"
        android:textSize="20sp"
        android:textAlignment="center"
        android:textColor="@color/design_default_color_secondary"          />

    <TextView
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:hint="Result"
        android:id="@+id/result"
        android:textSize="20sp"
        android:textAlignment="center"
        android:textColor="@color/design_default_color_secondary"         />

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

        <Button
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:id="@+id/seven"
            android:text="7"
            android:textAlignment="center"
            android:onClick="seven"              />

        <Button
```

```
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:id="@+id/eight"
            android:text="8"
            android:textAlignment="center"
            android:onClick="eight"             />

        <Button
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:id="@+id/nine"
            android:text="9"
            android:textAlignment="center"
            android:onClick="nine"              />

        <Button
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:id="@+id/divide"
            android:text="/"
            android:textAlignment="center"
            android:onClick="divide"             />



    </LinearLayout>

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

        <Button
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:id="@+id/four"
            android:text="4"
            android:textAlignment="center"
            android:onClick="four"              />

        <Button
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:id="@+id/five"
            android:text="5"
            android:textAlignment="center"
            android:onClick="five"              />

        <Button
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:id="@+id/six"
            android:text="6"
            android:textAlignment="center"
```

```
            android:onClick="six"              />

        <Button
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:id="@+id/multiply"
            android:text="*"
            android:textAlignment="center"
            android:onClick="multiply"              />


    </LinearLayout>

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

        <Button
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:id="@+id/one"
            android:text="1"
            android:textAlignment="center"
            android:onClick="one"              />

        <Button
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:id="@+id/two"
            android:text="2"
            android:textAlignment="center"
            android:onClick="two"              />

        <Button
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:id="@+id/three"
            android:text="3"
            android:textAlignment="center"
            android:onClick="three"              />

        <Button
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:id="@+id/minus"
            android:text="-"
            android:textAlignment="center"
            android:onClick="sub"              />


    </LinearLayout>

    <LinearLayout
```

```
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_gravity="center"
        android:orientation="horizontal">

        <Button
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:id="@+id/dot"
            android:text="."
            android:textAlignment="center"
            android:onClick="dot"             />

        <Button
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:id="@+id/zero"
            android:text="0"
            android:textAlignment="center"
            android:onClick="zero"             />

        <Button
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:id="@+id/equal"
            android:text="="
            android:textAlignment="center"
            android:onClick="compute"             />

        <Button
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:id="@+id/add"
            android:text="+"
            android:textAlignment="center"
            android:onClick="add"             />


    </LinearLayout>

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

        <Button
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:id="@+id/clear_one_digit_at_a_time"
            android:text="C"
            android:textAlignment="center"
            android:onClick="clear_one_digit_at_a_time"             />

        <Button
```

```
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:id="@+id/clear_one_field_completly"
            android:text="CE"
            android:textAlignment="center"
            android:onClick="clear_one_field_completly"              />

        <Button
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:id="@+id/all_clear"
            android:textAlignment="center"
            android:text="AC"
            android:onClick="all_clear"             />
    </LinearLayout>

</LinearLayout>
```

JAVA Code:

MainActivity.java

```
package com.example.labpgm2;

import androidx.appcompat.app.AppCompatActivity;

import android.os.Bundle;
import android.view.View;
import android.widget.EditText;
import android.widget.TextView;
import android.widget.Toast;

public class MainActivity extends AppCompatActivity {
    EditText number1, number2;
    TextView res;
    char op;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        number1=findViewById(R.id.num1);
        number2=findViewById(R.id.num2);
        res=findViewById(R.id.result);

    }

    public void one(View v)
    {
        if(number1.hasFocus())
        {
            number1.append("1");
```

```
        }
        else if(number2.hasFocus())
        {
            number2.append("1");
        }
        else
        {
            Toast.makeText(this, "Please get the focus of First or second
number", Toast.LENGTH_LONG).show();
        }
    }
    public void two(View v)
    {
        if(number1.hasFocus())
        {
            number1.append("2");
        }
        else if(number2.hasFocus())
        {
            number2.append("2");
        }
        else
        {
            Toast.makeText(this, "Please get the focus of First or second
number", Toast.LENGTH_LONG).show();
        }
    }

    public void three(View v)
    {
        if(number1.hasFocus())
        {
            number1.append("3");
        }
        else if(number2.hasFocus())
        {
            number2.append("3");
        }
        else
        {
            Toast.makeText(this, "Please get the focus of First or second
number", Toast.LENGTH_LONG).show();
        }
    }

    public void four(View v)
    {
        if(number1.hasFocus())
        {
            number1.append("4");
        }
        else if(number2.hasFocus())
        {
            number2.append("4");
        }
```

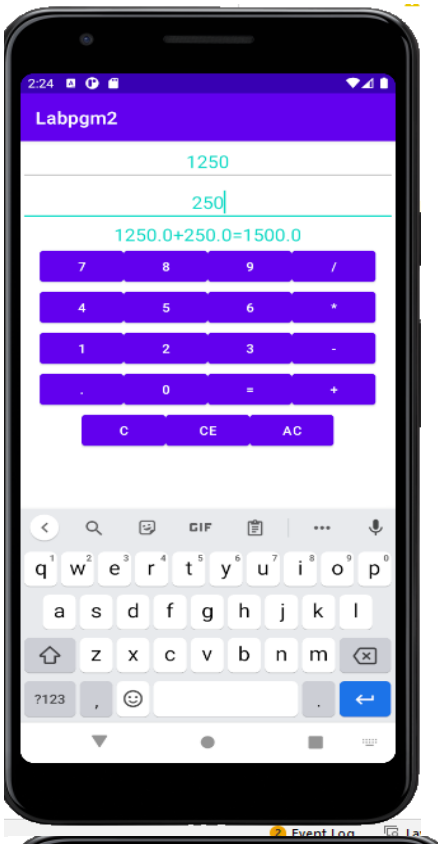
```
        else
        {
            Toast.makeText(this, "Please get the focus of First or second
number", Toast.LENGTH_LONG).show();
        }
    }

    public void five(View v)
    {
        if(number1.hasFocus())
        {
            number1.append("5");
        }
        else if(number2.hasFocus())
        {
            number2.append("5");
        }
        else
        {
            Toast.makeText(this, "Please get the focus of First or second
number", Toast.LENGTH_LONG).show();
        }
    }

    public void six(View v)
    {
        if(number1.hasFocus())
        {
            number1.append("6");
        }
        else if(number2.hasFocus())
        {
            number2.append("6");
        }
        else
        {
            Toast.makeText(this, "Please get the focus of First or second
number", Toast.LENGTH_LONG).show();
        }
    }

    public void seven(View v)
    {
        if(number1.hasFocus())
        {
            number1.append("7");
        }
        else if(number2.hasFocus())
        {
            number2.append("7");
        }
        else
        {
            Toast.makeText(this, "Please get the focus of First or second
number", Toast.LENGTH_LONG).show();
```

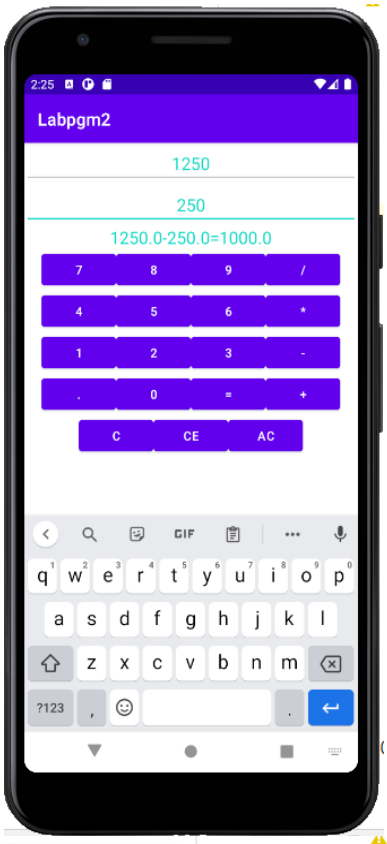
```
        }
    }

    public void eight(View v)
    {
        if(number1.hasFocus())
        {
            number1.append("8");
        }
        else if(number2.hasFocus())
        {
            number2.append("8");
        }
        else
        {
            Toast.makeText(this, "Please get the focus of First or second
number", Toast.LENGTH_LONG).show();
        }
    }

    public void nine(View v)
    {
        if(number1.hasFocus())
        {
            number1.append("9");
        }
        else if(number2.hasFocus())
        {
            number2.append("9");
        }
        else
        {
            Toast.makeText(this, "Please get the focus of First or second
number", Toast.LENGTH_LONG).show();
        }
    }

    public void zero(View v)
    {
        if(number1.hasFocus())
        {
            number1.append("0");
        }
        else if(number2.hasFocus())
        {
            number2.append("0");
        }
        else
        {
            Toast.makeText(this, "Please get the focus of First or second
number", Toast.LENGTH_LONG).show();
        }
    }

    public void dot(View v)
```

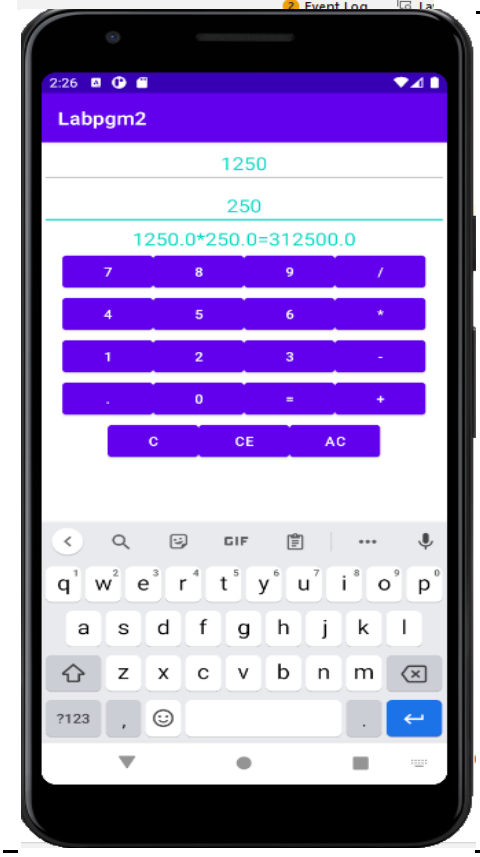
```
    {
        if(number1.hasFocus())
        {
            number1.append(".");
        }
        else if(number2.hasFocus())
        {
            number2.append(".");
        }
        else
        {
            Toast.makeText(this, "Please get the focus of First or second
number", Toast.LENGTH_LONG).show();
        }
    }
    public void add(View v)
    {
        op='+';
    }
    public void sub(View v)
    {
        op='-';
    }
    public void multiply(View v)
    {
        op='*';
    }
    public void divide(View v)
    {
        op='/';
    }
    public void compute(View v)
    {
        float n1,n2,r;
        switch(op)
        {
            case '+':   n1=Float.parseFloat(number1.getText().toString());
                        n2=Float.parseFloat(number2.getText().toString());
                        r=n1+n2;
                        res.setText(""+n1+"+"+n2+"="+r);
                        break;

            case '-':   n1=Float.parseFloat(number1.getText().toString());
                        n2=Float.parseFloat(number2.getText().toString());
                        r=n1-n2;
                res.setText(""+n1+"-"+n2+"="+r);
                        break;

            case '*':   n1=Float.parseFloat(number1.getText().toString());
                        n2=Float.parseFloat(number2.getText().toString());
                        r=n1*n2;
                res.setText(""+n1+"*"+n2+"="+r);
                        break;

            case '/':   n1=Float.parseFloat(number1.getText().toString());
```

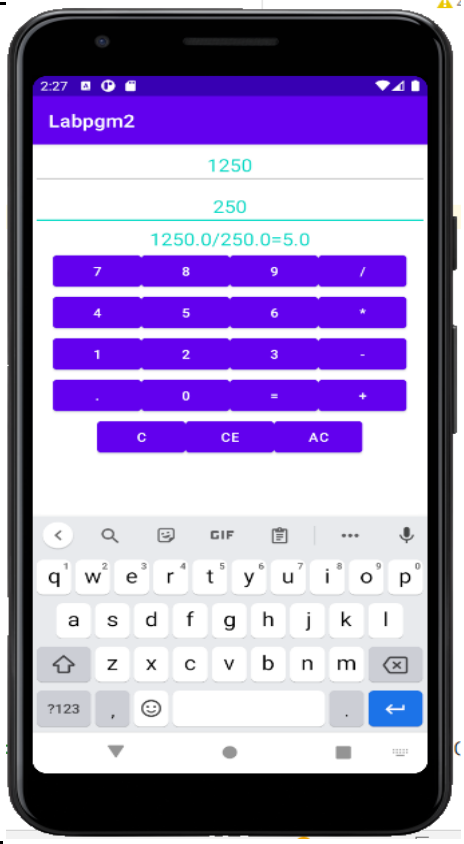
```
                        n2=Float.parseFloat(number2.getText().toString());
                        r=n1/n2;
                res.setText(""+n1+"/"+n2+"="+r);
                        break;
        }
    }

    public void all_clear(View v)
    {
        number1.setText("");
        number2.setText("");
        res.setText("");
    }

    public void clear_one_field_completly(View v)
    {
        if(number1.hasFocus())
        {
            number1.setText("");
            res.setText("");
        }
        else if (number2.hasFocus())
        {
            number2.setText("");
            res.setText("");
        }
        else
        {
            Toast.makeText(this, "Please click on Number1/Number2",
Toast.LENGTH_LONG).show();
        }
    }
    public void clear_one_digit_at_a_time(View v)
    {
        if(number1.hasFocus())
        {
            String n;
            n=number1.getText().toString();
            n=n.substring(0,n.length()-1);
            number1.setText(n);
        }
        else if(number2.hasFocus())
        {
            String n;
            n=number2.getText().toString();
            n=n.substring(0,n.length()-1);
            number2.setText(n);
        }
        else
        {
            Toast.makeText(this, "Please click on Number1/Number2",
Toast.LENGTH_LONG).show();
        }
    }
}
```

Output:


2:24
Labpgm2
1250
250
1250.0+250.0=1500.0
7 8 9 /
4 5 6 *
1 2 3 -
. 0 = +
C CE AC


2:25
Labpgm2
1250
250
1250.0-250.0=1000.0
7 8 9 /
4 5 6 *
1 2 3 -
. 0 = +
C CE AC

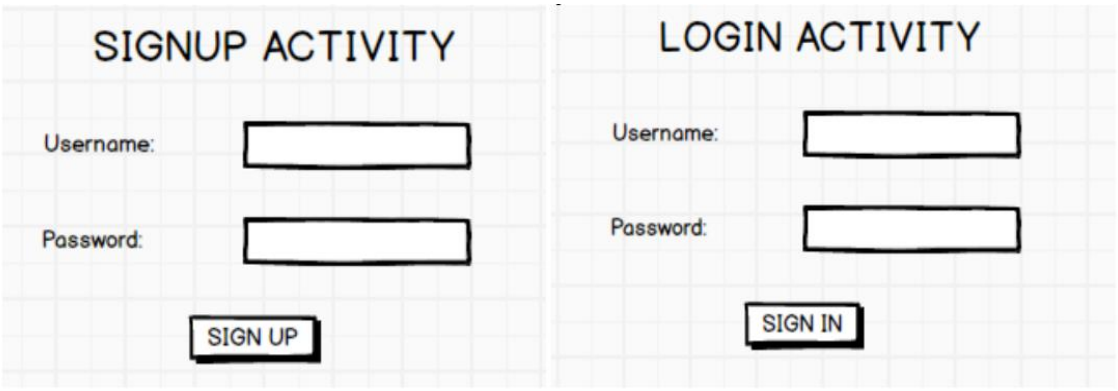

2:26
Labpgm2
1250
250
1250.0*250.0=312500.0
7 8 9 /
4 5 6 *
1 2 3 -
. 0 = +
C CE AC


2:27
Labpgm2
1250
250
1250.0/250.0=5.0
7 8 9 /
4 5 6 *
1 2 3 -
. 0 = +
C CE AC

Program No. 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.



Procedure:

1) Firstly Create an Application by Name "Labpgm3"
2) Go to xml code of design change the layout to "LinearLayout"
3) Add TextView component & change the following properties:

- Size: 30sp
- Text: "SIGNUP ACTIVITY"
- Center-Align

4) Add User ID (EditText) component & change the following properties in XML Code:

- Hint: "Enter User ID"
- id="@+id/uid"
- Center-Align
- Size: 20sp

5) Add Password (EditText) component & change the following properties in XML Code:

- Hint: "Enter Password"
- inputType: "textPassword"
- id="@+id/pwd"
- Center-Align
- Size: 20sp

6) Add Button component & change the following properties in XML

- Id: "@+id/signup"
- Text: "SIGN UP"
- onClick="signup"
- Center-Align

XML Code:

Activity_main.xml

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

    <TextView
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="SIGNUP ACTIVITY"
        android:textAlignment="center"
        android:textColor="@color/design_default_color_secondary_variant"
        android:textSize="30sp"          />

    <EditText
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:hint="Enter User ID"
        android:textSize="20sp"
        android:textAlignment="center"
        android:id="@+id/uid" />

    <EditText
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:hint="Enter Password"
        android:textSize="20sp"
        android:textAlignment="center"
        android:inputType="textPassword"
        android:id="@+id/pwd"        />

    <Button
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="SIGN UP"
        android:textAlignment="center"
        android:id="@+id/signup"
        android:onClick="signup"        />

</LinearLayout>
```

MainActivity.java

```
package com.example.labpgm3;
```

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

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

import java.util.regex.Matcher;
import java.util.regex.Pattern;

public class MainActivity extends AppCompatActivity {
    EditText username,passwd;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        username=findViewById(R.id.uid);
        passwd=findViewById(R.id.pwd);
    }

    public void signup(View v)
    {
        if(passwd.getText().toString().length()>=8 &&
validatepassword(passwd.getText().toString()))
        {
            Toast.makeText(this, "Sign Up Successful",
Toast.LENGTH_LONG).show();
            Intent i = new Intent(this,MainActivity2.class);
            Bundle b=new Bundle();
            b.putString("uid",username.getText().toString());
            b.putString("password",passwd.getText().toString());
            i.putExtras(b);
            startActivity(i);
        }
        else
        {
            Toast.makeText(this, "Password is not meeting the constraints",
Toast.LENGTH_LONG).show();
        }
    }

    public boolean validatepassword(String password)
    {
        Pattern ptrn;
        Matcher mat;
        String passwordptrn="^(?=.*[A-Z])(?=.*[a-z])(?=.*[0-
9])(?=.*[!@#$%^&*<>/+=-~])(?=\\S+$).{8,}$";
        ptrn=Pattern.compile(passwordptrn);
        mat=ptrn.matcher(password);
        return mat.matches();

    }
}
```

XML Code:

Activity_main2.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    tools:context=".MainActivity2">


    <TextView
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="SIGN IN ACTIVITY"
        android:textAlignment="center"
        android:textSize="30sp"
        android:textColor="@color/design_default_color_error"         />


    <EditText
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:id="@+id/uid"
        android:hint="Enter UID"
        android:textAlignment="center"
        android:textSize="20sp"         />

    <EditText
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:id="@+id/pwd"
        android:hint="Enter the Password"
        android:textAlignment="center"
        android:inputType="textPassword"
        android:textSize="20sp"         />

    <Button
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:id="@+id/signup"
        android:text="SIGN IN"
        android:textAlignment="center"
        android:onClick="signup"         />

</LinearLayout>
```

JAVA Code:

MainActivity2.java

```
package com.example.labpgm3;

import androidx.appcompat.app.AppCompatActivity;
import android.content.Intent;
import android.os.Bundle;
import android.view.View;
import android.widget.EditText;
import android.widget.Toast;

public class MainActivity2 extends AppCompatActivity {
    EditText username,password;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main2);
        username=findViewById(R.id.uid);
        password=findViewById(R.id.pwd);
    }
    public void signup(View v)
    {
        Bundle b=getIntent().getExtras();
        String uname=b.getString("uid");
        String passwd=b.getString("password");
        if(username.getText().toString().equals(uname) &&
password.getText().toString().equals(passwd))
        {
            Toast.makeText(this, "SIGN IN SUCCESSFUL",
Toast.LENGTH_LONG).show();
            Intent i =new Intent(this,Success.class);
            startActivity(i);
        }
        else
        {

            Toast.makeText(this, "SIGN IN UNSUCCESSFUL",
Toast.LENGTH_LONG).show();
        }
    }
}
```

XML Code:

Activity_success.xml

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

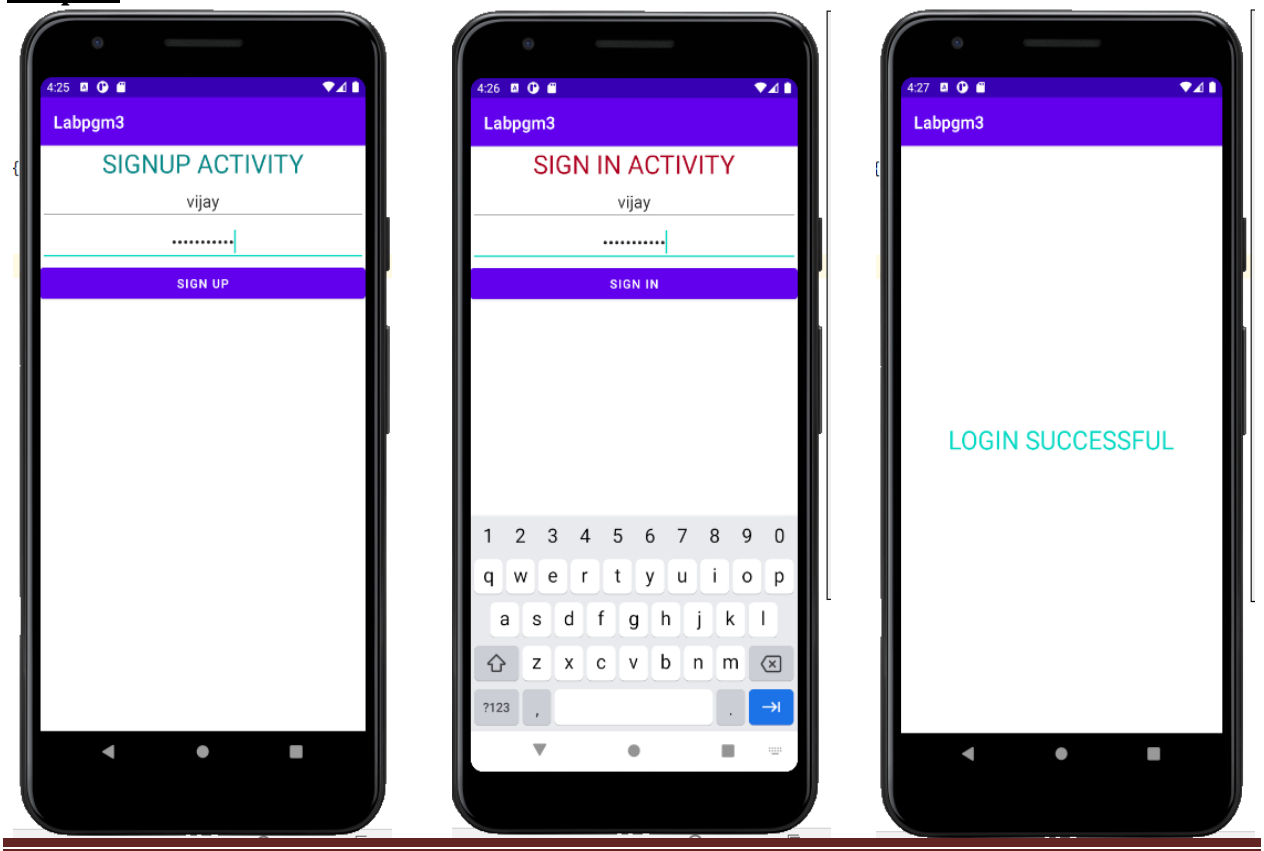
```
    android:layout_height="match_parent"
    tools:context=".Success">

    <TextView
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="LOGIN SUCCESSFUL"
        android:textSize="30sp"
        android:textAlignment="center"
        android:layout_gravity="center"
        android:textColor="@color/design_default_color_secondary"          />

</LinearLayout>
```

JAVA Code:

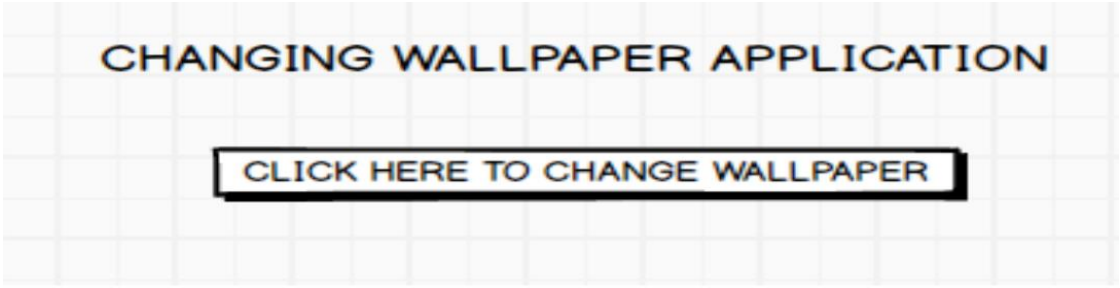
Success.java

```
package com.example.labpgm3;
import androidx.appcompat.app.AppCompatActivity;
import android.os.Bundle;
public class Success extends AppCompatActivity {

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

Output:



Program No. 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.



Procedure:

1) Firstly Create an Application by Name "Labpgm4"
2) Go to xml code of design change the layout to "LinearLayout" and `orientation="vertical"`
3) Add TextView component & change the following properties:
   - Size: 25sp
   - Text: "WALL PAPER CHANGING APP"
   - Center-Align

4) Add Button component & change the following properties:
   - android id: ="@+id/changewallpaper"
   - Text: "Change wall paper"
   - Onclick: "changewallpaper"

5) Save Ten 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:

Activity_main.xml

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

    <TextView
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="WALL PAPER CHANGING APP"
        android:textSize="25sp"
        android:textAlignment="center"
        android:textColor="@color/design_default_color_on_secondary"   />
```

```
    <Button
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:id="@+id/changewallpaper"
        android:text="Change wall paper"
        android:onClick="changewallpaper"          />

</LinearLayout>
```

JAVA Code:

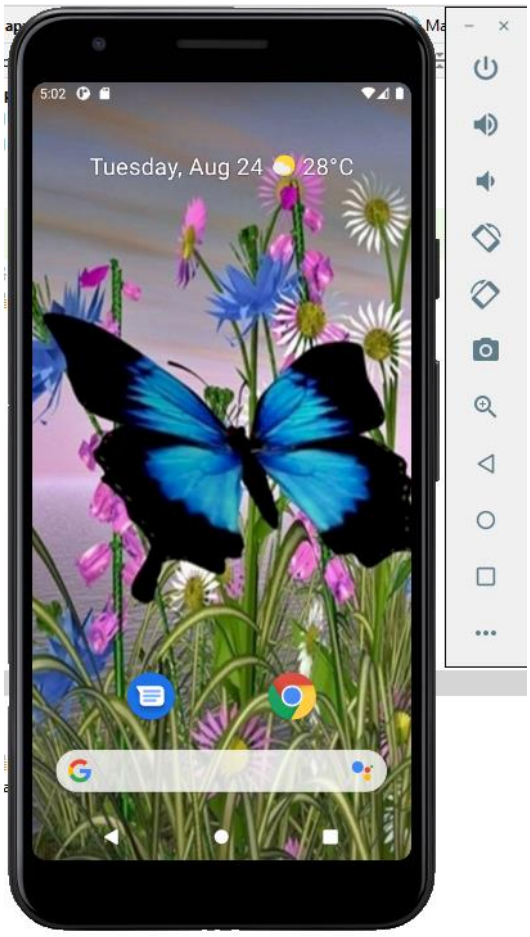
MainActivity.java

```
package com.example.labpgm4;

import androidx.appcompat.app.AppCompatActivity;

import android.app.WallpaperManager;
import android.graphics.Bitmap;
import android.graphics.drawable.BitmapDrawable;
import android.graphics.drawable.Drawable;
import android.os.Bundle;
import android.view.View;

import java.io.IOException;
import java.util.Timer;
import java.util.TimerTask;

public class MainActivity extends AppCompatActivity {
    Timer mytimer;
    Drawable drawable;
    WallpaperManager wpm;
    int next=1;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        mytimer=new Timer();
        wpm=WallpaperManager.getInstance(this);
    }

    public void changewallpaper(View v)
    {
        setwallpaper();
    }
    public void setwallpaper()
    {
        mytimer.schedule(new TimerTask() {
            @Override
            public void run() {
                if (next==1)
                {
```

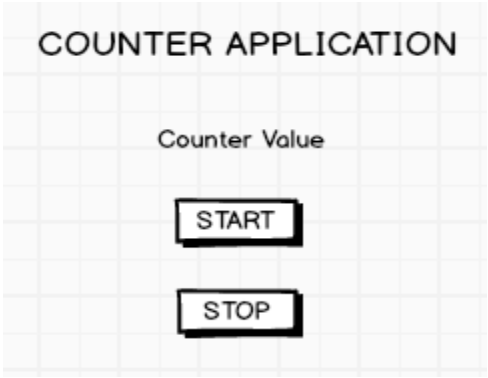
```
        drawable=getResources().getDrawable(R.drawable.one);
        next=2;
    }
    else if(next==2)
    {
        drawable=getResources().getDrawable(R.drawable.two);
        next=3;
    }
    else if(next==3)
    {
        drawable=getResources().getDrawable(R.drawable.three);
        next=4;
    }
    else if(next==4)
    {
        drawable=getResources().getDrawable(R.drawable.four);
        next=5;
    }
    else if(next==5)
    {
        drawable=getResources().getDrawable(R.drawable.five);
        next=6;
    }
    else if(next==6)
    {
        drawable=getResources().getDrawable(R.drawable.six);
        next=7;
    }
    else if(next==7)
    {
        drawable=getResources().getDrawable(R.drawable.seven);
        next=8;
    }
    else if(next==8)
    {
        drawable=getResources().getDrawable(R.drawable.eight);
        next=9;
    }
    else if(next==9)
    {
        drawable=getResources().getDrawable(R.drawable.nine);
        next=10;
    }
    else if(next==10)
    {
        drawable=getResources().getDrawable(R.drawable.ten);
        next=1;
    }

    Bitmap img=((BitmapDrawable)drawable).getBitmap();

    try {
        wpm.setBitmap(img);
    } catch (IOException e) {
        e.printStackTrace();
```

```
                    }

                }
            },30000,5000);
        }
}
```

Output:



Program No. 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.



Procedure:

1. Firstly Create an Application by Name "Labpgm5"
2. Go to xml code of design change the layout to "LinearLayout"
3. Add TextView component & change the following properties:
   - Size: 30sp
   - Text: "Counter Application"
   - Center-Align

4. Add TextView component & change the following properties:
   - Size: 20sp
   - Text: "Counter Application"
   - Center-Align
   - id: "@+id/counter"

5. Add Button components & change the following properties:
   - Text: Start
   - id: "@+id/Start"
   - Center-Align
   - android onclick "Startcounter"

6. Add Button components & change the following properties:
   - Text: Stop
   - id: "@+id/Stop"
   - Center-Align
   - android onclick "Stopcounter"

XML Code:

Activity_main.xml

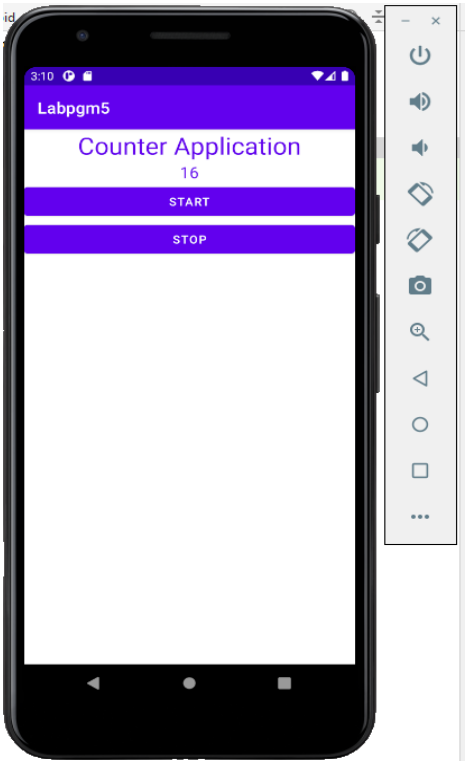
```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"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

    <TextView
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="Counter Application"
        android:textAlignment="center"
        android:textSize="30sp"
        android:textColor="@color/design_default_color_primary"    />

    <TextView
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:textSize="20sp"
        android:textAlignment="center"
        android:textColor="@color/design_default_color_primary"
        android:id="@+id/counter"   />

    <Button
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:id="@+id/Start"
        android:text="Start"
        android:textAlignment="center"
        android:onClick="Startcounter"   />

    <Button
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:id="@+id/Stop"
        android:text="Stop"
        android:textAlignment="center"
        android:onClick="Stopcounter"           />

</LinearLayout>
```

JAVA Code:

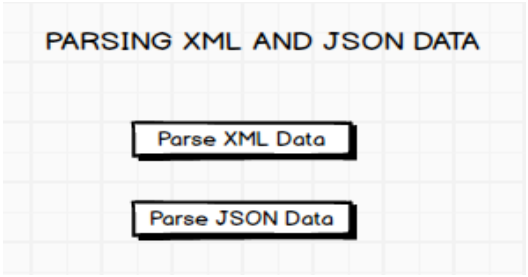
MainActivity.java

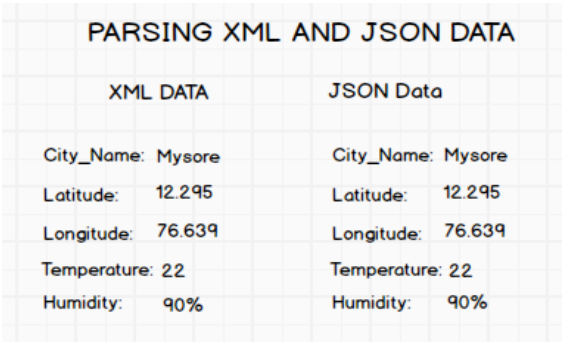
```
package com.example.labpgm5;
import androidx.appcompat.app.AppCompatActivity;

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

public class MainActivity extends AppCompatActivity {
    TextView tv;
    Handler myhandler=new Handler();
    int i=0;

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

    public void Startcounter(View v)
    {
        i=0;
        myhandler.postDelayed(Threadcount, 1000);

    }

    public void Stopcounter(View v)
    {
        myhandler.removeCallbacks(Threadcount);
    }

    public Runnable Threadcount=new Runnable()
    {

        @Override
        public void run() {
            tv.setText(""+i);
            i++;
            myhandler.postDelayed(Threadcount, 1000);
        }

    };
}
```

Output:



Program No. 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.





Procedure:

1. Firstly Create an Application by Name “Labpgm6”
2. Go to xml code of design change the layout to “LinearLayout”
3. Add TextView component & change the following properties:
   - Size: 30sp
   - Text: XML and JSON Parser
   - Center-Align
4. Add Two Buttons to Design & change the name “XMLParser” & “JSONParser” with following onclick functions:
   - XMLParser -Button: XMLParser
   - JSONParser -Button: JSONParser
5. Create a LinearLayout under the main layout & change the following properties:
6. Add TextView component & change the following properties:
   - Id: “@+id/resxml”
   - Size: 15sp
   - Align: Center
   - paddingLeft="10dp" & paddingRight="40dp"
7. Now change the view from Android view to Project View
8. Add Assets folder by following the given hierarchy:
   App->src->main->Create new Directory here with name “assets”
9. Inside the assets folder create new files of xml and json using the following hierarchy:
   - new->file->city.xml
   - new->file->city.json
10. once created place the following details inside the “city.xml” and “city.json”

"city.xml"

```
<?xml version="1.0" ?>
<records>

  <place>
    <cityname>mysore</cityname>
    <lat>96.95</lat>
    <long>76.3</long>
    <temp>22</temp>
    <humidity>90%</humidity>
  </place>

  <place>
    <cityname>Bangalore</cityname>
    <lat>96.98</lat>
    <long>76.73</long>
    <temp>20</temp>
    <humidity>80%</humidity>
  </place>
</records>
```

"city.json"

```
[
        {
          "name": "mysore",
          "lat": "96.95",
          "long": "76.63",
          "temp": "22",
          "humidity": "90%"
         },
        {
          "name": "Bangalore",
          "lat": "96.97",
          "long": "76.76",
          "temp": "20",
          "humidity": "80%"
        }
]
```

XML Code:

Activity_main.xml

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

    <TextView
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="XML and JSON Parser"
        android:textSize="30sp"
        android:textAlignment="center"

         />

    <Button
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:onClick="XMLParser"
        android:text="XMLParser"
        android:textAlignment="center"
        />

    <Button
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:onClick="JSONParser"
        android:text="JSONParser"
        android:textAlignment="center"
        />

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


        <TextView
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:id="@+id/resxml"
            android:textSize="15sp"
            android:textAlignment="center"
            android:paddingLeft="10dp"
            android:paddingRight="40dp"
            />
```

```
        <TextView
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:id="@+id/resjson"
            android:textSize="15sp"
            android:textAlignment="center"
            android:paddingLeft="40dp"

            />

    </LinearLayout>

</LinearLayout>
```

JAVA Code:

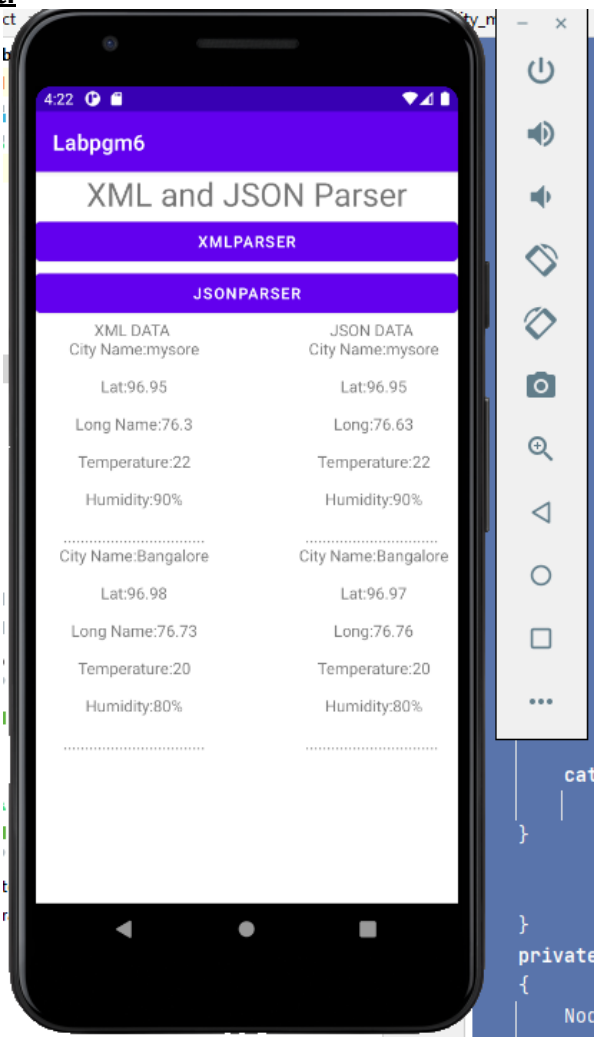
MainActivity.java

```
package com.example.labpgm6;

import androidx.appcompat.app.AppCompatActivity;

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

import org.json.JSONArray;
import org.json.JSONException;
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 org.xml.sax.SAXException;

import java.io.IOException;
import java.io.InputStream;

import javax.xml.parsers.DocumentBuilder;
import javax.xml.parsers.DocumentBuilderFactory;
import javax.xml.parsers.ParserConfigurationException;

public class MainActivity extends AppCompatActivity {
TextView resxml, resjson;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        resxml=findViewById(R.id.resxml);
        resjson=findViewById(R.id.resjson);
    }

    public void XMLParser(View v)
```

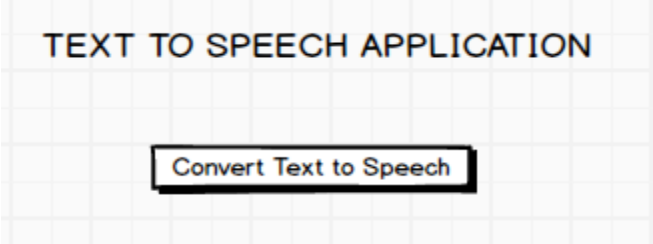
```
    {
        try {
            InputStream is=getAssets().open("city.xml");
            DocumentBuilderFactory
dbFactory=DocumentBuilderFactory.newInstance();
            DocumentBuilder dBuilder=dbFactory.newDocumentBuilder();
            Document doc=dBuilder.parse(is);
            Element element=doc.getDocumentElement();
            element.normalize();
            NodeList nList=doc.getElementsByTagName("place");
            resxml.setText("XML DATA");

            for(int i=0;i<nList.getLength();i++)
            {
                Node node=nList.item(i);
                if(node.getNodeType()==Node.ELEMENT_NODE)
                {
                    Element element2=(Element)node;
                    resxml.setText(resxml.getText()+"\n City
Name:"+getvalue("cityname",element2)+"\n");
                    resxml.setText(resxml.getText()+"\n
Lat:"+getvalue("lat",element2)+"\n");
                    resxml.setText(resxml.getText()+"\n Long
Name:"+getvalue("long",element2)+"\n");
                    resxml.setText(resxml.getText()+"\n
Temperature:"+getvalue("temp",element2)+"\n");
                    resxml.setText(resxml.getText()+"\n
Humidity:"+getvalue("humidity",element2)+"\n");
                    resxml.setText(resxml.getText()+"\n
.................................");
                }
            }
        }

        catch (IOException | ParserConfigurationException | SAXException e) {
            e.printStackTrace();
    }


    }
    private static String getvalue(String tag, Element element)
    {
        NodeList
nodeList=element.getElementsByTagName(tag).item(0).getChildNodes();
        Node node=nodeList.item(0);
        return node.getNodeValue();
    }

    public void JSONParser(View v) throws IOException, JSONException {
        String json;
        InputStream is=getAssets().open("city1.json");
        int size=is.available();
        byte[] buffer=new byte[size];
        is.read(buffer);
        is.close();
```

```
        json=new String(buffer, "UTF-8");
        JSONArray jsonArray=new JSONArray(json);
        resjson.setText("JSON DATA");

        for(int i=0; i<jsonArray.length(); i++)
        {
            JSONObject obj=jsonArray.getJSONObject(i);
            resjson.setText(resjson.getText()+"\n City
Name:"+obj.getString("name")+"\n");
            resjson.setText(resjson.getText()+"\n
Lat:"+obj.getString("lat")+"\n");
            resjson.setText(resjson.getText()+"\n
Long:"+obj.getString("long")+"\n");
            resjson.setText(resjson.getText()+"\n
Temperature:"+obj.getString("temp")+"\n");
            resjson.setText(resjson.getText()+"\n
Humidity:"+obj.getString("humidity")+"\n");

resjson.setText(resjson.getText()+"\n.............................");

        }
    }
}
```

Output:



Program No. 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.



Procedure:

1. Firstly Create an Application by Name "Labpgm7"
2. Go to xml code of design change the layout to "LinearLayout"
3. Add TextView component & change the following properties:
    - Size: 25sp
    - Text: Text To Speech
    - Center-Align
4. Add EditText component & change the following properties in XML Code:
    - Hint: "Enter pitch(1.0 is normal)"
    - id: "@+id/pitch"
    - Center-Align
    - textSize: 15sp

5. Add EditText component & change the following properties in XML Code:
    - Hint: "Enter Speech rate(1.0 is normal"
    - id: "@+id/speechrate"
    - Center-Align
    - textSize: 15sp

6. Add EditText component & change the following properties in XML Code:
    - Hint: "Enter text to speek"
    - id: "@+id/texttospeech"
    - Center-Align
    - textSize: 25sp

7. Add Button component & change the following properties in XML Code:
    - Text: "Text to speech"
    - id: "@+id/speek"
    - Center-Align
    - onclick "speak"

XML Code:

Activity_main.xml

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

    <TextView
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="Text To Speech"
        android:textAlignment="center"
        android:textColor="@color/material_on_background_emphasis_high_type"
        android:textSize="25sp" />

    <EditText
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:id="@+id/pitch"
        android:hint="Enter pitch(1.0 is normal)"
        android:textAlignment="center"
        android:textSize="15sp"
        />

    <EditText
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:id="@+id/speechrate"
        android:hint="Enter Speech rate(1.0 is normal"
        android:textAlignment="center"
        android:textSize="15sp"
        />

    <EditText
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:id="@+id/texttospeech"
        android:textSize="25sp"
        android:textAlignment="center"
        android:hint="Enter text to speek"
        />

    <Button
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:id="@+id/speek"
        android:text="Text to speech"
        android:textAlignment="center"
```

```
        android:onClick="speak"

        />

</LinearLayout>
```

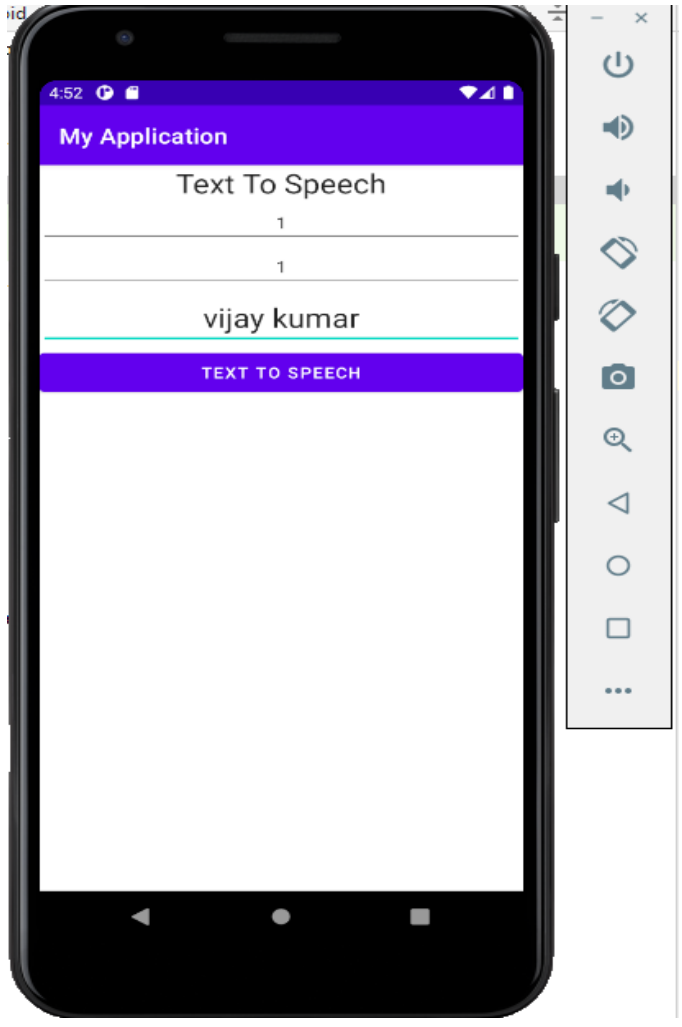
JAVA Code:

MainActivity.java

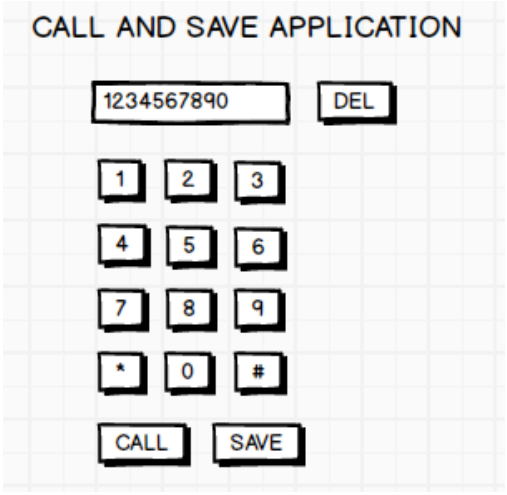
```
package com.example.myapplication;

import androidx.appcompat.app.AppCompatActivity;

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

import java.util.Locale;

public class MainActivity extends AppCompatActivity {
    EditText pitch,rate,texttospeech;
    Button speak;
    TextToSpeech t;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        pitch=findViewById(R.id.pitch);
        rate=findViewById(R.id.speechrate);
        texttospeech=findViewById(R.id.texttospeech);
        t=new TextToSpeech(this, new TextToSpeech.OnInitListener() {
            @Override

            public void onInit(int status) {
                if(status==TextToSpeech.SUCCESS)
                {
                    t.setLanguage(Locale.UK);
                    //t.setLanguage(Locale.ENGLISH );
                }
                else
                {
                    Toast.makeText(MainActivity.this, "text to Speech could
not be initialized", Toast.LENGTH_LONG).show();
                }
            }
        });
    }
```

```
    public void speak(View v)
    {
        String text=texttospeech.getText().toString();

        if(pitch.getText().toString().equals(""))
        {
            t.setPitch(1.0f);
        }
        else
        {
            t.setPitch(Float.parseFloat(pitch.getText().toString()));
        }
        if(rate.getText().toString().equals(""))
        {
            t.setSpeechRate(1.0f);
        }
        else
        {
            t.setSpeechRate(Float.parseFloat(rate.getText().toString()));
        }
        t.speak(text,TextToSpeech.QUEUE_FLUSH, null);
    }
}
```

Output:



Program No. 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.



Procedure:

1. Create an Application by name "Labpgm8"
2. Go to xml code of design change the layout to "LinearLayout"
3. Add TextView component & change the following properties:
   - Text: "CALL AND SAVE APPLICATION"
   - Size: "25sp"
   - Center-Align
4. Create a LinearLayout under the main layout & change the following properties:
   - assign width and height as "wrap_content"
   - Layout_gravity "center"
   - Android orientation "horizontal"
5. Add "EditText" component & change the following properties:
   - android id: "@+id/number"
   - android hint "Enter the Number to Dial/Save"
   - Center-Align
6. Add "Button" component & change the following properties:
   - android id: "@+id/delete"
   - android text "Del"
   - android onclick "delete"
7. Create a LinearLayout under the main layout & change the following properties:
   - assign width and height as "wrap_content"
   - Layout_gravity "center"
   - Android orientation "horizontal"
8. Add "Button" component & change the following properties:
   - android id: "@+id/one"
   - android text "1"
   - android onClick "one"

9. Repeat the step 8 for the numbers “2” and “3”:
10. Repeat the steps 7 and 8 for rest of the numbers and symbols:

XML Code:

Activity_main.xml

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

    <TextView
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="CALL AND SAVE APPLICATION"
        android:textSize="25sp"
        android:textColor="@color/black"
        android:textAlignment="center"/>

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

        <EditText
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:id="@+id/number"
            android:hint="Enter the Number to Dial/Save"
            android:textAlignment="center" />

        <Button
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="Del"
            android:id="@+id/delete"
            android:onClick="delete"  />


    </LinearLayout>

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

```
        <Button
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:id="@+id/one"
            android:text="1"
            android:onClick="one"   />

        <Button
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:id="@+id/two"
            android:text="2"
            android:onClick="two"               />
        <Button
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:id="@+id/three"
            android:text="3"
            android:onClick="three"              />

    </LinearLayout>

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

        <Button
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:id="@+id/four"
            android:text="4"
            android:onClick="four"              />

        <Button
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:id="@+id/five"
            android:text="5"
            android:onClick="five"              />
        <Button
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:id="@+id/six"
            android:text="6"
            android:onClick="six"              />

    </LinearLayout>

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

```
        <Button
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:id="@+id/seven"
            android:text="7"
            android:onClick="seven"             />

        <Button
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:id="@+id/eight"
            android:text="8"
            android:onClick="eight"             />

        <Button
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:id="@+id/nine"
            android:text="9"
            android:onClick="nine"             />

    </LinearLayout>

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

        <Button
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:id="@+id/star"
            android:text="*"
            android:onClick="star"             />

        <Button
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:id="@+id/zero"
            android:text="0"
            android:onClick="zero"             />
        <Button
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:id="@+id/hash"
            android:text="#"
            android:onClick="hash"             />

    </LinearLayout>

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

```
        android:orientation="horizontal"
        android:layout_gravity="center">

        <Button
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:id="@+id/call"
            android:text="CALL"
            android:onClick="call"              />

        <Button
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:id="@+id/save"
            android:text="SAVE"
            android:onClick="save"              />

    </LinearLayout>

</LinearLayout>
```

JAVA Code:

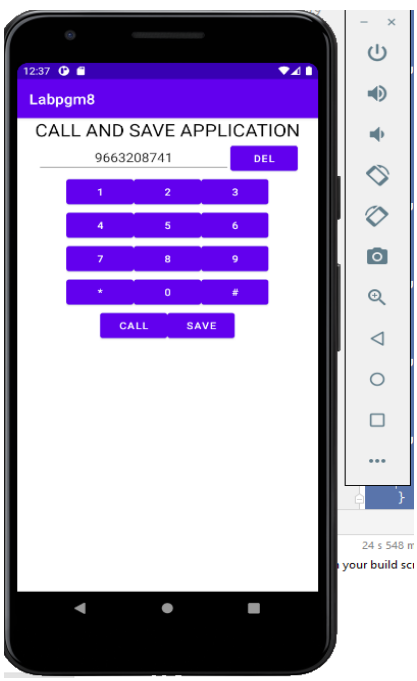
MainActivity.java

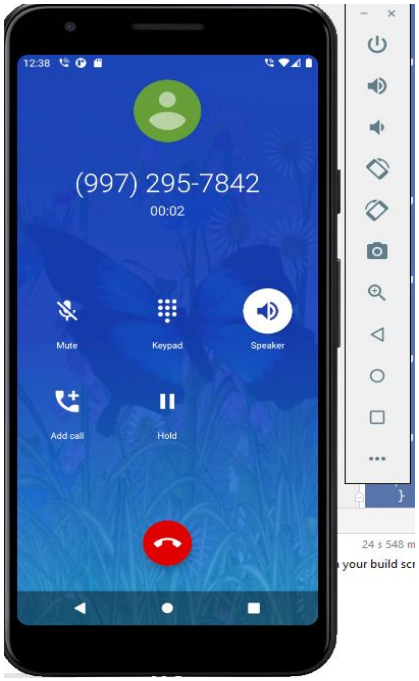
```
package com.example.labpgm8;

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.EditText;

public class MainActivity extends AppCompatActivity {
    EditText num;

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

    public void delete(View v)
    {
        String mynum=num.getText().toString();
        mynum=mynum.substring(0,mynum.length()-1);
        num.setText(mynum);
    }
```

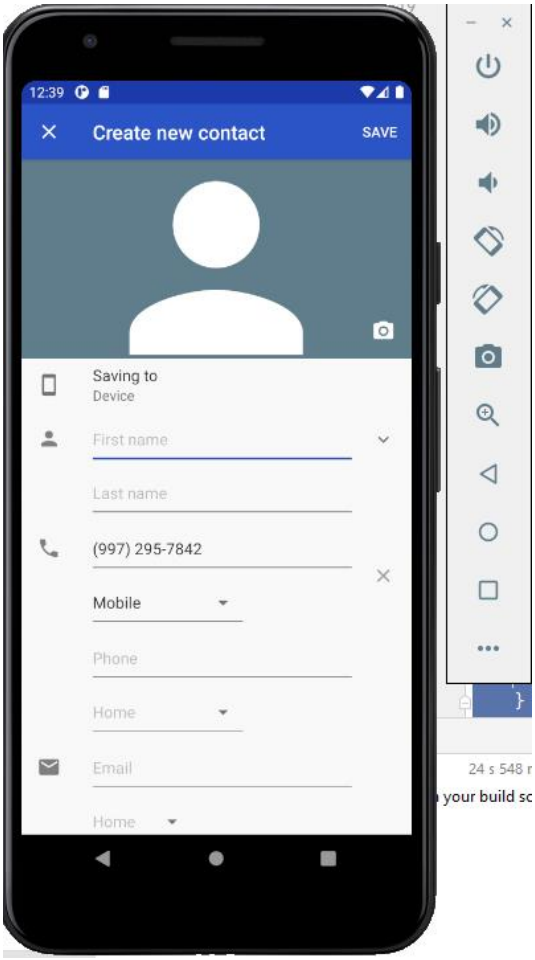
```
public void one(View v)
{
    num.append("1");
}
public void two(View v)
{
    num.append("2");
}
public void three(View v)
{
    num.append("3");
}
public void four(View v)
{
    num.append("4");
}
public void five(View v)
{
    num.append("5");
}
public void six(View v)
{
    num.append("6");
}
public void seven(View v)
{
    num.append("7");
}
public void eight(View v)
{
    num.append("8");
}
public void nine(View v)
{
    num.append("9");
}
public void star(View v)
{
    num.append("*");
}
public void zero(View v)
{
    num.append("0");
}
public void hash(View v)
{
    num.append("#");
}

public void call(View v)
{
    String mynum=num.getText().toString();
    Intent i=new Intent(Intent.ACTION_DIAL, Uri.parse("tel:"+mynum));
    startActivity(i);
```

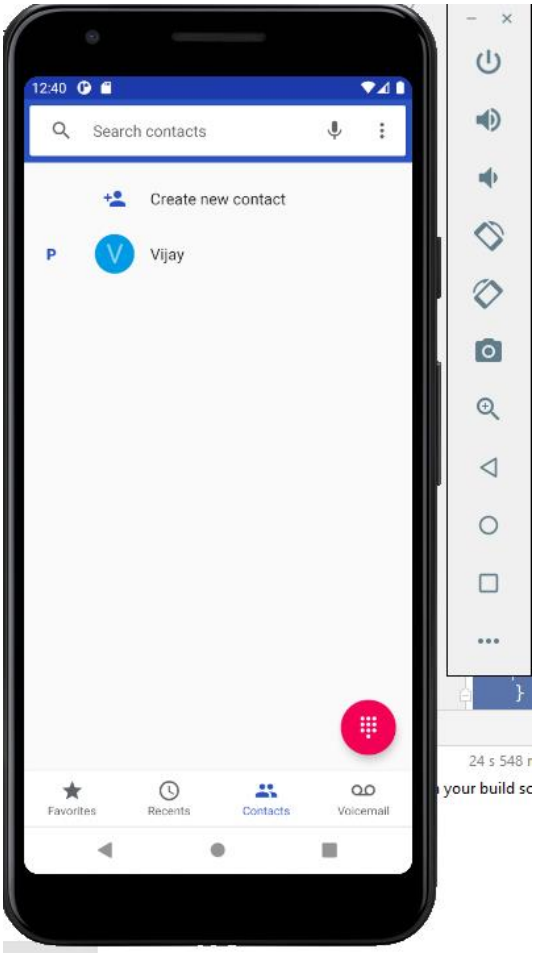
```
    }

    public void save(View v)
    {
        String mynum=num.getText().toString();
        Intent i=new Intent(Intent.ACTION_INSERT,
ContactsContract.Contacts.CONTENT_URI);
        i.putExtra(ContactsContract.Intents.Insert.PHONE,mynum);
        startActivity(i);
    }

}
```

Output:






12:39
Create new contact
SAVE
Saving to
Device
First name
Last name
(997) 295-7842
Mobile
Phone
Home
Email
Home


12:40
Search contacts
Create new contact
P
Vijay
Favorites
Recents
Contacts
Voicemail