Maharaja Education Trust® Mysore

MAHARAJA INSTITUTE OF TECHNOLOGY THANDAVAPURA

NH 766, Nanjangud Taluk, Mysuru - 571 302

(An ISO 9001:2015 and ISO 21001:2018 Certified Institution)

(Affiliated to VTU, Belagavi and approved by AICTE, New Delhi)

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

"MOBILE APPLICATION DEVELOPMENT LABORATORY MANUAL" (18CSMP68)

(2018 CBCS Scheme)

As per VTU Revised Syllabus for VI Semester CSE

Name: ____

USN: ____

Batch: _________ Sem: _________ Section: ____________

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

VISION OF THE DEPARTMENT

To impart quality education for producing world class engineers with latest knowledge and innovative ideas in Computer Science & Engineering to meet the expectations of industry and society and to produce globally competent graduates with moral values committed to build a vibrant nation.

MISSION OF THE DEPARTMENT

M 1: To promote technical proficiency by adopting effective teaching learning processes.

M 2: To pursue trending and emerging technologies in Computer Science and Engineering and learn their application across disciplines in order to serve the needs of industry, government, society, and the scientific community.

M 3: To provide environment & opportunity for students to bring out their inherent talents for their all round development.

M 4: To educate students to be successful, ethical, and effective problem solvers and lifelong learners who will contribute positively to the society.

M 5: To make computer science and engineering department a learning and agile centre to nurture the spirit of innovation, creativity and entrepreneurship among the students and faculty.

PROGRAMME SPECIFIC OUTCOMES (PSOs)

On completion of B.E Computer Science & Engineering Program, The graduates will be able to -

PSO 1 The ability to understand, analyse and develop computer programs in the areas related to algorithms, system software, multimedia, web design, big data analytics, and networking for efficient design of computer-based systems of varying complexity.

PSO 2 The ability to understand the evolutionary changes in computing, apply standard practices and strategies in software project development using open-ended programming environments to deliver a quality product for business success, real world problems and meet the challenges of the future.

PSO 3 The ability to employ modern computer languages, environments, and platforms in creating innovative career paths to be an entrepreneur, lifelong learning and a zest for higher studies and also to act as a good citizen by inculcating in them moral values & ethics.

PROGRAMME EDUCATIONAL OBJECTIVES (PEOs)

In support of the mission, within few years of graduation, the Computer Science & Engineering programme will enable its graduates to -

PEO 1 To be able to comprehend, understand and analyze Computer Science and Engineering problems and relate them with real life.

PEO 2 To provide in depth knowledge to design and develop novel products and innovative solution for real life problems in Computer Science and Engineering field and related domains.

PEO 3 To inculcate a conviction to believe in self, impart professional and ethical attitude, nurture to be an effective team member, infuse leadership qualities, build proficiency in soft skills and the abilities to relate engineering with the social issues.

PEO 4 To impart exhaustive knowledge of Computer Science & Engineering to take up key assignments in industry, undertake and excel in higher studies and Research & Development in computer science, related engineering fields and management.

PROGRAM OUTCOMES (POs)

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

PO 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.

PO 3: **Design/development of solutions**: Design solutions for complex engineering problems and 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.

PO 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 validconclusions.

PO 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.

PO 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 engineeringpractice.

PO 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 sustainabledevelopment.

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

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

PO 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.

PO 11: **Project management and finance**: Demonstrate knowledge and understanding of the engineering and management principles and apply these to one‟s own work, as a memberand leader in a team, to manage projects and in multidisciplinaryenvironments.

PO 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.

SYLLABUS

MOBILE APPLICATION DEVELOPMENT
B.E, VI Semester, Computer Science & Engineering
[As per Choice Based Credit System (CBCS) scheme]

Course Code	18CSMP68	CIE Marks	40
Number of Contact Hours/Week	03 Hours/Week	SEE Marks	60
RBT Levels	L1, L2, L3	Exam Hours	03
Credits – 02			

Laboratory Objectives: This Laboratory – 18CSMP68, will enable students to

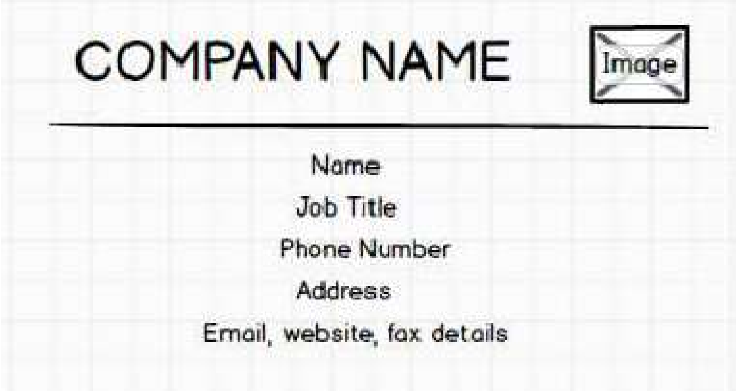
CO 1: To understand, Learn and acquire the art of Android Programming.
CO 2: To understand Configuring Android studio to run the applications.
CO 3: To understand and implement Android's User interface functions.
CO 4: To Create, modify and query on SQlite database.
CO 5: To Inspect different methods of sharing data using services.

Descriptions (if any)

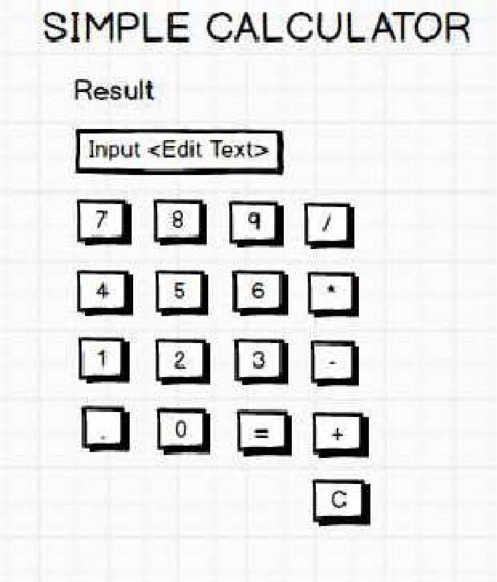
1. The installation procedure of the Android Studio/Java software must be demonstrated and carried out in groups.
2. Students should use the latest version of Android Studio/Java/Kotlin to execute these programs. Diagrams given are for representational purpose only, students are expected to improvise 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)**

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.

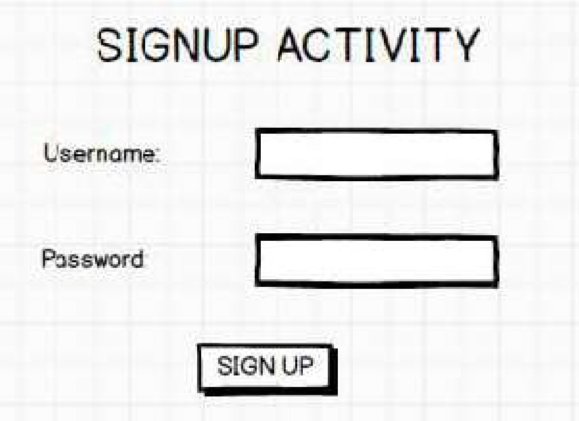


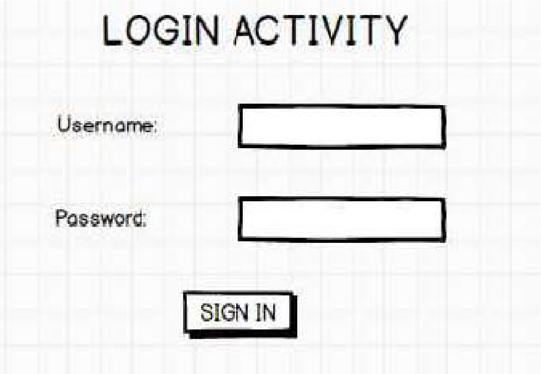
2. Develop an Android application using controls like Button, TextView, EditText for designing a calculator having basic functionality like Addition, Subtraction, Multiplication and Division.



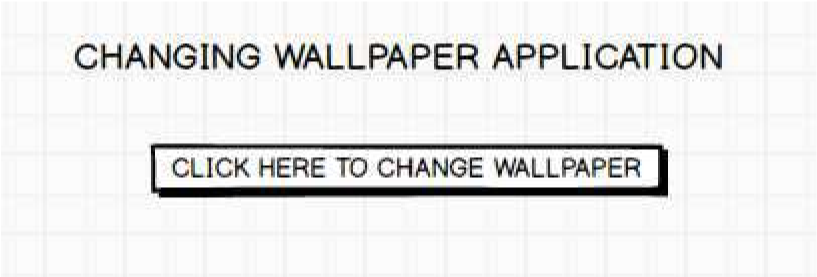
3. Create a SIGN Up activity with Username and Password. Validation of password should happen based on the following rules:
    - Password should contain uppercase and lowercase letters.
    - Password should contain letters and numbers.
    - Password should contain special characters.
    - · Minimum 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.

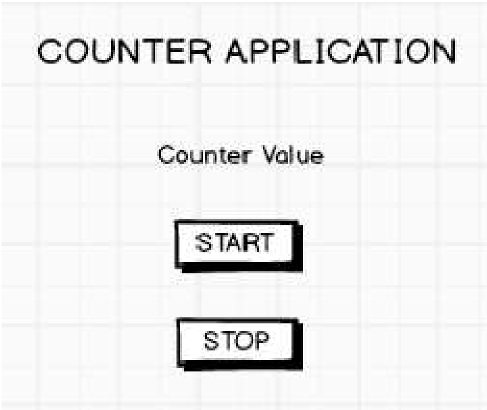




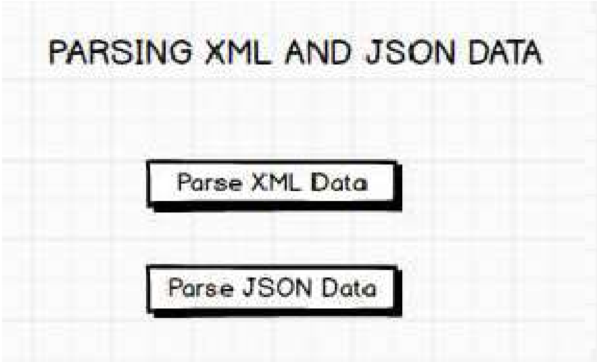
4. Develop an application to set an image as wallpaper. On click of a button, the wallpaper image should start to change randomly every 30 seconds.

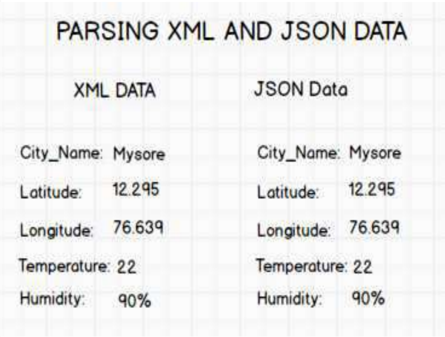

CHANGING WALLPAPER APPLICATION
CLICK HERE TO CHANGE WALLPAPER

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.

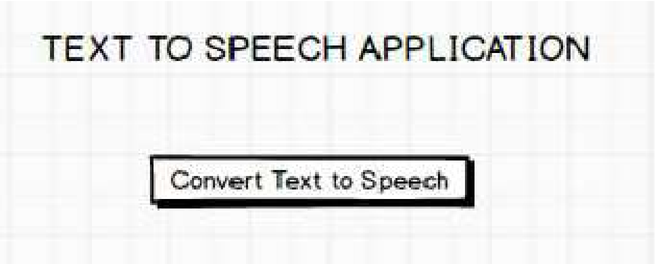




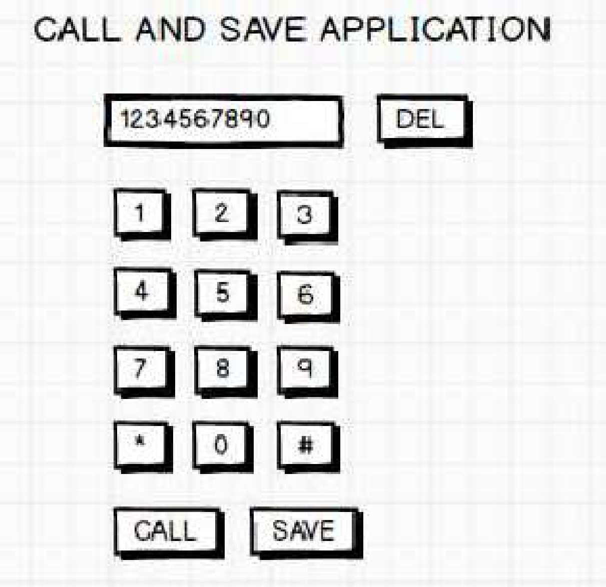
PARSING XML AND JSON DATA

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

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

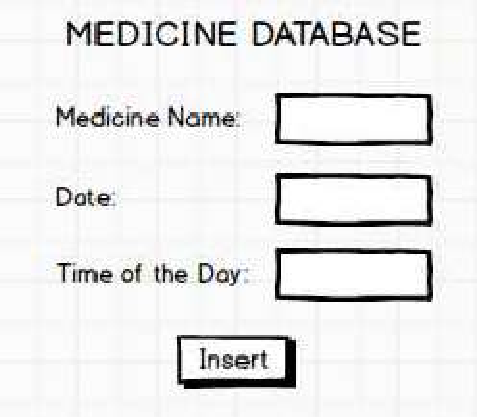


8. Create an activity like a phone dialer withCALLand SAVE buttons. On pressing the CALL button, it must call the phone number and on pressing the SAVE button it must save the number to the phone contacts.

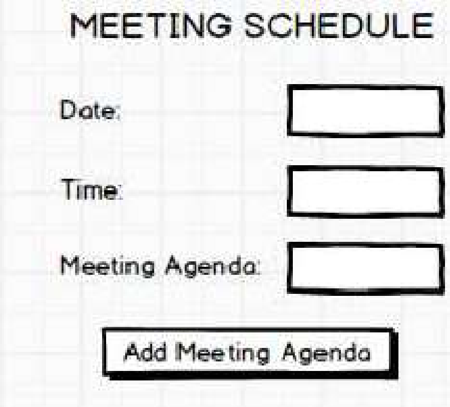


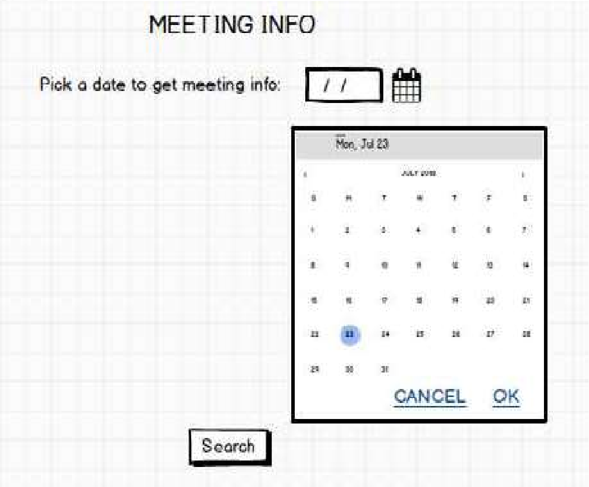
PART B

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

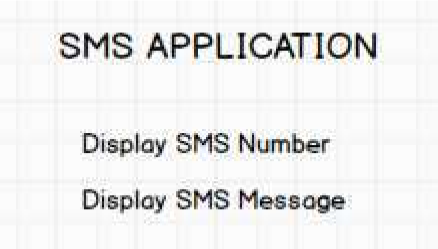


2. Develop a content provider application with an activity called “Meeting Schedule” which takes Date, Time and Meeting Agenda as input from the user and store this information into the SQLite database. Create another application with an activity called “Meeting Info” having DatePicker control, which on the selection of a date should display the Meeting Agenda information for that particular date, else it should display a toast message saying “No Meeting on this Date”.

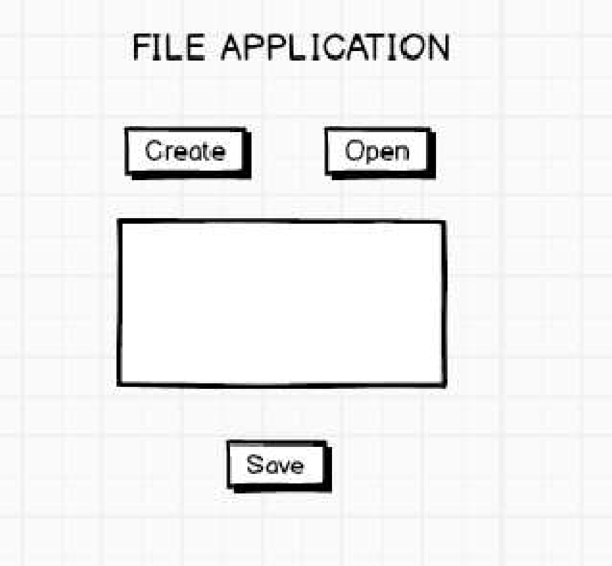




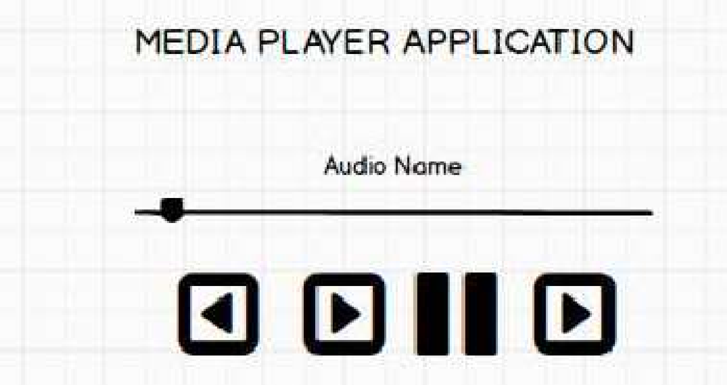
3. Create an application to receive an incoming SMS which is notified to the user. On clicking this SMS notification, the message content and the number should be displayed on the screen. Use appropriate emulator control to send the SMS message to your application.



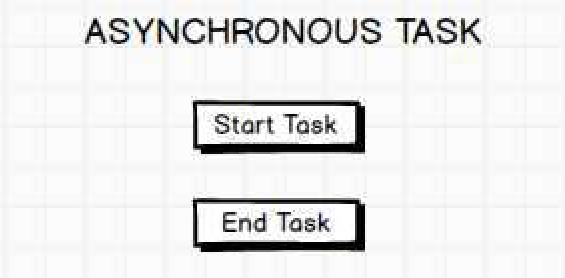
4. Write a program to create an activity having a Text box, and also Save, Open and Create buttons. The user has to write some text in the Text box. On pressing the Create button the text should be saved as a text file in MkSDcard. On subsequent changes to the text, the Save button should be pressed to store the latest content to the same file. On pressing the Open button, it should display the contents from the previously stored files in the Text box. If the user tries to save the contents in the Textbox to a file without creating it, then a toast message has to be displayed saying “First Create a File”.



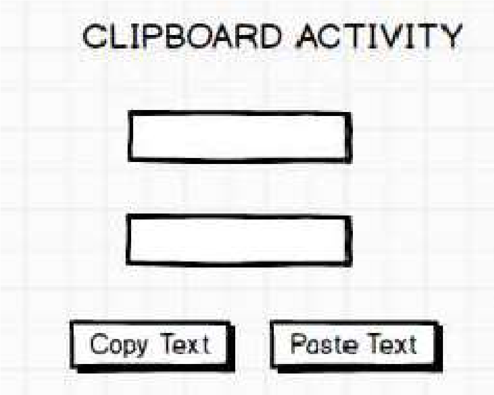
5. Create an application to demonstrate a basic media playerthat allows the user to Forward, Backward, Play and Pause an audio. Also, make use of the indicator in the seek bar to move the audio forward or backward as required.



6. Develop an application to demonstrate the use of Asynchronous tasks in android. The asynchronous task should implement the functionality of a simple moving banner. On pressing the **Start Task** button, the banner message should scrollfrom right to left. On pressing the **Stop Task** button, the banner message should stop.Let the banner message be “Demonstration of Asynchronous Task”.



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

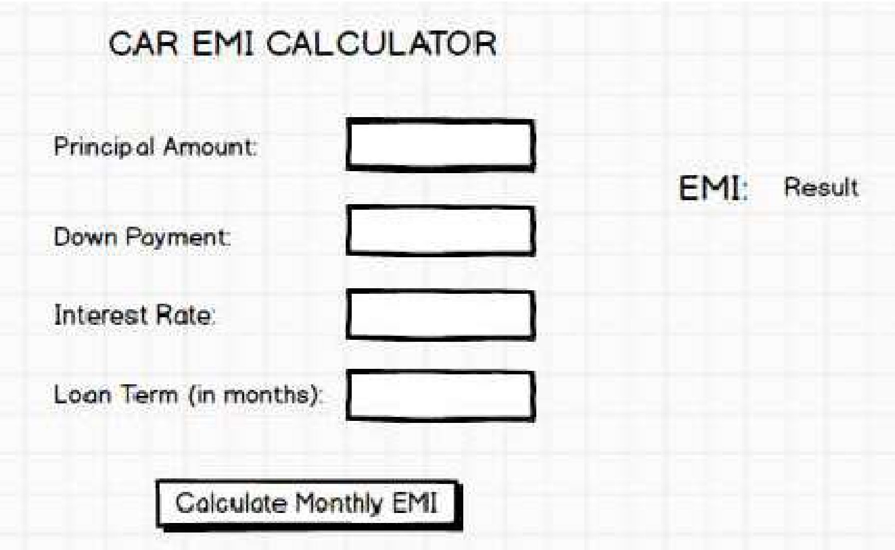


8. Create an AIDL service that calculates Car Loan EMI. The formula to calculate EMI is

$$\mathbf{E = P * (r(1+r)^n)/((1+r)^n-1)}$$

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

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



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

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

Procedure to Conduct Practical Examination

- Experiment distribution
  - For laboratories having only one part: Students are allowed to pick 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

Google Developer Training, **"Android Developer Fundamentals Course – Concept Reference",** Google Developer Training Team, 2017. https://www.gitbook.com/book/google-developer-training/android-developerfundamentals-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

CONTENTS

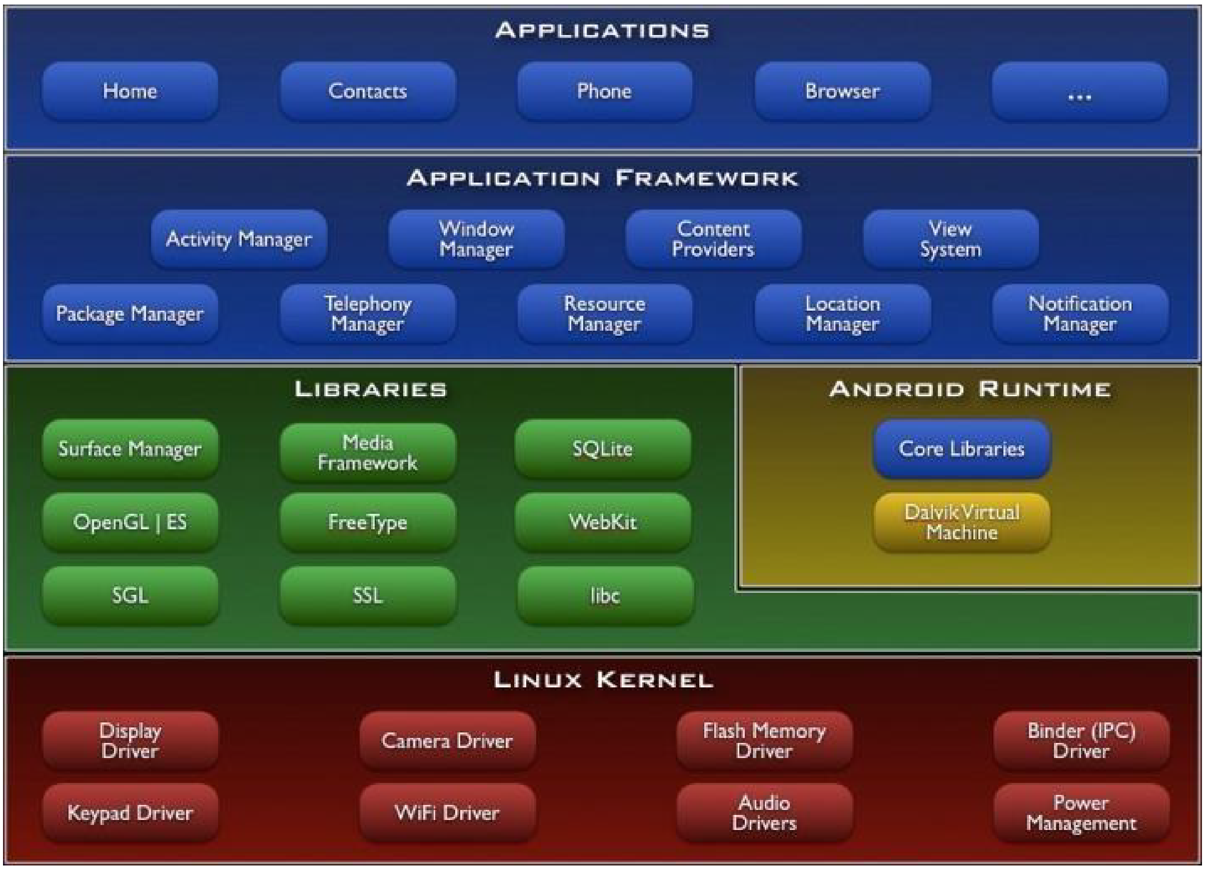
Chapter - 01

ANDROID

Android is a mobile operating system based on a modified version of the Linux kernel and other open source software, designed primarily for touch screen mobile devices such as smart phones and tablets. Android is developed by a consortium of developers known as the Open Handset Alliance, with the main contributor and commercial marketer being Google. Initially developed by Android Inc., which Google bought in 2005, Android was unveiled in 2007, with the first commercial Android device launched in September 2008. The current stable version is Android 11, released on September 8, 2020.

1.1 ANDROID ARCHITECTURE

Android operating system is a stack of software components which is roughly divided into five sections and four main layers as shown below in the architecture diagram.



1.2 LINUX KERNEL

At the bottom of the layers is Linux - Linux 2.6 with approximately 115 patches. This provides basic system functionality like process management, memory management, device management like camera, keypad, display etc. Also, the kernel handles all the things that Linux is really good at such as networking and a vast array of device drivers, which take the pain out of interfacing to peripheral hardware.

1.3 LIBRARIES

On top of Linux kernel there is a set of libraries including open -source Web browser engine WebKit, well known library libc, SQLite database which is a useful repository for storage and sharing of application data, libraries to play and record audio and video, SSL libraries responsible for Internet security etc.

1.4 ANDROID RUNTIME

This is the third section of the architecture and available on the second layer from the bottom. This section provides a key component called **Dalvik Virtual Machine** which is a kind of Java Virtual Machine specially designed and optimized for Android.

The Dalvik VM makes use of Linux core features like memory management and multi-threading, which is intrinsic in the Java language. The Dalvik VM enables every Android application to run in its own process, with its own instance of the Dalvik virtual machine.

The Android runtime also provides a set of core libraries which enable Android application developers to write Android applications using standard Java programming language.

1.5 APPLICATION FRAMEWORK

The Application Framework layer provides many higher-level services to applications in the form of Java classes. Application developers are allowed to make use of these services in their applications.

The Android runtime also provides a set of core libraries which enable Android application developers to write Android applications using standard Java programming language. The Application Framework layer provides many higher-level services to applications in the form of Java classes. Application developers are allowed to make use of these services in their applications.

1.6 APPLICATIONS

You will find all the Android application at the top layer. You will write your application to be installed on this layer only. Examples of such applications are Contacts Books, Browser, and Games etc.

1.7 ANDROID UI

An Android application user interface is everything that the user can see and interact with.

Chapter - 02

ANDROID STUDIO

Android Studio is the official Integrated Development Environment (IDE) for Android app development, based on IntelliJ IDEA . On top of IntelliJ's powerful code editor and developer tools, Android Studio offers even more features that enhance your productivity when building Android apps, such as:

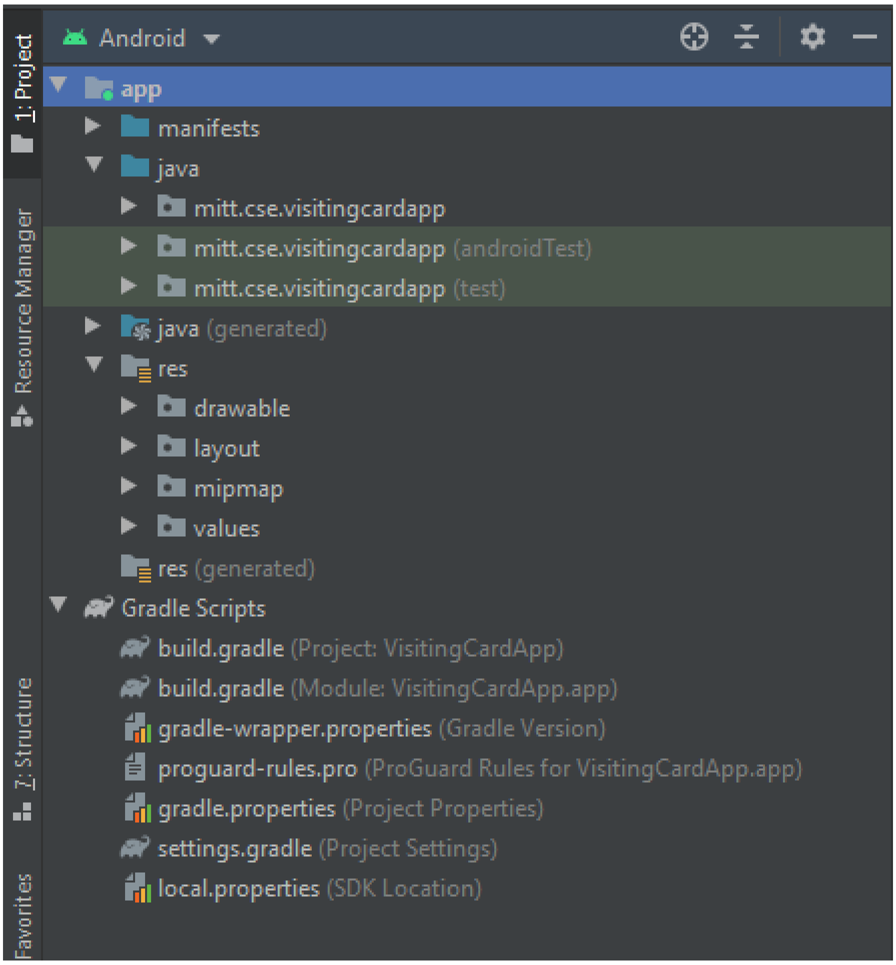
- A flexible Gradle-based build system
- A fast and feature-rich emulator
- A unified environment where you can develop for all Android devices
- Apply Changes to push code and resource changes to your running app without restarting your app
- Code templates and GitHub integration to help you build common app features and import sample code
- Extensive testing tools and frameworks
- Lint tools to catch performance, usability, version compatibility, and other problems
- C++ and NDK support
- Built-in support for Google Cloud Platform, making it easy to integrate Google Cloud Messaging and App Engine

2.1 ANDROID STUDIO INSTALLATION

To install Android Studio on Windows, proceed as follows:

1. If you downloaded an .exe file (recommended), double-click to launch it.
   If you downloaded a .zip file, unpack the ZIP, copy the **android-studio** folder into your **Program Files** folder, and then open the **android-studio > bin** folder and launch studio64.exe (for 64-bit machines) or studio.exe (for 32-bit machines).
2. Follow the setup wizard in Android Studio and install any SDK packages that it recommends.

2.2 PROJECT STRUCTURE



```
app
  manifests
  java
    mitt.cse.visitingcardapp
    mitt.cse.visitingcardapp (androidTest)
    mitt.cse.visitingcardapp (test)
  java (generated)
  res
    drawable
    layout
    mipmap
    values
  res (generated)
Gradle Scripts
  build.gradle (Project: VisitingCardApp)
  build.gradle (Module: VisitingCardApp.app)
  gradle-wrapper.properties (Gradle Version)
  proguard-rules.pro (ProGuard Rules for VisitingCardApp.app)
  gradle.properties (Project Properties)
  settings.gradle (Project Settings)
  local.properties (SDK Location)
```

Fig 2.1: Project Structure

Each project in Android Studio contains one or more modules with source code files and resource files. Types of modules include:

- **Android app modules**
- **Library modules**
- **Google App Engine modules**

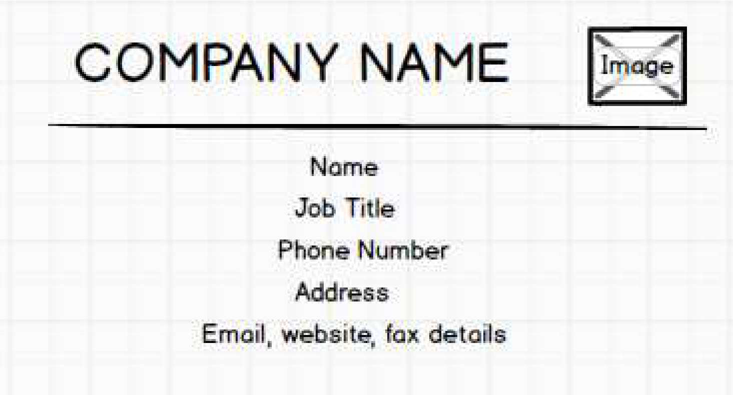
By default, Android Studio displays your project files in the Android project view, as shown in Fig 2.1. This view is organized by modules to provide quick access to your project's key source files.

All the build files are visible at the top level under **Gradle Scripts** and each app module contains the following folders:

- **manifests**: Contains the AndroidManifest.xml file.
- **java**: Contains the Java source code files, including JUnit test code.
- **res**: Contains all non-code resources, such as XML layouts, UI strings, and bitmap images.

The Android project structure on disk differs from this flattened representation. To see the actual file structure of the project, select **Project** from the **Project** dropdown (in Fig 2.1, it's showing as **Android**).

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



1. Firstly Create an Application by Name “VisitingCardApp”
2. Go to xml code of design change the layout to “RelativeLayout”
3. Add TextView component change the following properties:
    - Size: 38dp
    - Text: MITT
    - Align left top
4. Add ImageView to design and in type choose “IC_LAUNCHER_FOREGROUND”
    - Download the logo & copy the same in res->drawable folder
    - In xml code of imageview change srcCompat=”@drawable/logo”
    - Align right top
5. Add View component & change the following properties:
    - Height: 4dp
    - Background: “#4444” (black color)
6. Add TextView component change the following properties:
    - Size: 20dp
    - Text: N N Naveen
    - Style: Bold
    - Align center

7. Add TextView component change the following properties:
   - Size: 20dp
   - Text: Assistant Professor-CSE
   - Align center
8. Add TextView component change the following properties:
   - Size: 20dp
   - Text: Address-#6273, Sri Krishna Dhama, 2nd Phase, 4th Stage Vijaya Nagar, Mysuru-17
   - Align: center
9. Add TextView component change the following properties:
   - Size: 20dp
   - Text: Email-nnnaveen.nag@gmail.com
   - Align: center
10. Add TextView component change the following properties:
    - Size: 20dp
    - Text: Phone-9036915739

XML CODE

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

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

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

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

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

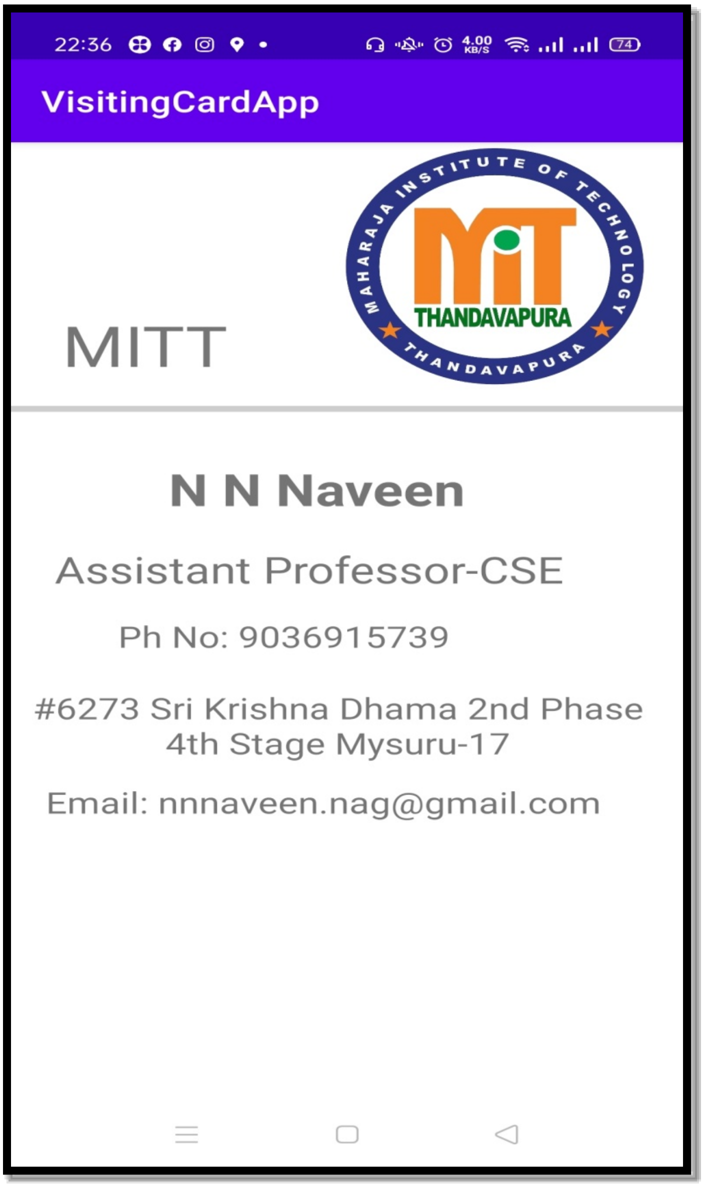
```
<TextView
    android:id="@+id/textView3"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_alignParentEnd="true"
    android:layout_alignParentRight="true"
    android:layout_alignParentBottom="true"
    android:layout_marginEnd="64dp"
    android:layout_marginRight="64dp"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

<TextView
    android:id="@+id/textView4"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_alignParentEnd="true"
    android:layout_alignParentRight="true"
    android:layout_alignParentBottom="true"
    android:layout_marginEnd="127dp"
    android:layout_marginRight="127dp"
    android:layout_marginBottom="294dp"
    android:text="Ph No: 9036915739"
    android:textSize="20dp" />

<TextView
    android:id="@+id/textView5"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_alignParentEnd="true"
    android:layout_alignParentRight="true"
    android:layout_alignParentBottom="true"
    android:layout_marginEnd="10dp"
    android:layout_marginRight="10dp"
    android:layout_marginBottom="229dp"
    android:text="#6273, Sri Krishna Dhama, 2nd Phase, 4th Stage
    Vijaya Nagar, Mysuru-17"
    android:textSize="20dp" />
```

```
<TextView
     android:id="@+id/textView6"
     android:layout_width="wrap_content"
     android:layout_height="wrap_content"
     android:layout_alignParentEnd="true"
     android:layout_alignParentRight="true"
     android:layout_alignParentBottom="true"
     android:layout_marginEnd="44dp"
     android:layout_marginRight="44dp"
     android:layout_marginBottom="189dp"
     android:text="Email: nnnaveen.nag@gmail.com"
     android:textSize="20dp" />

</RelativeLayout>
```

JAVA-CODE

```
import androidx.appcompat.app.AppCompatActivity;
import android.os.Bundle;
public class MainActivity extends AppCompatActivity {
     protected void onCreate(Bundle savedInstanceState) {
          super.onCreate(savedInstanceState);
          setContentView(R.layout.activity_main);
     }
 }
```

OUTPUT



VisitingCardApp

MITT

N N Naveen

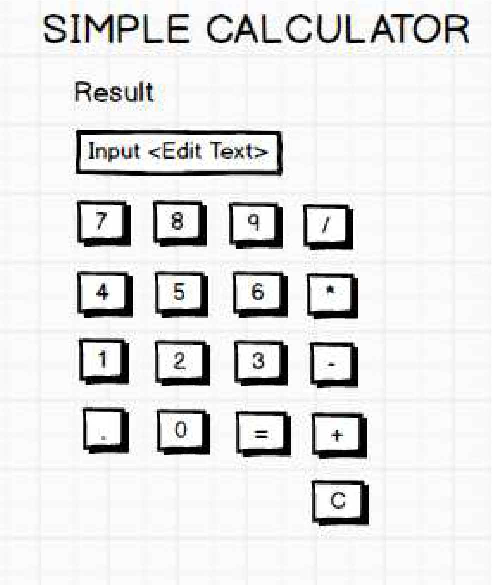
Assistant Professor-CSE

Ph No: 9036915739

#6273 Sri Krishna Dhama 2nd Phase
4th Stage Mysuru-17

Email: nnnaveen.nag@gmail.com

Program-2: Develop an Android application using controls like Button, TextView, EditText for designing a calculator having basic functionality like Addition, Subtraction, Multiplication, and Division.



1. Firstly Create an Application by Name “SimpleCalculator”
2. Go to xml code of design change the layout to “RelativeLayout”
3. Add TextView component & change the following properties:
    - Size: 38dp
    - Text: Simple Calci
    - Center-Align
4. Add PlainText(EditText) component & change the following properties in XML Code:
    - Text: “”
    - Hint: “Enter the first number”
    - id: “@+id/editText1”
5. Add PlainText(EditText) component & change the following properties in XML Code:
    - Text: “”
    - Hint: “Enter the second number”
    - id: “@+id/editText2”

6. Add TextView component to display result & change the following properties:
   - Size: 40dp
   - Text: “0”
   - Center-Align
   - id: “@+id/textView1”
7. Add 4 Buttons & rename the four buttons “Add”, “Sub”, ”Mul” and “Div” with following addition:
   - Onclick: “doAdd”(Add Button)
   - Onclick: “doSub”(Sub Button)
   - Onclick: “doMul”(Mul Button)
   - Onclick: “doDiv”(Div Button)

XML CODE

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

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignParentEnd="true"
        android:layout_alignParentBottom="true"
        android:layout_marginEnd="98dp"
        android:layout_marginBottom="653dp"
        android:text="SIMPLE CALCULATOR"
        android:textSize="32dp"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintHorizontal_bias="0.498"
        app:layout_constraintLeft_toLeftOf="parent"
        app:layout_constraintRight_toRightOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        app:layout_constraintVertical_bias="0.042" />
```

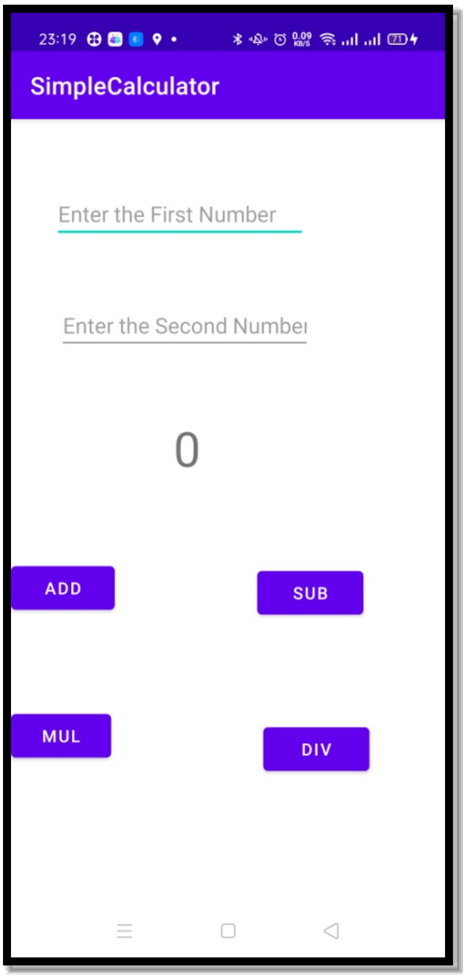
```
<EditText
    android:id="@+id/editText1"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_alignParentEnd="true"
    android:layout_alignParentBottom="true"
    android:layout_marginEnd="115dp"
    android:layout_marginBottom="547dp"
    android:ems="10"
    android:hint="Enter the First Number"
    android:inputType="textPersonName"
    android:text="" />

<EditText
    android:id="@+id/editText2"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_alignParentEnd="true"
    android:layout_alignParentBottom="true"
    android:layout_marginEnd="111dp"
    android:layout_marginBottom="455dp"
    android:ems="10"
    android:inputType="textPersonName"
    android:hint="Enter the Second Number"
    android:text="" />

<TextView
    android:id="@+id/textView1"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_alignParentEnd="true"
    android:layout_alignParentBottom="true"
    android:layout_marginEnd="203dp"
    android:layout_marginBottom="350dp"
    android:text="0"
    android:textSize="40dp" />

<Button
    android:id="@+id/button"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_alignParentEnd="true"
    android:layout_alignParentBottom="true"
```

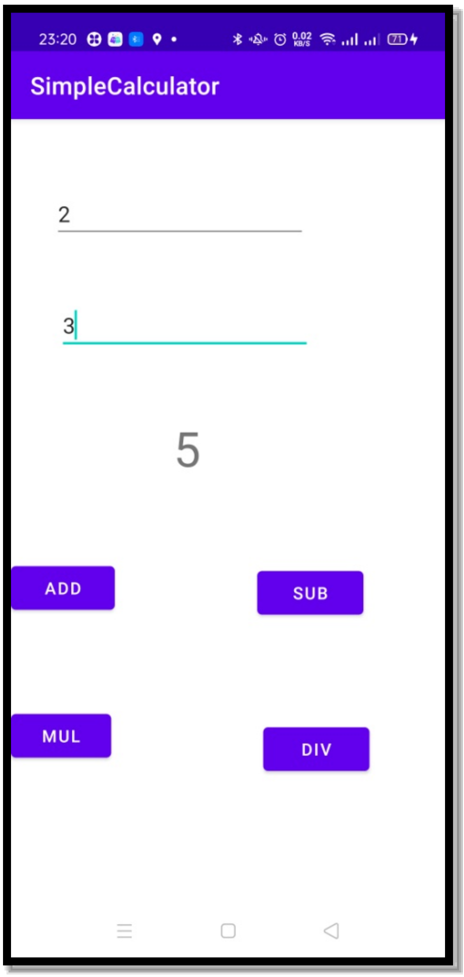
```
        android:layout_marginEnd="274dp"
        android:layout_marginBottom="237dp"
        android:onClick="doAdd"
        android:text="ADD" />

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

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

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

</RelativeLayout>
```

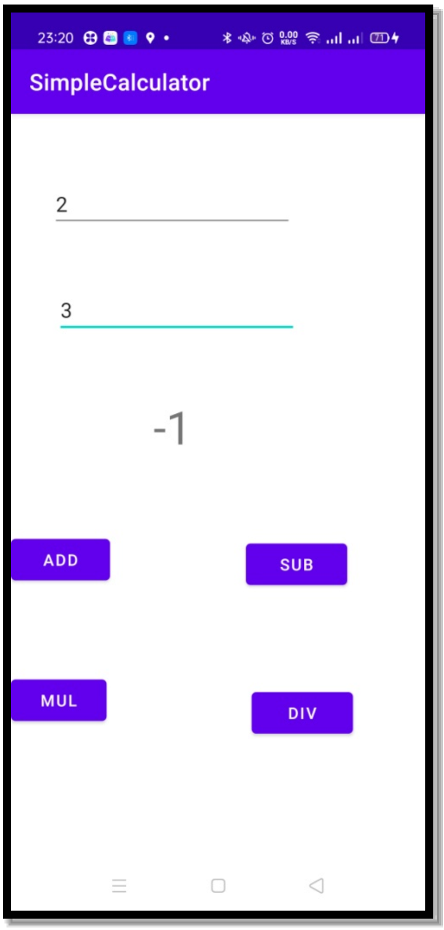
JAVA CODE

```
import androidx.appcompat.app.AppCompatActivity;
import android.os.Bundle;
import android.view.View;
import android.widget.EditText;
import android.widget.TextView;

public class MainActivity extends AppCompatActivity {
    EditText e1,e2;
    TextView tv1;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        e1 = findViewById(R.id.editText1);
        e2 = findViewById(R.id.editText2);
        tv1 = findViewById(R.id.textView1);
    }

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

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

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

OUTPUT



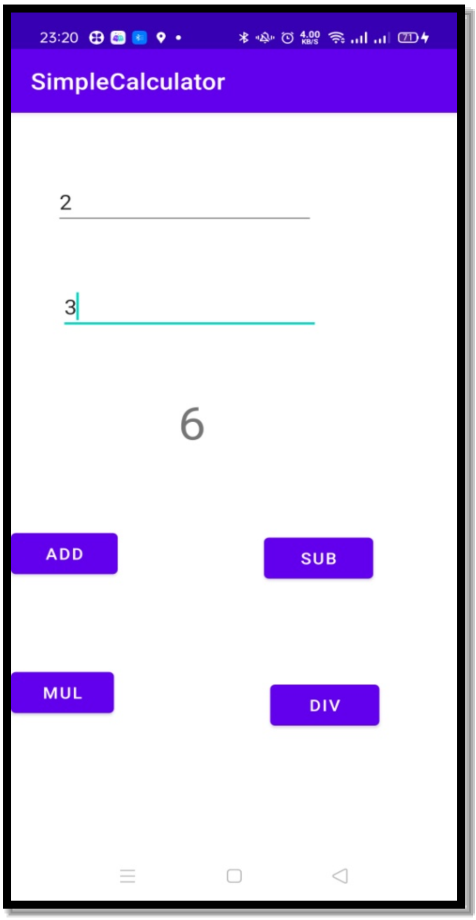



23:20
SimpleCalculator
ADD
SUB
MUL
DIV

2

3

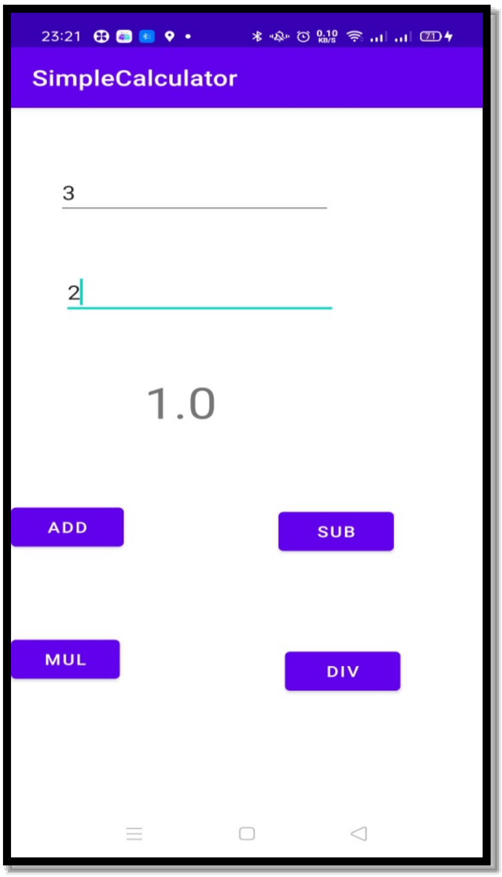
-1


23:20
SimpleCalculator
ADD
SUB
MUL
DIV

2

3

6


23:21
SimpleCalculator
ADD
SUB
MUL
DIV

3

2

1.0

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

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

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

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

XML CODE

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

    <TextView
        android:layout_width="160dp"
        android:layout_height="42dp"
        android:layout_alignParentEnd="true"
        android:layout_alignParentBottom="true"
        android:layout_marginEnd="112dp"
        android:layout_marginBottom="573dp"
        android:text="Sign Up"
        android:textSize="28dp"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintLeft_toLeftOf="parent"
        app:layout_constraintRight_toRightOf="parent"
        app:layout_constraintTop_toTopOf="parent" />

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

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

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

</RelativeLayout>
```

JAVA CODE

```
import androidx.appcompat.app.AppCompatActivity;
import android.content.Intent;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.Toast;
import java.util.regex.Pattern;
public class MainActivity extends AppCompatActivity {
    EditText emailEditText, passwordEditText;
    Button signUpBtn;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        emailEditText = findViewById(R.id.emailEditText);
        passwordEditText = findViewById(R.id.passwordEditText);
        signUpBtn = findViewById(R.id.signUpBtn);

    signUpBtn.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                String email = emailEditText.getText().toString();
                String password =
passwordEditText.getText().toString();
                if (!isValidPassword(password)) {
                    Toast.makeText(MainActivity.this, "Password Does
not match the rules",
                            Toast.LENGTH_LONG).show();
                    return;
                }
                Intent intent = new Intent(MainActivity.this,
LoginActivity.class);
                intent.putExtra("email", email);
                intent.putExtra("password", password);
                startActivity(intent);
            }
        });
    }
    Pattern lowercase = Pattern.compile("^.*[a-z].*$");
    Pattern uppercase = Pattern.compile("^.*[A-Z].*$");
    Pattern number = Pattern.compile("^.*[0-9].*$");
    Pattern specialCharacter = Pattern.compile("^.*[^a-zA-Z0-
9].*$");
    private Boolean isValidPassword(String password) {
        if (password.length() < 8) {
            return false;
        }
        if (!lowercase.matcher(password).matches()) {
            return false;
        }
        if (!uppercase.matcher(password).matches()) {
            return false;
        }
        if (!number.matcher(password).matches()) {
            return false;
        }
        if (!specialCharacter.matcher(password).matches()) {
            return false;
        }
        return true;
    }
}
```

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

<u>XML CODE</u>

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

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

```
 <EditText
        android:id="@+id/emailEditText"
        android:layout_width="363dp"
        android:layout_height="83dp"
        android:layout_alignParentEnd="true"
        android:layout_alignParentBottom="true"
        android:layout_marginEnd="27dp"
        android:layout_marginBottom="419dp"
        android:ems="10"
        android:hint="Email ID"
        android:inputType="textEmailAddress"
        android:textSize="28dp" />

    <EditText
        android:id="@+id/passwordEditText"
        android:layout_width="354dp"
        android:layout_height="wrap_content"
        android:layout_alignParentEnd="true"
        android:layout_alignParentBottom="true"
        android:layout_marginEnd="40dp"
        android:layout_marginBottom="299dp"
        android:ems="10"
        android:hint="Password"
        android:inputType="textPassword"
        android:textSize="28dp" />

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

</RelativeLayout>
```

JAVA CODE

```
import androidx.appcompat.app.AppCompatActivity;
import android.content.Intent;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.Toast;
public class LoginActivity extends AppCompatActivity {
    EditText emailEditText, passwordEditText;
    Button loginBtn;
    int counter=2;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_login);
        emailEditText=findViewById(R.id.emailEditText);
        passwordEditText=findViewById(R.id.passwordEditText);
        loginBtn=findViewById(R.id.loginBtn);

        String registeredEmail=getIntent().getStringExtra("email");
        String
registeredPassword=getIntent().getStringExtra("password");

        loginBtn.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                String email=emailEditText.getText().toString();
                String
password=passwordEditText.getText().toString();
                if(registeredEmail.equals(email)&&
registeredPassword.equals(password))
                {
                    Intent intent=new
Intent(LoginActivity.this,LoginSuccessful.class);
                    startActivity(intent);
                }
                else{

Toast.makeText(LoginActivity.this,"InvalidCredentials",Toast.LENGTH_
LONG).show();
                }
                counter--;
                if (counter==0)
                {
                    Toast.makeText(getBaseContext(),"FAILED LOGIN
ATTEMPTS",Toast.LENGTH_LONG).show();
                            loginBtn.setEnabled(false);
                }
            }
        });
    }
}
```

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

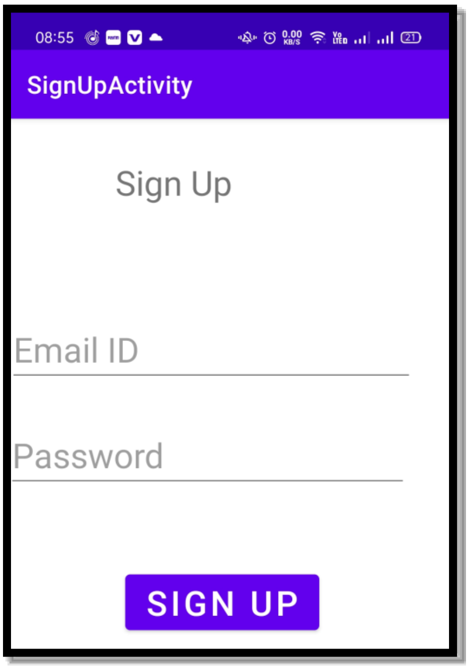
XML CODE

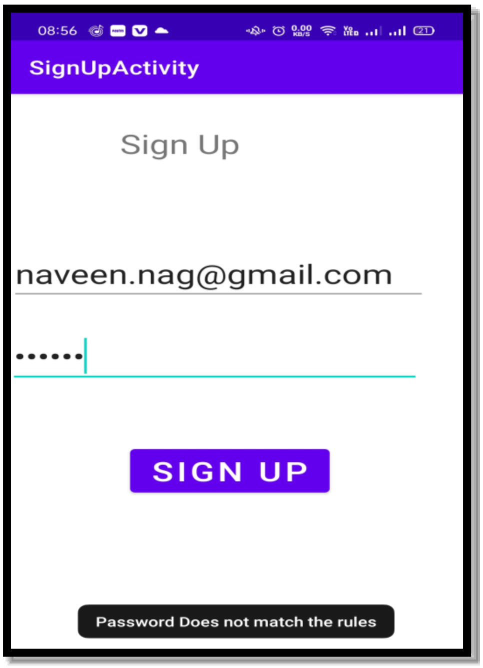
```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout
xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"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

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

</RelativeLayout>
```

JAVA CODE

```
import androidx.appcompat.app.AppCompatActivity;
import android.os.Bundle;

public class LoginSuccessful extends AppCompatActivity {
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_login_successful);
    }
}
```

OUTPUT

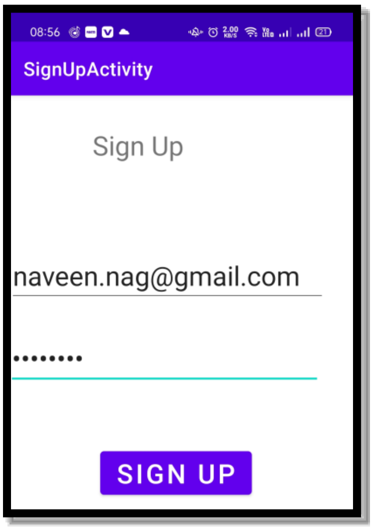




naveen.nag@gmail.com

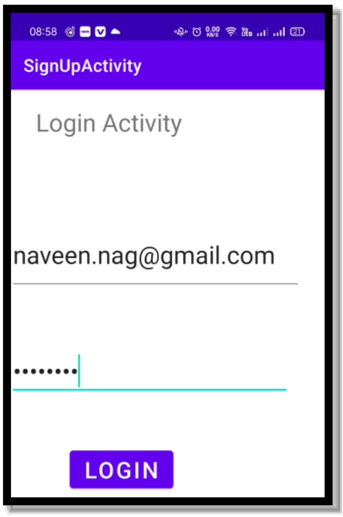
••••••

Password Does not match the rules



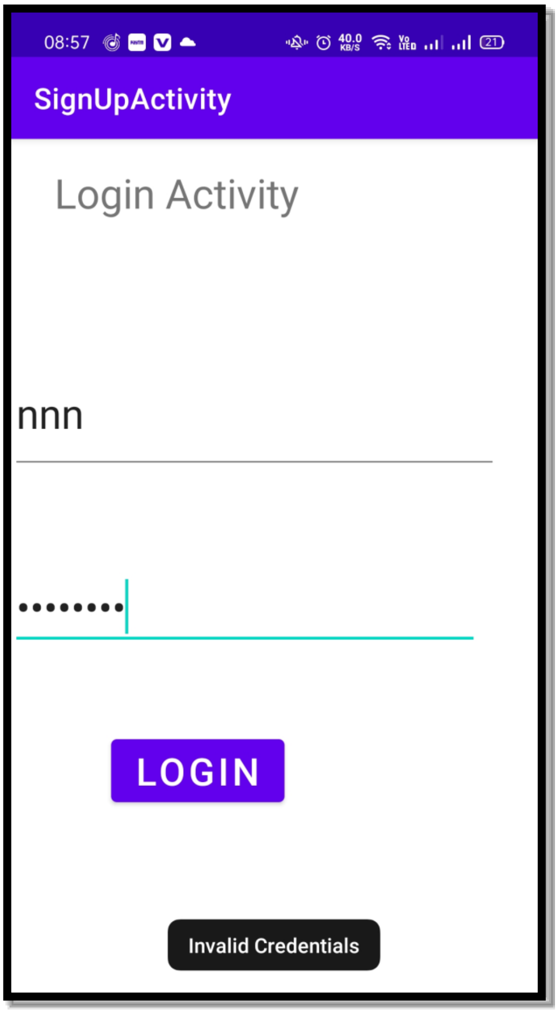
naveen.nag@gmail.com

••••••••

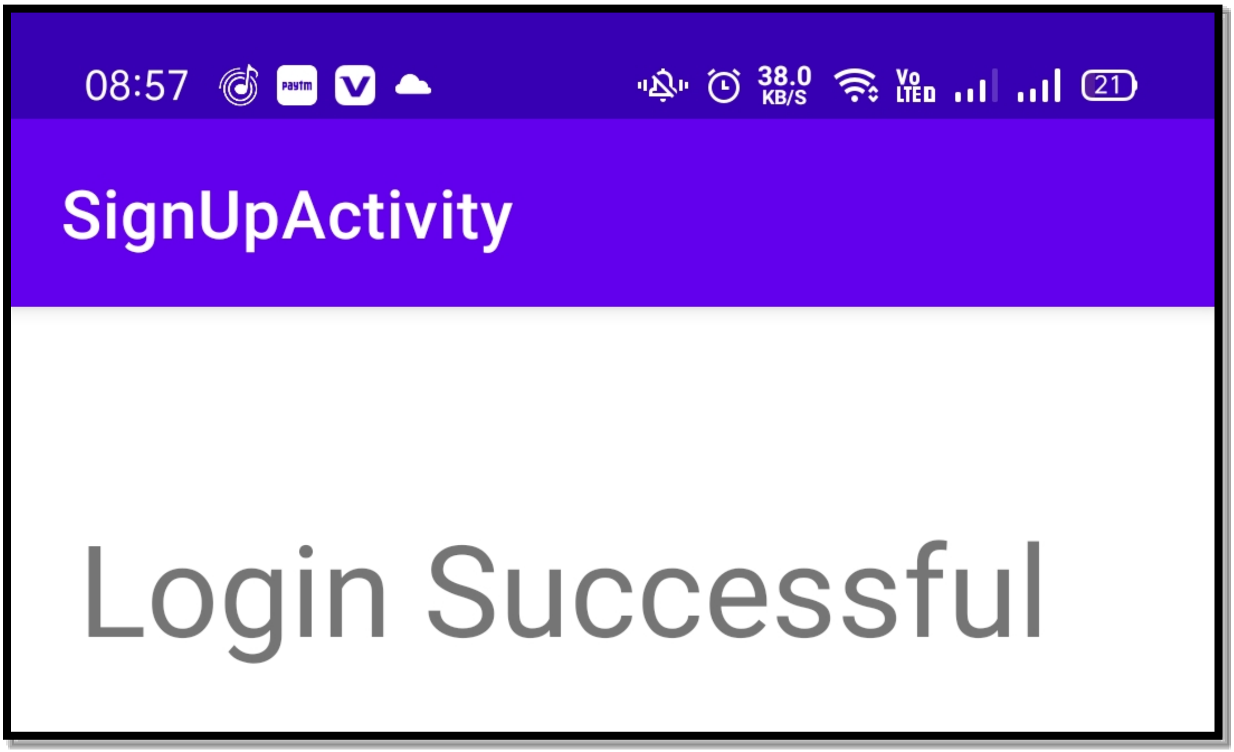


naveen.nag@gmail.com

••••••••



08:57
SignUpActivity
Login Activity
nnn
••••••••
LOGIN
Invalid Credentials



08:57
SignUpActivity
Login Successful

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

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

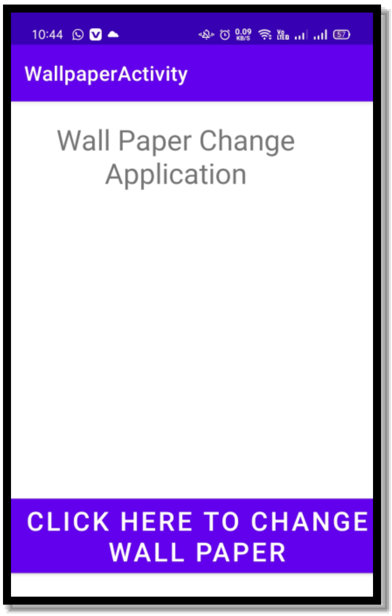
XML CODE

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

    <TextView
        android:id="@+id/textView"
        android:layout_width="243dp"
        android:layout_height="wrap_content"
        android:layout_alignParentEnd="true"
        android:layout_alignParentBottom="true"
        android:layout_marginEnd="73dp"
        android:layout_marginBottom="559dp"
        android:text="Wall Paper Change Application"
        android:textAlignment="center"
        android:textSize="28dp" />

    <Button
        android:id="@+id/button"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignParentEnd="true"
        android:layout_alignParentBottom="true"
        android:layout_marginEnd="-13dp"
        android:layout_marginBottom="177dp"
        android:text="Click Here To Change Wall Paper"
        android:textSize="26dp" />

</RelativeLayout>
```

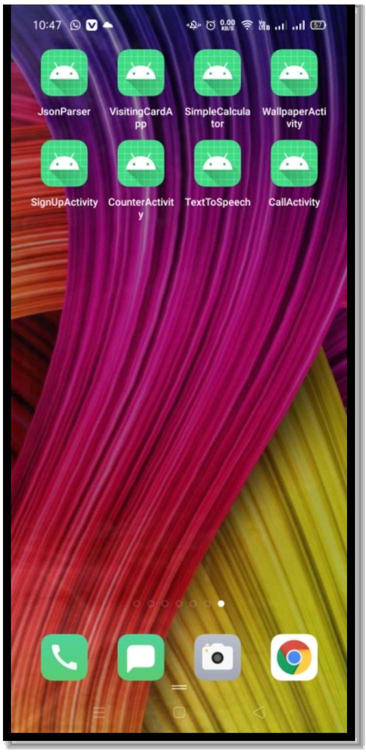
JAVA CODE

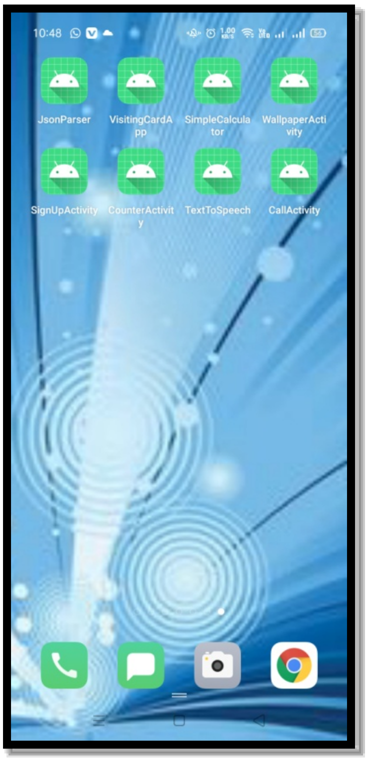
```
import androidx.appcompat.app.AppCompatActivity;
import android.app.WallpaperManager;
import android.graphics.Bitmap;
import android.graphics.BitmapFactory;
import android.graphics.drawable.AnimationDrawable;
import android.graphics.drawable.BitmapDrawable;
import android.graphics.drawable.Drawable;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.Toast;
import java.io.IOException;
import java.util.Timer;
import java.util.TimerTask;

public class MainActivity extends AppCompatActivity {
    Button changewallpaper;
    Timer mytimer;
    Drawable drawable;
    WallpaperManager wpm;
    int prev=1;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        mytimer = new Timer();
        wpm = WallpaperManager.getInstance(this);
        changewallpaper = findViewById(R.id.button);
        changewallpaper.setOnClickListener(new
View.OnClickListener() {
            @Override public void onClick(View view) {
                setWallpaper();
            }
        });
    }
    private void setWallpaper() {
        mytimer.schedule(new TimerTask() {
            @Override
            public void run() {
                if(prev==1) {
                    drawable =
getResources().getDrawable(R.drawable.one);
                    prev = 2;
                }
                else if(prev==2) {
                    drawable =
getResources().getDrawable(R.drawable.two);
                    prev=3;
                }
                else if(prev==3) {
                    drawable =
getResources().getDrawable(R.drawable.three);
                    prev=4;
                }
                else if(prev==4) {
                    drawable =
getResources().getDrawable(R.drawable.four);
                    prev=5;
                }
                else if(prev==5) {
                    drawable =
getResources().getDrawable(R.drawable.five);
                    prev=1;
                }
                Bitmap wallpaper =
((BitmapDrawable)drawable).getBitmap();
                try {
                    wpm.setBitmap(wallpaper);
                } catch (IOException e) {
                    e.printStackTrace();
                }
            }
        },0,30000); } }
```

OUTPUT

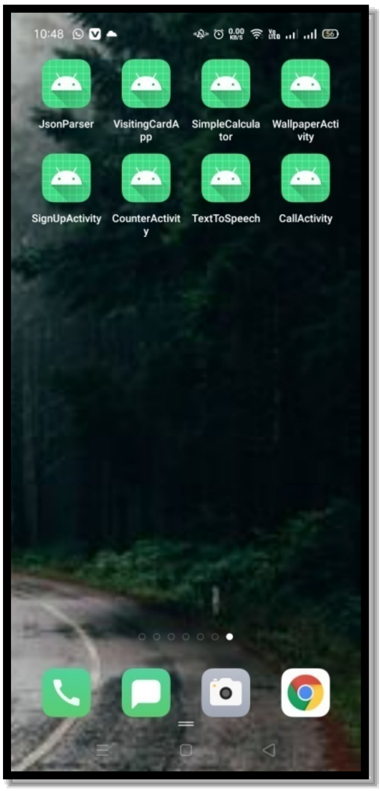


Wall Paper Change Application





-



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

1. Firstly Create an Application by Name "CounterActivity"
2. Go to xml code of design change the layout to "RelativeLayout"
3. Add TextView component & change the following properties:
   - Size: 38dp
   - Text: "Counter Application"
   - Center-Align
4. Add TextView component & change the following properties:
   - Text: "Counter Value"
5. Add Button components & change the following properties:
   - Size: 38dp
   - Text: Start
   - id: "@+id/btn_start"
6. Add Button components & change the following properties:
   - Size: 38dp
   - Text: Stop
   - id: "@+id/btn_stop"

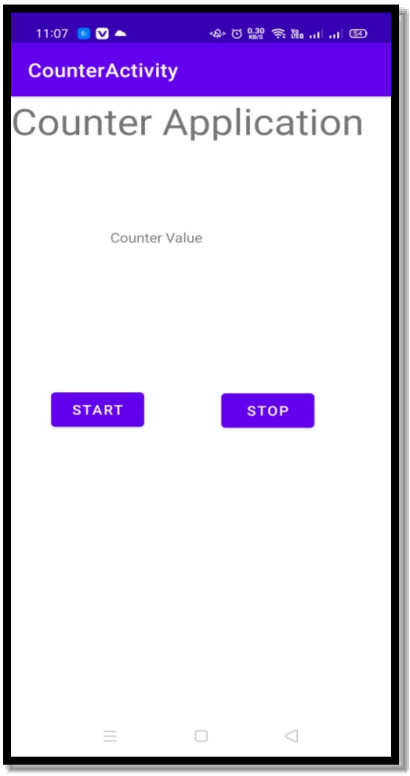
XML CODE

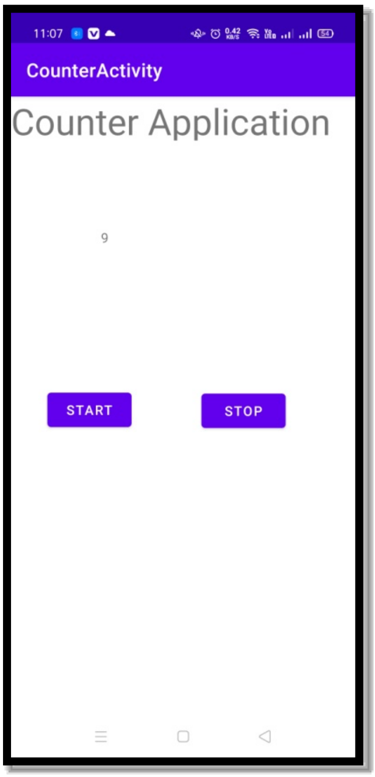
```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout
xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"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

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

    <TextView
        android:id="@+id/textView"
        android:layout_width="121dp"
        android:layout_height="32dp"
        android:layout_alignParentEnd="true"
        android:layout_alignParentBottom="true"
        android:layout_marginEnd="145dp"
        android:layout_marginBottom="478dp"
        android:text="Counter Value" />

    <Button
        android:id="@+id/btn_start"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignParentEnd="true"
        android:layout_alignParentBottom="true"
        android:layout_marginEnd="234dp"
        android:layout_marginBottom="296dp"
        android:text="Start" />

    <Button
        android:id="@+id/btn_stop"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignParentEnd="true"
        android:layout_alignParentBottom="true"
        android:layout_marginEnd="73dp"
        android:layout_marginBottom="295dp"
        android:text="Stop" />

</RelativeLayout>
```

JAVA CODE

```
import androidx.appcompat.app.AppCompatActivity;
import android.os.Bundle;
import android.os.Handler;
import android.view.View;
import android.widget.Button;
import android.widget.TextView;
public class MainActivity extends AppCompatActivity {
    Button btnstart, btnstop;
    TextView txtcounter;
    int i=1;
    Handler customHandler=new Handler();
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        btnstart=findViewById(R.id.btn_start);
        btnstop=findViewById(R.id.btn_stop);
        txtcounter=findViewById(R.id.textView);
        btnstart.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                customHandler.postDelayed(updateTimerThread,0);
            }
        });
        btnstop.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                customHandler.removeCallbacks(updateTimerThread);
            }
        });
    }
    private final Runnable updateTimerThread=new Runnable() {
        @Override
        public void run() {
            txtcounter.setText(""+i);
            customHandler.postDelayed(this,1000);
            i++;
        }
    };
}
```

OUTPUT


11:07
CounterActivity
Counter Application
Counter Value
START
STOP


11:07
CounterActivity
Counter Application
9
START
STOP

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

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

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

    new->file->city.xml

    new->file->city.json

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

city.xml

```
<?xml version="1.0"?>
<records>
    <place>
        <name>Mysore</name>
        <lat>12.295</lat>
        <long>76.639</long>
        <temperature>22</temperature>
        <humidity>90 %</humidity>
    </place>
    <place>
        <name>Bangalore</name>
        <lat>12.97165</lat>
        <long>77.5946</long>
        <temperature>25</temperature>
        <humidity>74 %</humidity>
    </place>
</records>
```

city.json

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

XML CODE

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

    <TextView
        android:id="@+id/textView"
        android:layout_width="292dp"
        android:layout_height="wrap_content"
        android:layout_alignParentEnd="true"
        android:layout_alignParentBottom="true"
        android:layout_marginEnd="57dp"
        android:layout_marginBottom="597dp"
        android:text="Parsing XML and JSON"
        android:textAlignment="center"
        android:textSize="26dp" />

    <Button
        android:id="@+id/button"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignParentEnd="true"
        android:layout_alignParentBottom="true"
        android:layout_marginEnd="116dp"
        android:layout_marginBottom="486dp"
        android:onClick="parsexml"
        android:text="ParseXML" />

    <Button
        android:id="@+id/button2"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignParentEnd="true"
        android:layout_alignParentBottom="true"
        android:layout_marginEnd="116dp"
        android:layout_marginBottom="366dp"
        android:onClick="parsejson"
        android:text="ParseJSON" />

    <TextView
        android:id="@+id/display"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_alignParentEnd="true"
        android:layout_alignParentBottom="true"
        android:layout_marginEnd="3dp"
        android:layout_marginBottom="68dp"
        android:text=""
        android:textAlignment="center" />

</RelativeLayout>
```

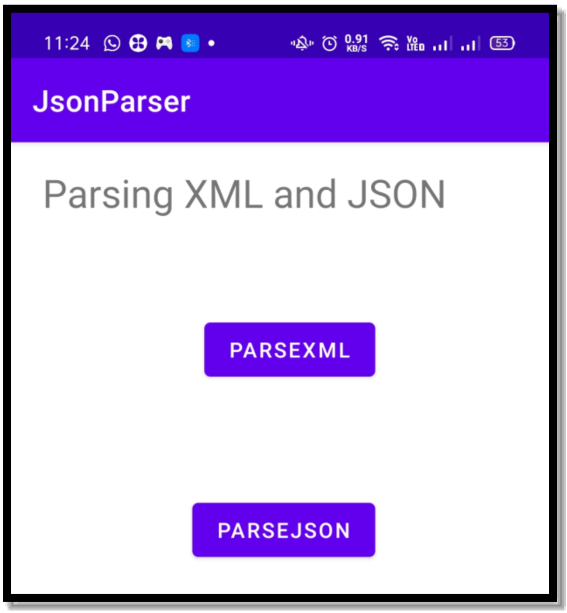
JAVA CODE

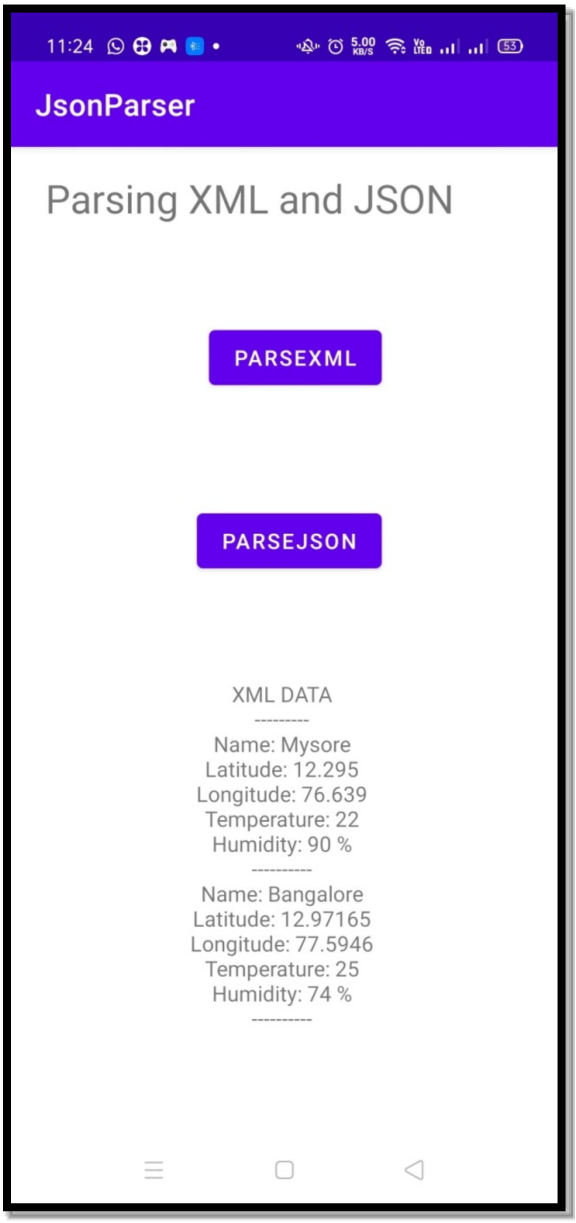
```
import android.os.Build;
import android.os.Bundle;
import android.text.style.TabStopSpan;
import android.view.View;
import android.widget.TextView;
import android.widget.Toast;
import androidx.annotation.RequiresApi;
import androidx.appcompat.app.AppCompatActivity;
import org.json.JSONArray;
import org.json.JSONObject;
import org.w3c.dom.Document;
import org.w3c.dom.Element;
import org.w3c.dom.Node;
import org.w3c.dom.NodeList;
import java.io.InputStream;
import java.nio.charset.StandardCharsets;
import javax.xml.parsers.DocumentBuilder;
import javax.xml.parsers.DocumentBuilderFactory;

public class MainActivity extends AppCompatActivity {
    TextView display;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        display = (TextView)findViewById(R.id.display);
    }
    public void parsexml(View V)
    {
        try {
            InputStream is = getAssets().open("city.xml");
            DocumentBuilderFactory documentBuilderFactory =
DocumentBuilderFactory.newInstance();
            DocumentBuilder documentBuilder =
documentBuilderFactory.newDocumentBuilder();
            Document document = documentBuilder.parse(is);
            StringBuilder stringBuilder = new StringBuilder();
            stringBuilder.append("XML DATA");
            stringBuilder.append("\n---------");
            NodeList nodeList =
document.getElementsByTagName("place");
```

```
            for (int i = 0; i < nodeList.getLength(); i++)
            {
                Node node = nodeList.item(i);
                if (node.getNodeType() == Node.ELEMENT_NODE) {
                    Element element = (Element) node;
                    stringBuilder.append("\nName:
").append(getValue("name", element));
                    stringBuilder.append("\nLatitude:
").append(getValue("lat", element));
                    stringBuilder.append("\nLongitude:
").append(getValue("long", element));
                    stringBuilder.append("\nTemperature:
").append(getValue("temperature", element));
                    stringBuilder.append("\nHumidity:
").append(getValue("humidity", element));
                    stringBuilder.append("\n----------");
                }
            }
            display.setText(stringBuilder.toString());
        }catch (Exception e){
            e.printStackTrace();
            Toast.makeText(MainActivity.this,"Error Parsing
XML",Toast.LENGTH_LONG).show();
        }
    }
    @RequiresApi(api = Build.VERSION_CODES.KITKAT)
    public void parsejson(View V){
        String json;
        StringBuilder stringBuilder = new StringBuilder();
        try {
            InputStream is = getAssets().open("city.json");
            int size = is.available();
            byte[] buffer = new byte[size];
            is.read(buffer);
            json = new String(buffer, StandardCharsets.UTF_8);
            JSONArray jsonArray = new JSONArray(json);
            stringBuilder.append("JSON DATA");
            stringBuilder.append("\n--------");
            for (int i = 0; i < jsonArray.length(); i++) {
                JSONObject jsonObject = jsonArray.getJSONObject(i);
                stringBuilder.append("\nName:
").append(jsonObject.getString("name"));
                stringBuilder.append("\nLatitude:
```

```
").append(jsonObject.getString("lat"));
                stringBuilder.append("\nLongitude:
").append(jsonObject.getString("long"));
                stringBuilder.append("\nTemperature:
").append(jsonObject.getString("temperature"));
                stringBuilder.append("\nHumidity:
").append(jsonObject.getString("humidity"));
                stringBuilder.append("\n----------");
            }
            display.setText(stringBuilder.toString());
            is.close();
        }
        catch (Exception e){
            e.printStackTrace();
            Toast.makeText(MainActivity.this,"Error in
reading",Toast.LENGTH_LONG).show();
        }
    }
    private String getValue(String tag, Element element)
    {
        return
element.getElementsByTagName(tag).item(0).getChildNodes().item(0).ge
tNodeValue();
    }
}
```

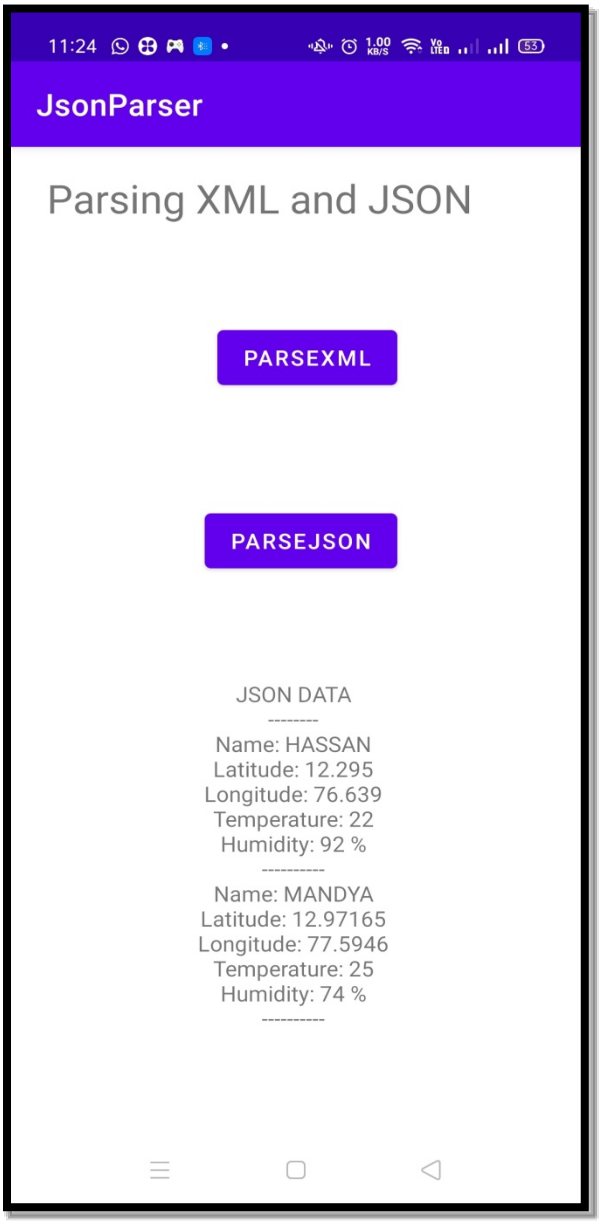
OUTPUT





Parsing XML and JSON

```
XML DATA
---------
Name: Mysore
Latitude: 12.295
Longitude: 76.639
Temperature: 22
Humidity: 90 %
----------
Name: Bangalore
Latitude: 12.97165
Longitude: 77.5946
Temperature: 25
Humidity: 74 %
----------
```



Parsing XML and JSON

```
JSON DATA
--------
Name: HASSAN
Latitude: 12.295
Longitude: 76.639
Temperature: 22
Humidity: 92 %
----------
Name: MANDYA
Latitude: 12.97165
Longitude: 77.5946
Temperature: 25
Humidity: 74 %
----------
```

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

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

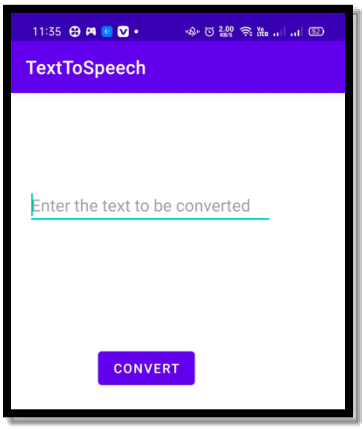
XML CODE

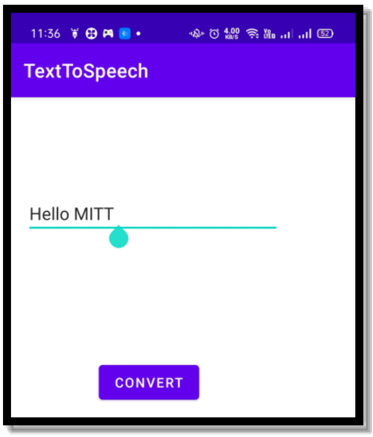
```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout
xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"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

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

    <EditText
        android:id="@+id/editText"
        android:layout_width="264dp"
        android:layout_height="wrap_content"
        android:layout_alignParentEnd="true"
        android:layout_alignParentRight="true"
        android:layout_alignParentBottom="true"
        android:layout_marginEnd="78dp"
        android:layout_marginRight="78dp"
        android:layout_marginBottom="505dp"
        android:ems="10"
        android:hint="Enter the text to be converted"
        android:inputType="textPersonName"
        android:text="" />

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

</RelativeLayout>
```

JAVA CODE

```
import androidx.appcompat.app.AppCompatActivity;
import android.os.Bundle;
import android.speech.tts.TextToSpeech;
import android.view.View;
import android.widget.EditText;
import android.widget.Toast;
import java.util.Locale;
public class MainActivity extends AppCompatActivity {
    TextToSpeech t1;
    EditText e1;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        e1 = findViewById(R.id.editText);
        t1 = new TextToSpeech(getApplicationContext(), new
                TextToSpeech.OnInitListener() {
                    @Override
                    public void onInit(int status) {
                        if (status!=TextToSpeech.ERROR){
                            t1.setLanguage(Locale.UK);
                        }
                    }
                });
    }
    public void convert(View view){
        String tospeak = e1.getText().toString();
        t1.speak(tospeak,TextToSpeech.QUEUE_FLUSH,null);
    }
}
```

OUTPUT





Hello MITT

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

1. Firstly Create an Application by Name “CallActivity”
2. Go to xml code of design change the layout to “RelativeLayout”
3. Add TextView component & change the following properties:
   - Size: 38dp
   - Text: Call Activity
   - Center-Align
4. Add EditText component & change the following properties in XML Code:
   - id: “@+id/phoneNumberEditText”
5. Add PlainText(EditText) component & change the following properties in XML Code:
   - Text: “”
   - Hint: “Copied Text”
   - id: “@+id/editText2”
6. Add three buttons to the design & change the text of the Buttons to “Clear”, “Call”, “Save” and change the id as follows:
   - id:”@+id/clearBtn”
   - id:”@+id/callBtn”
   - id:”@+id/saveBtn”
7. Add twelve buttons to the design & change the text of the Buttons as 1,2,3,4,5,6,7,8,9,0,*,#

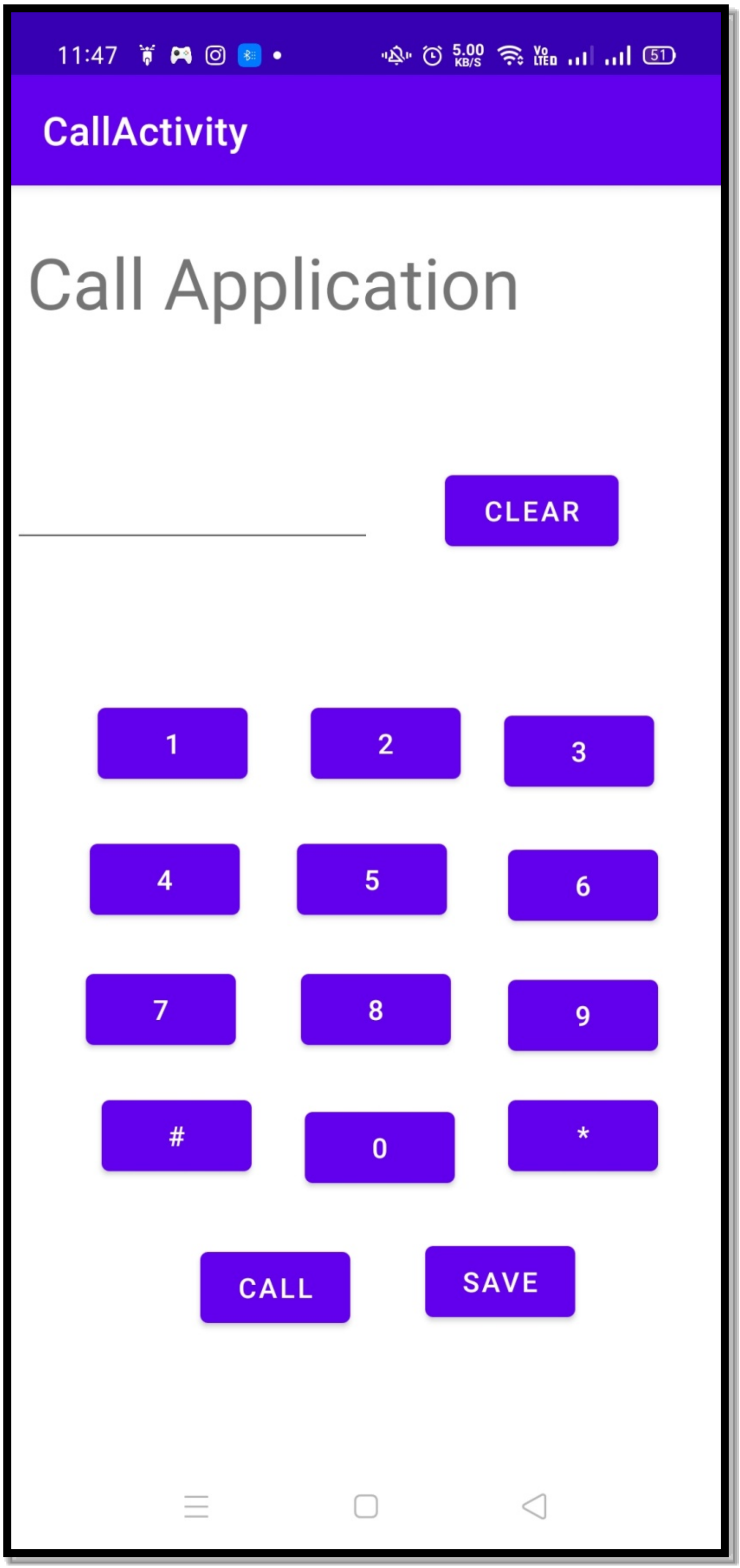
XML CODE

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout
xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".MainActivity">
    <TextView
        android:layout_width="298dp"
        android:layout_height="wrap_content"
        android:layout_alignParentEnd="true"
        android:layout_alignParentBottom="true"
        android:layout_marginEnd="54dp"
        android:layout_marginBottom="575dp"
        android:text="Call Application"
        android:textSize="36dp"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintLeft_toLeftOf="parent"
        app:layout_constraintRight_toRightOf="parent"
        app:layout_constraintTop_toTopOf="parent" />

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

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

    <Button
        android:id="@+id/button9"
        android:layout_width="76dp"
        android:layout_height="wrap_content"
        android:layout_alignParentEnd="true"
        android:layout_alignParentBottom="true"
        android:layout_marginEnd="137dp"
        android:layout_marginBottom="206dp"
        android:onClick="inputNumber"
        android:text="8" />

    <Button
        android:id="@+id/button3"
        android:layout_width="76dp"
        android:layout_height="wrap_content"
        android:layout_alignParentEnd="true"
        android:layout_alignParentBottom="true"
        android:layout_marginEnd="132dp"
        android:layout_marginBottom="341dp"
        android:onClick="inputNumber"
        android:text="2" />

    <Button
        android:id="@+id/button4"
        android:layout_width="76dp"
        android:layout_height="wrap_content"
        android:layout_alignParentEnd="true"
        android:layout_alignParentBottom="true"
        android:layout_marginEnd="34dp"
        android:layout_marginBottom="337dp"
        android:onClick="inputNumber"
        android:text="3" />

    <Button
        android:id="@+id/button5"
        android:layout_width="76dp"
        android:layout_height="wrap_content"
        android:layout_alignParentEnd="true"
        android:layout_alignParentBottom="true"
        android:layout_marginEnd="244dp"
        android:layout_marginBottom="272dp"
        android:onClick="inputNumber"
        android:text="4" />

    <Button
        android:id="@+id/button6"
        android:layout_width="76dp"
        android:layout_height="wrap_content"
        android:layout_alignParentEnd="true"
        android:layout_alignParentBottom="true"
        android:layout_marginEnd="139dp"
        android:layout_marginBottom="272dp"
        android:onClick="inputNumber"
        android:text="5" />

    <Button
        android:id="@+id/button7"
        android:layout_width="76dp"
        android:layout_height="wrap_content"
        android:layout_alignParentEnd="true"
        android:layout_alignParentBottom="true"
        android:layout_marginEnd="32dp"
        android:layout_marginBottom="269dp"
        android:onClick="inputNumber"
        android:text="6" />

    <Button
        android:id="@+id/button2"
        android:layout_width="76dp"
        android:layout_height="wrap_content"
        android:layout_alignParentEnd="true"
        android:layout_alignParentBottom="true"
        android:layout_marginEnd="240dp"
        android:layout_marginBottom="341dp"
        android:onClick="inputNumber"
        android:text="1" />

    <Button
        android:id="@+id/button8"
        android:layout_width="76dp"
        android:layout_height="wrap_content"
        android:layout_alignParentEnd="true"
        android:layout_alignParentBottom="true"
        android:layout_marginEnd="246dp"
        android:layout_marginBottom="206dp"
        android:onClick="inputNumber"
        android:text="7" />

    <Button
        android:id="@+id/button11"
        android:layout_width="76dp"
        android:layout_height="wrap_content"
        android:layout_alignParentEnd="true"
        android:layout_alignParentBottom="true"
        android:layout_marginEnd="238dp"
        android:layout_marginBottom="142dp"
        android:onClick="inputNumber"
        android:text="#" />

    <Button
        android:id="@+id/button12"
        android:layout_width="76dp"
        android:layout_height="wrap_content"
        android:layout_alignParentEnd="true"
        android:layout_alignParentBottom="true"
        android:layout_marginEnd="135dp"
        android:layout_marginBottom="136dp"
        android:onClick="inputNumber"
        android:text="0" />

    <Button
        android:id="@+id/button13"
        android:layout_width="76dp"
        android:layout_height="wrap_content"
        android:layout_alignParentEnd="true"
        android:layout_alignParentBottom="true"
        android:layout_marginEnd="32dp"
        android:layout_marginBottom="142dp"
        android:onClick="inputNumber"
        android:text="*" />

    <Button
        android:id="@+id/callBtn"
        android:layout_width="76dp"
        android:layout_height="wrap_content"
        android:layout_alignParentEnd="true"
        android:layout_alignParentBottom="true"
        android:layout_marginEnd="188dp"
        android:layout_marginBottom="65dp"
        android:text="Call" />

    <Button
        android:id="@+id/saveBtn"
        android:layout_width="76dp"
        android:layout_height="wrap_content"
        android:layout_alignParentEnd="true"
        android:layout_alignParentBottom="true"
        android:layout_marginEnd="74dp"
        android:layout_marginBottom="68dp"
        android:text="Save" />

    <Button
        android:id="@+id/button10"
        android:layout_width="76dp"
        android:layout_height="wrap_content"
        android:layout_alignParentEnd="true"
        android:layout_alignParentBottom="true"
        android:layout_marginEnd="32dp"
        android:layout_marginBottom="203dp"
        android:onClick="inputNumber"
        android:text="9" />

</RelativeLayout>
```

JAVA CODE

```
import androidx.appcompat.app.AppCompatActivity;
import android.content.Intent;
import android.net.Uri;
import android.os.Bundle;
import android.provider.ContactsContract;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
public class MainActivity extends AppCompatActivity {
    EditText phoneNumberEditText;
    Button clearBtn,callBtn,saveBtn;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        phoneNumberEditText=findViewById(R.id.phoneNumberEditText);
        callBtn=findViewById(R.id.callBtn);
        saveBtn=findViewById(R.id.saveBtn);
        clearBtn=findViewById(R.id.clearBtn);
        clearBtn.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                phoneNumberEditText.setText("");
            }
        });
        callBtn.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                String
phoneNumber=phoneNumberEditText.getText().toString();
                Intent intent=new Intent(Intent.ACTION_DIAL);
                intent.setData(Uri.parse("tel:"+phoneNumber));
                startActivity(intent);
            }
        });
        saveBtn.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                String
phoneNumber=phoneNumberEditText.getText().toString();
                Intent intent=new Intent(Intent.ACTION_INSERT);

intent.setType(ContactsContract.Contacts.CONTENT_TYPE);

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

OUTPUT



11:47
5.00 KB/S
51
CallActivity
Call Application
CLEAR
1
2
3
4
5
6
7
8
9
#
0
*
CALL
SAVE

Viva Questions

1 What is Android?
2 What Is the Google Android SDK?
3 What is the Android Architecture?
4 Describe the Android Framework.
5 What is the importance of having an emulator within the Android environment?
6 What is the importance of XML-based layouts?
7 What items are important in every Android project?
8 Name the languages supported for Android development
9 What are Intents?
10 Describe Activities.
11 What is ConstraintLayout?
12 What is the AndroidManifest.xml?
13 What are the different versions of Android OS?
14 What is Query Language?
15 What is SQLite?
16 Who was the designer of SQLite?
17 What are the most important features of SQLite?
18 What are the advantages of using SQLite?
19 How would you create a database in SQLite?
20 How would you create a table in SQLite database?
21 How can you delete the existing records from a table in SQLite?
22 Explain the difference between SQL and SQLite.
23 List Out The Standard Sqlite Commands?
24 What is drawable folder in android?
25 How to launch an activity in android?
26 What is .apk extension in Android?
27 Describe Lifecycle of an Activity.
28 What is Toast in Android?