



Department of Computer Science and Engineering

Jnanaprabha, Virgo Nagar Post, Bengaluru-560049

Academic Year: 2022-23

LABORATORY MANUAL

SEMESTER : VI

SUBJECT : MOBILE APPLICATION DEVELOPMENT LAB

SUBCODE : 18CSMP68

NAME : _____

USN : _____

SECTION: _____

PROGRAM OUTCOMES

PO1: Engineering knowledge: Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.

PO2: Problem analysis: Identify, formulate, review research literature, and analyse complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences.

PO3: 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.

PO4: 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.

PO5: Modern tool usage: Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modelling to complex engineering activities with an understanding of the limitations.

PO6: 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.

PO7: 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.

PO8: Ethics: Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.

PO9: Individual and Team Work: Function effectively as an individual and as a member or leader in diverse teams, and in multi – disciplinary settings.

PO10: 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.

PO11: Project management and finance: Demonstrate knowledge and understanding of the 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.

PO12: 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.



Department of Computer Science and Engineering

INSTITUTE VISION AND MISSION

VISION

The East Point College of Engineering and Technology aspires to be a globally acclaimed institution, recognized for excellence in engineering education, applied research and nurturing students for holistic development.

MISSION

M1: To create engineering graduates through quality education and to nurture innovation, creativity and excellence in teaching, learning and research

M2: To serve the technical, scientific, economic and societal developmental needs of our communities

M3: To induce integrity, teamwork, critical thinking, personality development and ethics in students and to lay the foundation for lifelong learning



Department of Computer Science and Engineering

DEPARTMENT VISION AND MISSION

VISION

The department aspires to be a Centre of excellence in Computer Science & Engineering to develop competent professionals through holistic development.

MISSION

M1: To create successful Computer Science Engineering graduates through effective pedagogies, the latest tools and technologies, and excellence in teaching and learning.

M2: To augment experiential learning skills to serve technical, scientific, economic, and social developmental needs.

M3: To instil integrity, critical thinking, personality development, and ethics in students for a successful career in Industries, Research, and Entrepreneurship.

PROGRAM EDUCATIONAL OBJECTIVES (PEOs)

PEO 1: To produce graduates who can perform technical roles to contribute effectively in software industries and R&D Centre

PEO 2: To produce graduates having the ability to adapt and contribute in key domains of computer science and engineering to develop competent solutions.

PEO 3: To produce graduates who can provide socially and ethically responsible solutions while adapting to new trends in the domain to carve a successful career in the industry

PROGRAM SPECIFIC OUTCOMES (PSOs)

PSO1: To conceptualize, model, design, simulate, analyse, develop, test, and validate computing systems and solve technical problems arising in the field of computer science & engineering.

PSO2: To specialize in the sub-areas of computer science & engineering systems such as cloud computing, Robotic Process Automation, cyber security, big data analytics, user interface design, and IOT to meet industry requirements.

PSO3: To build innovative solutions to meet the demands of the industry using appropriate tools and techniques.

COURSE LEARNING OBJECTIVES

CLO 1: Learn and acquire the art of Android Programming

CLO 2: Configure Android studio to run the applications

CLO 3: Understand and implement Android's User interface functions

CLO 4: Create, modify and query on SQLite database.

CLO 5: Inspect different methods of sharing data using services

COURSE OUTCOMES

At the end of the course the student will be able to:

CO1: Create, test and debug Android application by setting up Android development environment.

CO2: Implement adaptive, responsive user interfaces that work across a wide range of devices.

CO3: Infer long running tasks and background work in Android applications.

CO4: Demonstrate methods in storing, sharing and retrieving data in Android applications.

CO5: Infer the role of permissions and security for Android applications

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


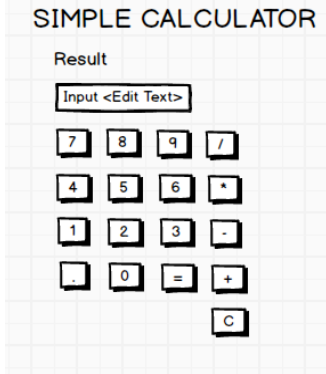
- Learn and acquire the art of Android Programming.
- Configure Android studio to run the applications.
- Understand and implement Android's User interface functions.
- Create, modify and query on SQLite database.
- Inspect different methods of sharing data using services.

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

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.

The image shows two side-by-side UI mockups. The left mockup is titled 'SIGNUP ACTIVITY' and contains labels for 'Username:' and 'Password:' followed by text input fields. Below the fields is a button labeled 'SIGN UP'. The right mockup is titled 'LOGIN ACTIVITY' and contains similar labels and input fields for 'Username' and 'Password', with a button labeled 'SIGN IN' below them.

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.

The image shows a UI mockup titled 'CHANGING WALLPAPER APPLICATION'. It features a single button in the center labeled '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.

The image shows a UI mockup titled 'COUNTER APPLICATION'. It displays a label 'Counter Value' above two buttons: 'START' and 'STOP'.

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.

The image shows a UI mockup titled 'TEXT TO SPEECH APPLICATION'. It features a button labeled 'Convert Text to Speech'.

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.

The image shows a UI mockup titled 'CALL AND SAVE APPLICATION'. It includes a text input field containing '1234567890' and a 'DEL' button. Below these is a numeric keypad with buttons for digits 1-9, 0, and a hash symbol. At the bottom are two buttons labeled 'CALL' and 'SAVE'.

PART - B

1

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

MEDICINE DATABASE

Medicine Name:

Date:

Time of the Day:

Insert

2

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

MEETING SCHEDULE

Date:

Time:

Meeting Agenda:

Add Meeting Agenda

MEETING INFO

Pick a date to get meeting info: / /

Search

3

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

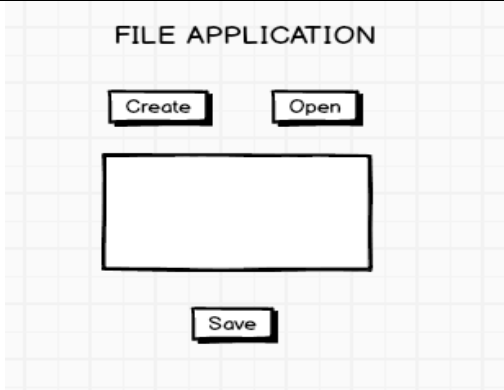
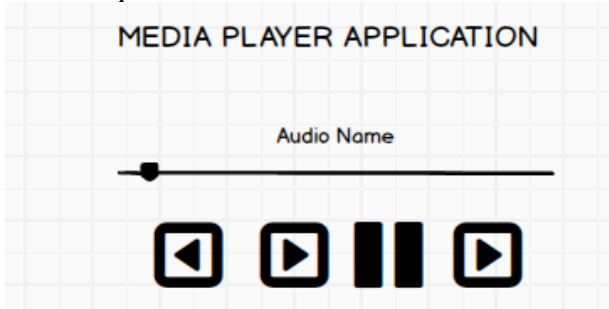
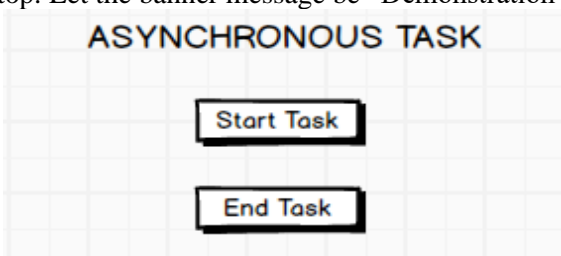
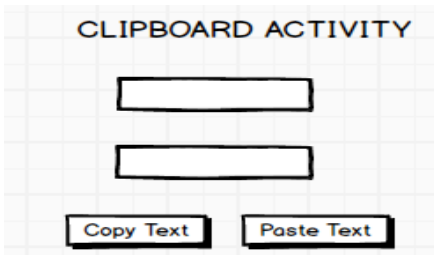
SMS APPLICATION

Display SMS Number

Display SMS Message

4

Write a program to create an activity having a Text box, and also Save, Open and Create buttons. The user has to write some text in the Text box. On pressing the Create button the text should be saved as a text file in Mkdirsdcard. 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	<p>Create an application to demonstrate a basic media player that allows the user to Forward, Backward, Play and Pause an audio. Also, make use of the indicator in the seek bar to move the audio forward or backward as required.</p> 
6	<p>Develop an application to demonstrate the use of Asynchronous tasks in android. The asynchronous task should implement the functionality of a simple moving banner. On pressing the Start Task button, the banner message should scroll from right to left. On pressing the Stop Task button, the banner message should stop. Let the banner message be “Demonstration of Asynchronous Task”.</p> 
7	<p>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.</p> 
8	<p>Create an AIDL service that calculates Car Loan EMI. The formula to calculate EMI is $E = P * (r(1+r)^n) / ((1+r)^n - 1)$ where E = The EMI payable on the car loan amount P = The Car loan Principal Amount r = The interest rate value computed on a monthly basis n = The loan tenure in the form of months The down payment amount has to be deducted from the principal amount paid towards buying the Car. Develop an application that makes use of this AIDL service to calculate the EMI. This application should have four EditText to read the Principal Amount, Down Payment, Interest Rate, Loan Term (in months) and a button named as “Calculate Monthly EMI”. On click of this button, the result should be shown in a TextView. Also, calculate the EMI by varying the Loan Term and Interest Rate values.</p>

CAR EMI CALCULATOR

Principal Amount:

Down Payment:

Interest Rate:

Loan Term (in months):

EMI: Result

Calculate Monthly EMI

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

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

Procedure to Conduct Practical Examination

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

Text Books:

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

Reference Books:

1. Erik Hellman, "**Android Programming – Pushing the Limits**", 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

Index

SL. NO	PROGRAM	CO	PO	RBT	PAGE NO.
1	Installing Android Studio and Packages		PO1	L2	2
2	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.	CO1 to CO5	PO1, PO3, PO5	L3	9
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.	CO1 to CO5	PO1, PO3, PO5	L3	16
4	Program 3: Create a SIGN Up activity with Username and Password. Validation of password should happen based on the following rules: <ul style="list-style-type: none"> • 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). 	CO1 to CO5	PO1, PO3, PO5	L3	31
5	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	CO1 to CO5	PO1, PO3, PO5	L3	44
6	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	CO1 to CO5	PO1, PO3, PO5	L3	49

	until the STOP button is pressed. Display the counter L3				
7	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.	CO1 to CO5	PO1, PO3, PO5	L3	55
8	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.	CO1 to CO5	PO1, PO3, PO5	L3	66
9	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.	CO1 to CO5	PO1, PO3, PO5	L3	72
10	Viva questions	-	-	L2	85

Course Articulation Matrix

COs	POs												PSOs		
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO1	3	2	2	-	3	-	-	-	3	3	1	1	3	2	-
CO2	3	2	2	-	3	-	-	-	3	3	1	1	3	2	-
CO3	3	2	2	-	3	-	-	-	3	3	1	1	3	2	-
CO4	3	2	2	-	3	-	-	-	3	3	1	1	3	2	-
CO5	3	2	2	-	3	-	-	-	3	3	1	1	3	2	-

3 - High Correlation

2 - Medium Correlation

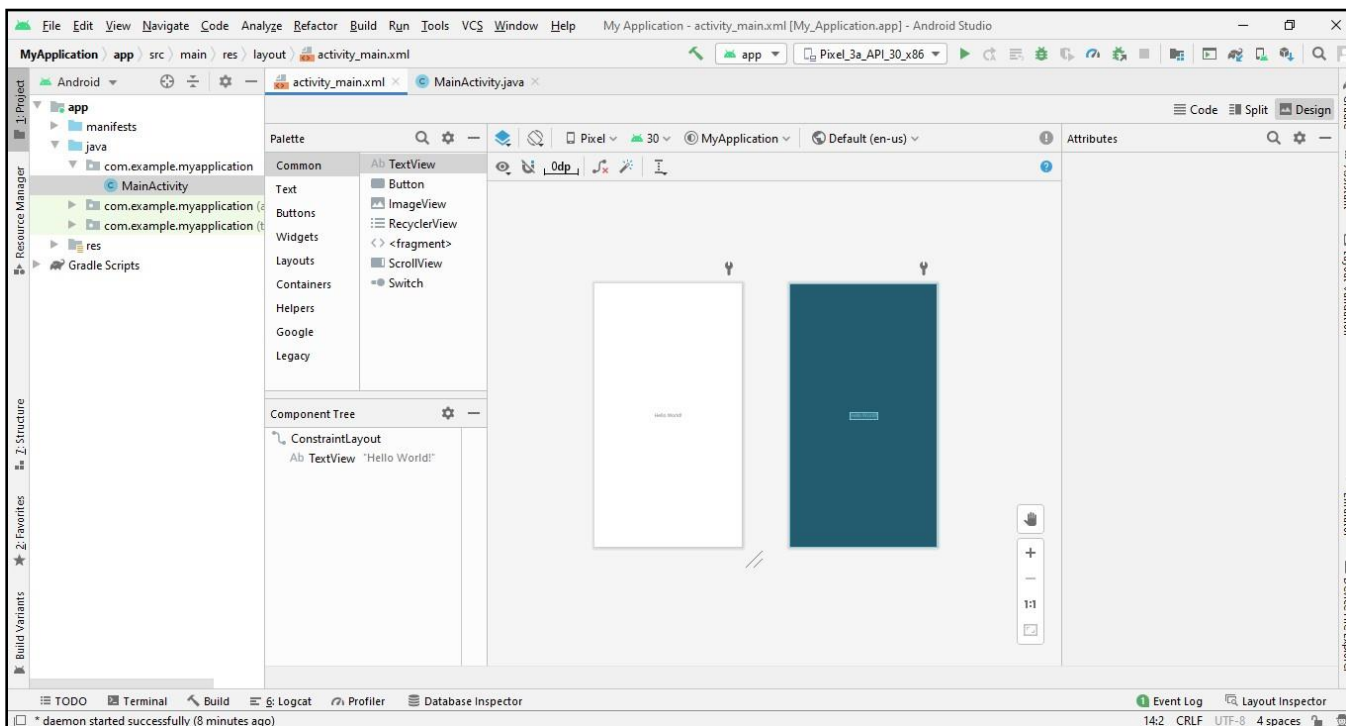
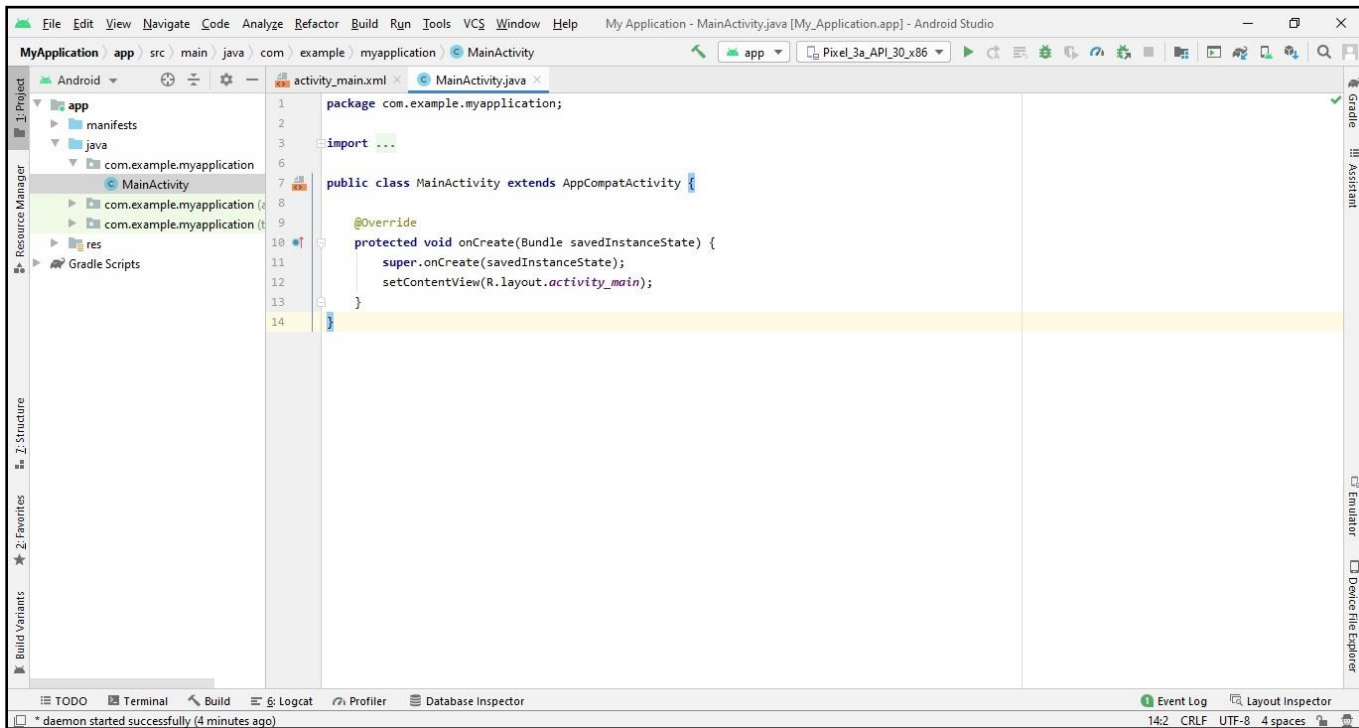
1 – Low Correlation

Installing Android Studio and Packages

Download Android Version 4.0.2 from the below link

<https://redirector.gvt1.com/edgedl/android/studio/install/4.0.2.0/android-studio-ide-193.6821437-windows.exe>

- Once installed successfully will be able to see following screen



Explore the project structure

In this practical, you will explore how the project files are organized in Android Studio.

These steps assume that your Hello World project starts out as shown in screenshots in page 1.

➤ Explore the project structure and layout

In the **Project > Android** view of your previous task, there are three top-level folders below your **app** folder: **manifests**, **java**, and **res**.

1. Expand the **manifests** folder.

This folder contains **AndroidManifest.xml**. This file describes all of the components of your Android app and is read by the Android run-time system when your program is executed.

2. Expand the **java** folder.

All your Java language files are organized in this folder. The **java** folder contains three subfolders:

- **com.example.hello.helloworld (or the domain name you have specified):** All the files for a package are in a folder named after the package. For your Hello World application, there is one package and it only contains MainActivity.java (the file extension may be omitted in the Project view).
- **com.example.hello.helloworld(androidTest):** This folder is for your instrumented tests, and starts out with a skeleton test file.
- **com.example.hello.helloworld(test):** This folder is for your unit tests and starts out with an automatically created skeleton unit test file.

3. Expand the **res** folder. This folder contains all the resources for your app, including images, layout files, strings, icons, and styling. It includes these subfolders:

- **drawable.** Store all your app's images in this folder.
- **layout.** Every activity has at least one layout file that describes the UI in XML. For Hello World, this folder contains activity_main.xml.
- **mipmap.** Store your launcher icons in this folder. There is a sub-folder for each supported screen density. Android uses the screen density, that is, the number of pixels per inch to determine the required image resolution. Android groups all actual screen densities into generalized densities, such as medium (mdpi), high (hdpi), or extra-extraextra-high (xxxhdpi). The ic_launcher.png folder contains the default launcher icons for all the densities supported by your app.

- **values.** Instead of hardcoding values like strings, dimensions, and colors in your XML and Java files, it is best practice to define them in their respective values file. This makes it easier to change and be consistent across your app.

4. Expand the **values** subfolder within the res folder. It includes these subfolders:

- **colors.xml.** Shows the default colors for your chosen theme, and you can add your own colors or change them based on your app's requirements.
- **dimens.xml.** Store the sizes of views and objects for different resolutions.
- **strings.xml.** Create resources for all your strings. This makes it easy to translate them to other languages.
- **styles.xml.** All the styles for your app and theme go here. Styles help give your app a consistent look for all UI elements.

Create a virtual device (emulator)

- In this task, you will use the Android Virtual Device (AVD) manager to create a virtual device or emulator that simulates the configuration for a particular type of Android device.
- Using the AVD Manager, you define the hardware characteristics of a device and its API level, and save it as a virtual device configuration.
- When you start the Android emulator, it reads a specified configuration and creates an emulated device that behaves exactly like a physical version of that device , but it resides on your computer .
- **Why:** With virtual devices, you can test your apps on different devices (tablets, phones) with different API levels to make sure it looks good and works for most users. You do not need to depend on having a physical device available for app development.

➤ Create a virtual device

In order to run an emulator on your computer, you have to create a configuration that describes the virtual device.

1. In Android Studio, select **Tools > Android > AVD Manager**, or click the **AVD Manager**  icon in the toolbar.

2. If you have created a virtual device before, the window shows all of your existing devices and the button is at the bottom. Otherwise follow step 3

3. Click the **+Create Virtual Device....**

The Select Hardware screen appears showing a list of preconfigured hardware devices. For each device, the table shows its diagonal display size (Size), screen resolution in pixels (Resolution), and pixel density (Density). For example the Nexus 5 device, the pixel density is xxhdpi, which means your app uses the launcher icons in the xxhdpi folder of the mipmap folder. Likewise, your app will use layouts and drawables from folders defined for that density as well.

4. Choose the for example Nexus 5 hardware device and click **Next**.

5. On the **System Image** screen, from the **Recommended** tab, choose which version of the Android system to run on the virtual device. You can select the latest system image. There are many more versions available than shown in the **Recommended** tab. Look at the **x86 Images** and **Other Images** tabs to see them.

6. If a **Download** link is visible next to a system image version, it is not installed yet, and you need to download it. If necessary, click the link to start the download, and click **Finish** when it's done.

6. On **System Image** screen, choose a system image and click **Next**.

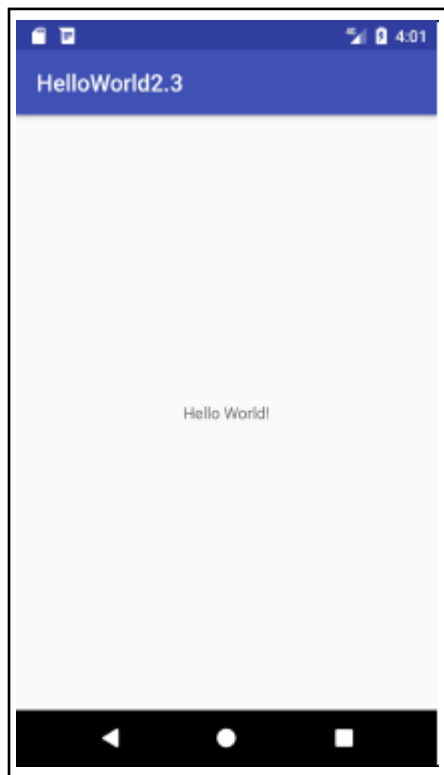
7. Verify your configuration, and click **Finish**. (If the **Your Android Devices** AVD Manager window stays open, you can go ahead and close it.)

Run your app on an emulator

1. In Android Studio, select **Run > Run app** or click the **Run icon**  in the toolbar.

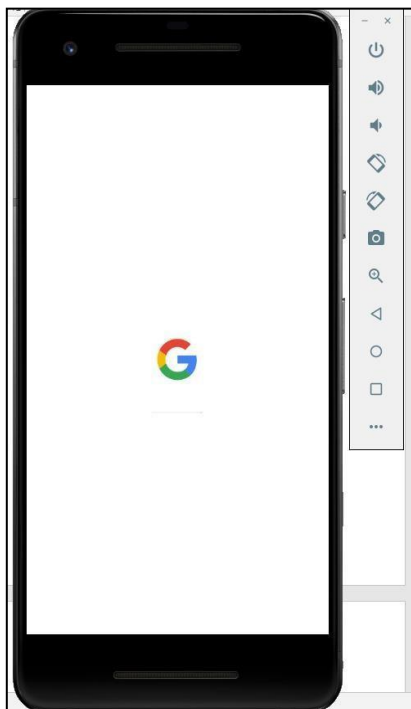
2. In the **Select Deployment Target** window, under **Available Emulators**, select **Nexus 5 API 23** and click **OK**.

- The emulator starts and boots just like a physical device. Depending on the speed of your computer, this may take a while.
- Your app builds, and once the emulator is ready, Android Studio will upload the app to the emulator and run it.
- You should see the Hello World app as shown in the following screenshot.



Nexus 5 API 23 Emulator showing output for hello world program.

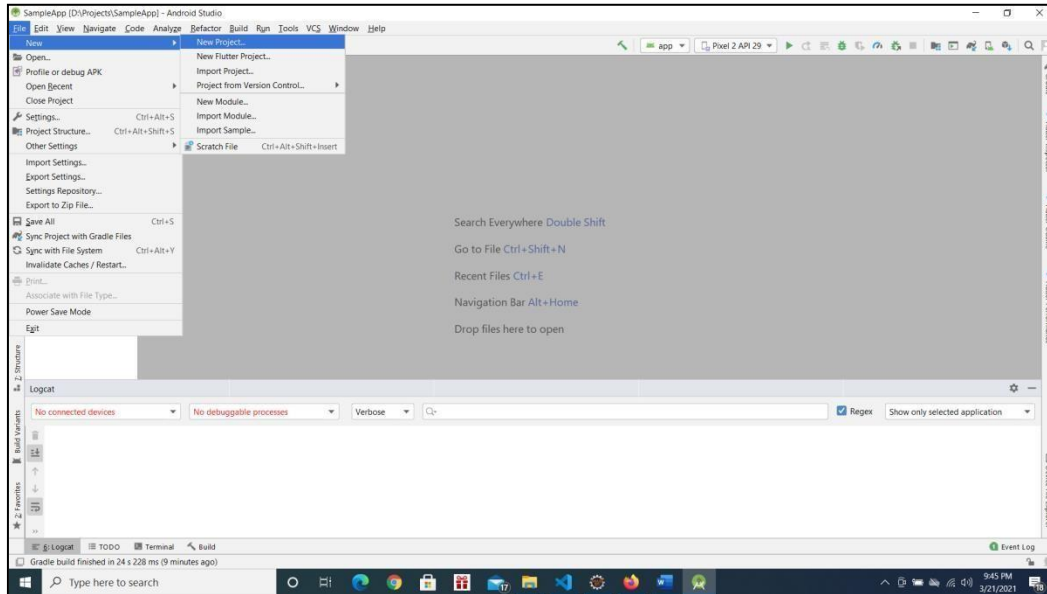
➤ If Pixel_3a_Api_30_x86 emulator is installed , then the emulator would look as shown below,



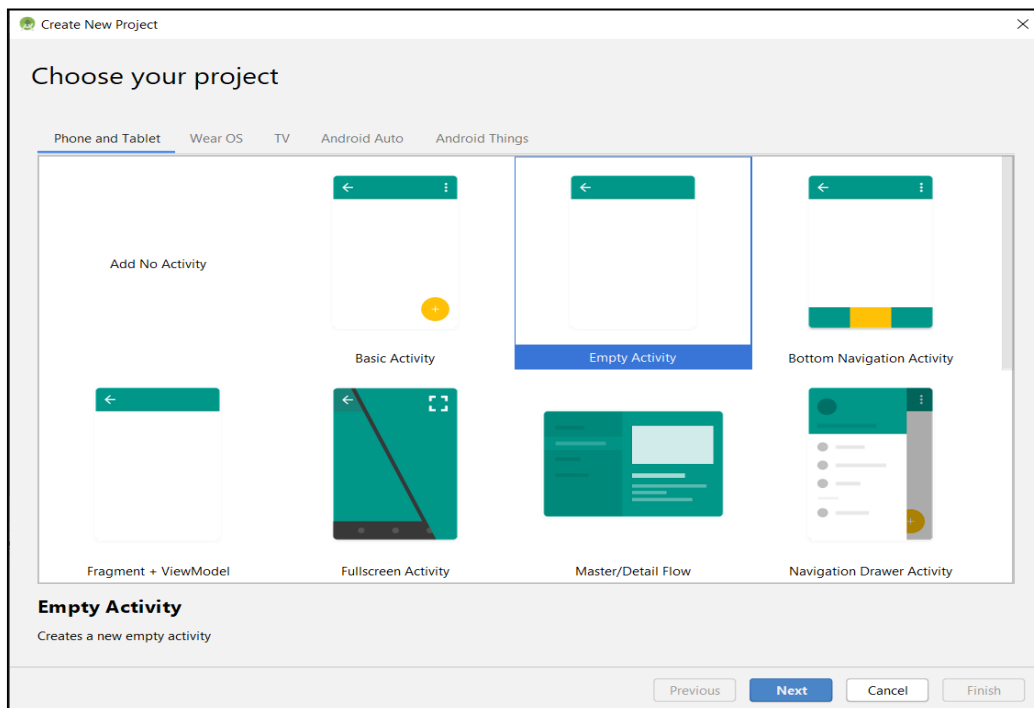
1.5 Creating a New Project in Android

While creating a New Project for First Time, make sure you have connected to internet, It downloads the required packages from internet.

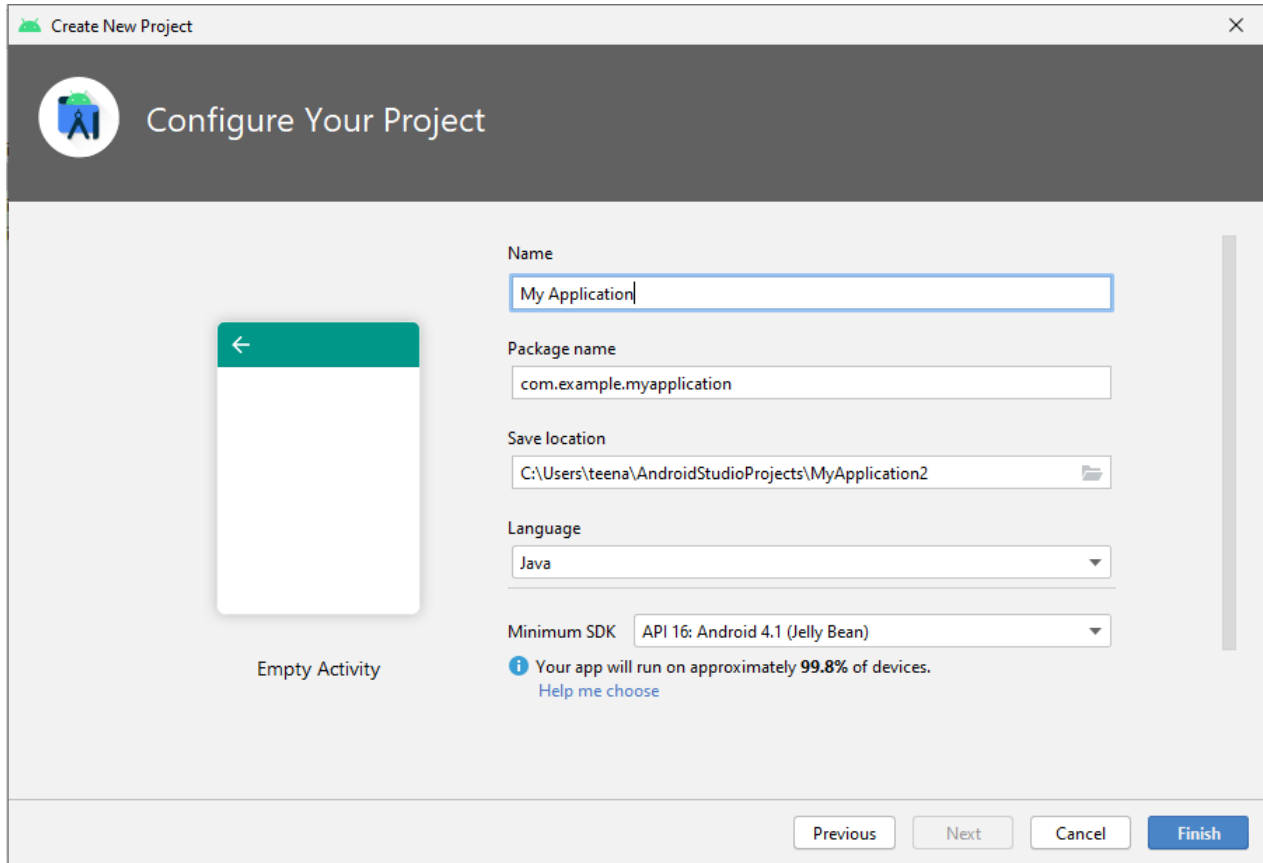
1. Go to File next click **New** then **New Project**, you will get the following screen



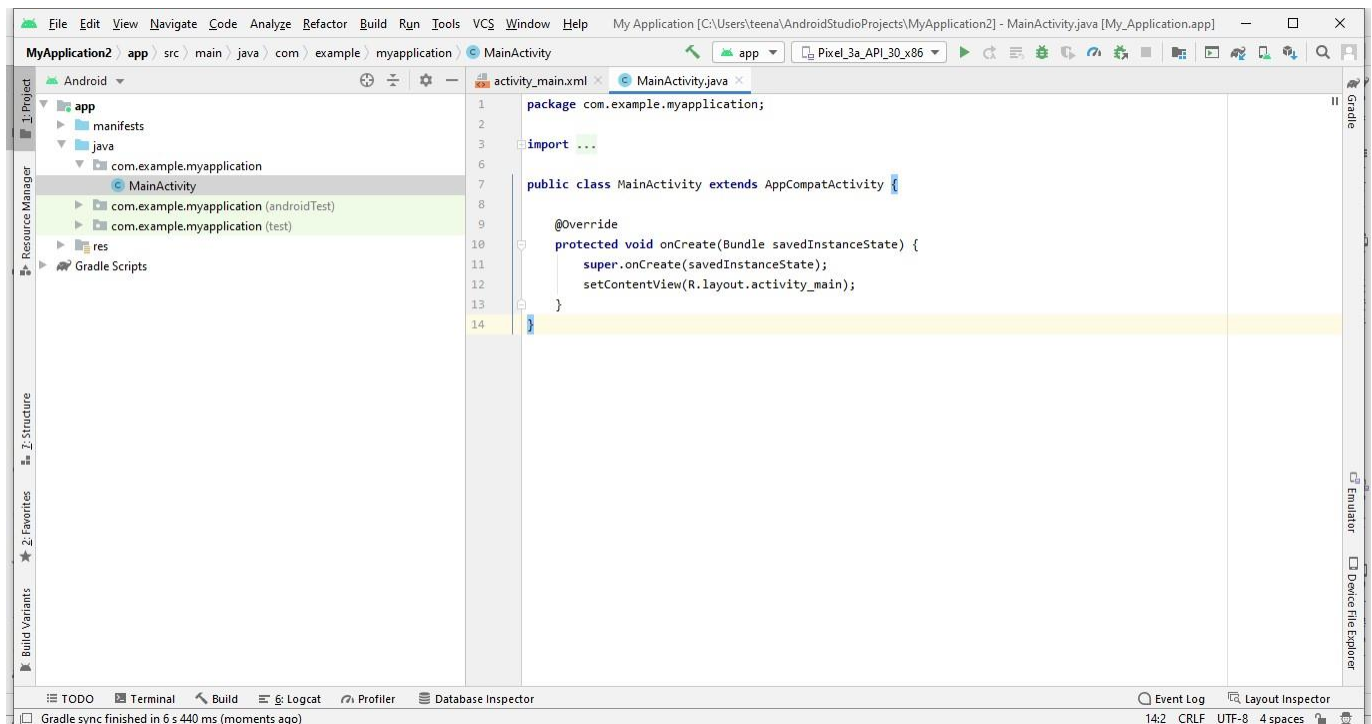
2. Choose **Phone and Tablet** tab and select **Empty Activity** and press **Next** as show in below screenshot



3. **Configure your Project** window will open, enter below details and press **Finish** button.

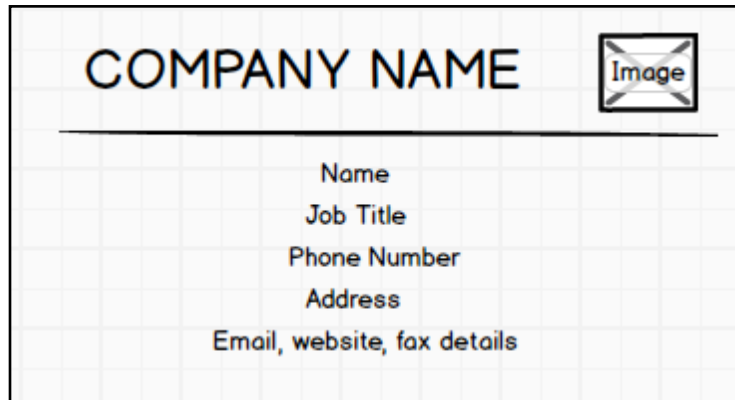


4. After pressing **Finish** button you will get the following screen, and you can start developing the app.



Program 1:

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

**UI Design (activity_main.xml):**

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
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">

<androidx.constraintlayout.widget.ConstraintLayout
    android:id="@+id/cardView"
    android:layout_width="421dp"
    android:layout_height="393dp"
    android:layout_marginStart="37dp"
    android:layout_marginLeft="37dp"
```

```
android:layout_marginTop="140dp" android:layout_marginEnd="32dp"
android:layout_marginRight="32dp"
android:layout_marginBottom="288dp"
android:background="#CDDC39"
app:layout_constraintBottom_toBottomOf="parent"
app:layout_constraintCircleRadius="12dp"
app:layout_constraintEnd_toEndOf="parent"
app:layout_constraintStart_toStartOf="parent"
app:layout_constraintTop_toTopOf="parent">
```

```
<TextView android:id="@+id/textView"
    android:layout_width="149dp"
    android:layout_height="61dp"
    android:layout_marginStart="44dp"
    android:layout_marginLeft="44dp"
    android:layout_marginTop="50dp"
    android:text="EPCET"
    android:textColor="#121111"
    android:textSize="46sp"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.0"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintVertical_bias="0.044" />
```

```
<View
    android:id="@+id/divider" android:layout_width="match_parent"
    android:layout_height="4dp"
```

```
android:layout_marginTop="130dp"
android:layout_marginBottom="259dp"
android:background="@color/black" android:divider="@color/black"
app:layout_constraintBottom_toBottomOf="parent"
app:layout_constraintEnd_toEndOf="parent"
app:layout_constraintStart_toStartOf="parent"
app:layout_constraintTop_toTopOf="parent" />
```

<ImageView

```
android:id="@+id/imageView2"
android:layout_width="146dp"
android:layout_height="106dp"
app:layout_constraintBottom_toBottomOf="parent"
app:layout_constraintEnd_toEndOf="parent"
app:layout_constraintHorizontal_bias="0.8"
app:layout_constraintStart_toEndOf="@+id/textView"
app:layout_constraintTop_toTopOf="parent"
app:layout_constraintVertical_bias="0.019"
app:srcCompat="@android:drawable/btn_star_big_on"
tools:srcCompat="@tools:sample/avatars" />
```

<TextView android:id="@+id/textView3"

```
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_marginStart="161dp"
android:layout_marginLeft="161dp"
android:layout_marginTop="145dp"
android:layout_marginEnd="185dp"
android:layout_marginRight="185dp"
android:layout_marginBottom="216dp"
android:text="xxxxxxxxx"
```

```
    android:textColor="#0E0D0D" android:textSize="24sp"
    android:textStyle="bold|italic"
    android:translationX="10dp"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="1.0"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintVertical_bias="1.0"
    tools:ignore="MissingConstraints" />
```

```
<TextView android:id="@+id/textView4"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginStart="129dp"
    android:layout_marginLeft="129dp"
    android:layout_marginTop="180dp"
    android:layout_marginEnd="146dp"
    android:layout_marginRight="146dp"
    android:layout_marginBottom="181dp"
    android:text="Asst. Professor"
    android:textColor="#0C0C0C" android:textSize="24sp"
    android:textStyle="normal" android:translationX="8dp"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent" />
```

```
<TextView
```

```
android:id="@+id/textView5"
android:layout_width="144dp"
android:layout_height="35dp"
android:layout_marginStart="143dp"
android:layout_marginLeft="143dp"
android:layout_marginTop="222dp"
android:layout_marginEnd="154dp"
android:layout_marginRight="154dp"
android:layout_marginBottom="136dp"
android:text="9898989898"
android:textColor="#130D0D" android:textSize="24sp"
android:translationX="8dp"
app:layout_constraintBottom_toBottomOf="parent"
app:layout_constraintEnd_toEndOf="parent"
app:layout_constraintStart_toStartOf="parent"
app:layout_constraintTop_toTopOf="parent" />
```

```
<TextView android:id="@+id/textView2"
    android:layout_width="wrap_content"
    android:layout_height="0dp"
    android:layout_marginStart="10dp"
    android:layout_marginLeft="10dp"
    android:layout_marginTop="257dp"
    android:layout_marginEnd="7dp"
    android:layout_marginRight="7dp"
    android:layout_marginBottom="105dp"
    android:text="'Jnana Prabha", Bidarahalli,Bngl-49'
    android:textColor="#0B0A0A" android:textSize="23sp"
    android:textStyle="normal"
    app:layout_constraintBottom_toBottomOf="parent"
```



```
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent" />
<TextView android:id="@+id/textView6"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginStart="43dp"
    android:layout_marginLeft="43dp"
    android:layout_marginTop="288dp"
    android:layout_marginEnd="56dp"
    android:layout_marginRight="56dp"
    android:layout_marginBottom="73dp"
    android:text="xyz@gmail.com, www.xyz.com"
    android:textColor="#0C0C0C" android:textSize="24sp"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent" />
</androidx.constraintlayout.widget.ConstraintLayout>
</androidx.constraintlayout.widget.ConstraintLayout>
```

JAVA code:No need to change anything...

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

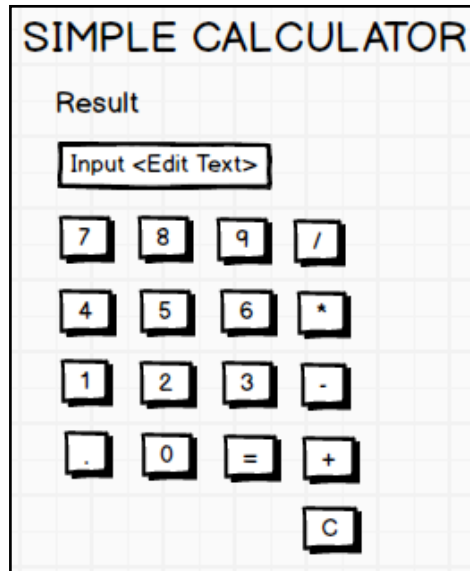
```
}  
}
```

Output:



Program 2:

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

**UI Design (activity_main.xml):**

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
    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">

    <Button
        android:id="@+id/clear_btn"
        android:layout_width="65dp"
        android:layout_height="55dp"
        android:layout_marginBottom="16dp" android:text="C"
```

```
android:textSize="24sp"
android:textStyle="bold"
app:layout_constraintBottom_toBottomOf="parent"
app:layout_constraintEnd_toEndOf="parent"
app:layout_constraintHorizontal_bias="0.891"
app:layout_constraintStart_toStartOf="parent"
tools:ignore="HardcodedText" />
```

```
<Button android:id="@+id/add_butn"
    android:layout_width="65dp"
    android:layout_height="55dp"
    android:layout_marginEnd="40dp"
    android:layout_marginRight="40dp"
    android:layout_marginBottom="20dp"
    android:text="+" android:textSize="24sp"
    android:textStyle="bold"
    app:layout_constraintBottom_toTopOf="@+id/clr_butn"
    app:layout_constraintEnd_toEndOf="parent"
    tools:ignore="HardcodedText" />
```

```
<Button
    android:id="@+id/eq_butn"
    android:layout_width="58dp"
    android:layout_height="55dp"
    android:layout_marginEnd="32dp"
    android:layout_marginRight="32dp"
    android:layout_marginBottom="92dp"
    android:text="=" android:textSize="24sp"
    android:textStyle="bold"
    app:layout_constraintBottom_toBottomOf="parent"
```

```
app:layout_constraintEnd_toStartOf="@+id/add_butn"  
tools:ignore="HardcodedText" />
```

<Button

```
android:id="@+id/zer_butn"  
android:layout_width="58dp"  
android:layout_height="55dp"  
android:layout_marginEnd="20dp"  
android:layout_marginRight="20dp"  
android:layout_marginBottom="92dp"  
android:text="0" android:textSize="24sp"  
android:textStyle="bold"  
app:layout_constraintBottom_toBottomOf="parent"  
app:layout_constraintEnd_toStartOf="@+id/eq_butn"  
tools:ignore="HardcodedText" />
```

<Button

```
android:id="@+id/dot_butn"  
android:layout_width="58dp"  
android:layout_height="55dp"  
android:layout_marginEnd="24dp"  
android:layout_marginRight="24dp"  
android:layout_marginBottom="92dp"  
android:text="." android:textSize="24sp"  
android:textStyle="bold"  
app:layout_constraintBottom_toBottomOf="parent"  
app:layout_constraintEnd_toStartOf="@+id/zer_butn"  
app:layout_constraintHorizontal_bias="1.0"  
app:layout_constraintStart_toStartOf="parent"  
tools:ignore="HardcodedText" />
```

```
<Button android:id="@+id/one_butn"
    android:layout_width="58dp"
    android:layout_height="55dp"
    android:layout_marginEnd="24dp"
    android:layout_marginRight="24dp"
    android:layout_marginBottom="12dp"
    android:text="1" android:textSize="24sp"
    android:textStyle="bold"
    app:layout_constraintBottom_toTopOf="@+id/dot_butn"
    app:layout_constraintEnd_toStartOf="@+id/two_butn"
    tools:ignore="HardcodedText" />
```

```
<Button android:id="@+id/two_butn"
    android:layout_width="58dp"
    android:layout_height="55dp"
    android:layout_marginEnd="20dp"
    android:layout_marginRight="20dp"
    android:layout_marginBottom="12dp"
    android:text="2" android:textSize="24sp"
    android:textStyle="bold"
    app:layout_constraintBottom_toTopOf="@+id/zer_butn"
    app:layout_constraintEnd_toStartOf="@+id/three_butn"
    tools:ignore="HardcodedText" />
```

```
<Button android:id="@+id/three_butn"
    android:layout_width="58dp"
    android:layout_height="55dp"
    android:layout_marginEnd="32dp"
```

```
android:layout_marginRight="32dp"
android:layout_marginBottom="12dp"
android:text="3" android:textSize="24sp"
android:textStyle="bold"
app:layout_constraintBottom_toTopOf="@+id/eq_butn"
app:layout_constraintEnd_toStartOf="@+id/sub_butn"
tools:ignore="HardcodedText" />
```

```
<Button android:id="@+id/sub_butn"
    android:layout_width="65dp"
    android:layout_height="55dp"
    android:layout_marginEnd="36dp"
    android:layout_marginRight="36dp"
    android:layout_marginBottom="12dp"
    android:text="-" android:textSize="24sp"
    android:textStyle="bold"
    app:layout_constraintBottom_toTopOf="@+id/add_butn"
    app:layout_constraintEnd_toEndOf="parent"
    tools:ignore="HardcodedText" />
```

```
<Button android:id="@+id/four_butn"
    android:layout_width="58dp"
    android:layout_height="55dp"
    android:layout_marginEnd="20dp"
    android:layout_marginRight="20dp"
    android:layout_marginBottom="16dp"
    android:text="4" android:textSize="24sp"
    android:textStyle="bold"
```

```
app:layout_constraintBottom_toTopOf="@+id/two_btn"
app:layout_constraintEnd_toStartOf="@+id/six_btn"
tools:ignore="HardcodedText" />
```

```
<Button android:id="@+id/five_btn"
    android:layout_width="58dp"
    android:layout_height="55dp"
    android:layout_marginEnd="24dp"
    android:layout_marginRight="24dp"
    android:layout_marginBottom="16dp"
    android:text="5" android:textSize="24sp"
    android:textStyle="bold"
    app:layout_constraintBottom_toTopOf="@+id/one_btn"
    app:layout_constraintEnd_toStartOf="@+id/four_btn"
    tools:ignore="HardcodedText" />
```

```
<Button
    android:id="@+id/six_btn"
    android:layout_width="58dp"
    android:layout_height="55dp"
    android:layout_marginEnd="28dp"
    android:layout_marginRight="28dp"
    android:layout_marginBottom="16dp"
    android:text="6" android:textSize="24sp"
    android:textStyle="bold"
    app:layout_constraintBottom_toTopOf="@+id/three_btn"
    app:layout_constraintEnd_toStartOf="@+id/mul_btn"
    tools:ignore="HardcodedText" />
```

```
<Button
```



```
android:id="@+id/mul_butn"
android:layout_width="65dp"
android:layout_height="55dp"
android:layout_marginEnd="40dp"
android:layout_marginRight="40dp"
android:layout_marginBottom="16dp"
android:text="*" android:textSize="24sp"
android:textStyle="bold"
app:layout_constraintBottom_toTopOf="@+id/sub_butn"
app:layout_constraintEnd_toEndOf="parent"
tools:ignore="HardcodedText" />
```

```
<Button android:id="@+id/seven_butn"
    android:layout_width="58dp"
    android:layout_height="55dp"
    android:layout_marginEnd="20dp"
    android:layout_marginRight="20dp"
    android:layout_marginBottom="20dp"
    android:text="7" android:textSize="24sp"
    android:textStyle="bold"
    app:layout_constraintBottom_toTopOf="@+id/five_butn"
    app:layout_constraintEnd_toStartOf="@+id/eight_butn"
    tools:ignore="HardcodedText" />
```

```
<Button android:id="@+id/eight_butn"
    android:layout_width="58dp"
    android:layout_height="55dp"
    android:layout_marginEnd="20dp"
    android:layout_marginRight="20dp"
```

```
android:layout_marginBottom="20dp"
android:text="8" android:textSize="24sp"
android:textStyle="bold"
app:layout_constraintBottom_toTopOf="@+id/four_btn"
app:layout_constraintEnd_toStartOf="@+id/nine_btn"
tools:ignore="HardcodedText" />
```

```
<Button android:id="@+id/nine_btn"
    android:layout_width="58dp"
    android:layout_height="55dp"
    android:layout_marginEnd="28dp"
    android:layout_marginRight="28dp"
    android:layout_marginBottom="20dp"
    android:text="9" android:textSize="24sp"
    android:textStyle="bold"
    app:layout_constraintBottom_toTopOf="@+id/six_btn"
    app:layout_constraintEnd_toStartOf="@+id/div_btn"
    tools:ignore="HardcodedText" />
```

```
<Button
    android:id="@+id/div_btn"
    android:layout_width="65dp"
    android:layout_height="55dp"
    android:layout_marginEnd="40dp"
    android:layout_marginRight="40dp"
    android:layout_marginBottom="20dp"
    android:text="/" android:textSize="24sp"
    android:textStyle="bold"
    app:layout_constraintBottom_toTopOf="@+id/mul_btn"
```

```
app:layout_constraintEnd_toEndOf="parent"  
tools:ignore="HardcodedText" />
```

<TextView

```
android:id="@+id/txt_res"  
android:layout_width="113dp"  
android:layout_height="36dp"  
android:layout_marginBottom="24dp"  
android:text="RESULT"  
android:textSize="30sp"  
android:textStyle="bold"  
app:layout_constraintBottom_toTopOf="@+id/result"  
app:layout_constraintEnd_toEndOf="parent"  
app:layout_constraintHorizontal_bias="0.167"  
app:layout_constraintStart_toStartOf="parent"  
tools:ignore="HardcodedText" />
```

<TextView

```
android:id="@+id/SIMPLE_CALC"  
android:layout_width="376dp"  
android:layout_height="46dp"  
android:layout_marginTop="4dp"  
android:text="SIMPLE CALCULATOR"  
android:textSize="36sp"  
android:textStyle="bold"  
app:layout_constraintBottom_toTopOf="@+id/txt_res"  
app:layout_constraintEnd_toEndOf="parent"  
app:layout_constraintStart_toStartOf="parent"  
app:layout_constraintTop_toTopOf="parent"  
tools:ignore="HardcodedText" />
```

<EditText

```
android:id="@+id/result"
```

```
    android:layout_width="231dp"
    android:layout_height="56dp"
    android:layout_marginBottom="152dp"android:ems="10"
    android:inputType="textPersonName"
    app:layout_constraintBottom_toTopOf="@+id/seven_btn"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.594"
    app:layout_constraintStart_toStartOf="parent"
    tools:ignore="Autofill,LabelFor" />
</androidx.constraintlayout.widget.ConstraintLayout>
```

MainActivity.java

```
package com.example.tkbcalc;
import androidx.appcompat.app.AppCompatActivity;
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;
import android.os.Bundle;

public class MainActivity extends AppCompatActivity implements View.OnClickListener {
    Button btnOne, btnTwo, btnThree, btnFour, btnFive, btnSix;
    Button btnSeven, btnEight, btnNine, btnZero;
    Button btnAdd,btnSub,btnMul,btnDiv;
    Button btnClear,btnEqual,btnDot;
    EditText txtResult;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
```

```
setContentView(R.layout.activity_main);

btnOne=(Button)findViewById(R.id.one_butn);
btnOne.setOnClickListener(this);
btnTwo=(Button)findViewById(R.id.two_butn);
btnTwo.setOnClickListener(this);
btnThree=(Button)findViewById(R.id.three_butn);
btnThree.setOnClickListener(this);
btnFour=(Button)findViewById(R.id.four_butn);
btnFour.setOnClickListener(this);
btnFive=(Button)findViewById(R.id.five_butn);
btnFive.setOnClickListener(this);
btnSix=(Button)findViewById(R.id.six_butn);
btnSix.setOnClickListener(this);
btnSeven=(Button)findViewById(R.id.seven_butn);
btnSeven.setOnClickListener(this);
btnEight=(Button)findViewById(R.id.eight_butn);
btnEight.setOnClickListener(this);
btnNine=(Button)findViewById(R.id.nine_butn);
btnNine.setOnClickListener(this);
btnZero=(Button)findViewById(R.id.zer_butn);
btnZero.setOnClickListener(this);
btnAdd=(Button)findViewById(R.id.add_butn);
btnAdd.setOnClickListener(this);
btnSub=(Button)findViewById(R.id.sub_butn);
btnSub.setOnClickListener(this);
btnMul=(Button)findViewById(R.id.mul_butn);
btnMul.setOnClickListener(this);
btnDiv=(Button)findViewById(R.id.div_butn);
btnDiv.setOnClickListener(this);
btnClear=(Button)findViewById(R.id.clr_butn);
btnClear.setOnClickListener(this);
btnEqual=(Button)findViewById(R.id.eq_butn);
```

```
btnEqual.setOnClickListener(this);
btnDot=(Button)findViewById(R.id.dot_btn);
btnDot.setOnClickListener(this);
txtResult=(EditText)findViewById(R.id.result);
txtResult.setText("");
}

public void onClick(View v)
{
    if(v.equals(btnOne))
        txtResult.append("1");
    if(v.equals(btnTwo))
        txtResult.append("2");
    if(v.equals(btnThree))
        txtResult.append("3");
    if(v.equals(btnFour))
        txtResult.append("4");
    if(v.equals(btnFive))
        txtResult.append("5");
    if(v.equals(btnSix))
        txtResult.append("6");
    if(v.equals(btnSeven))
        txtResult.append("7");
    if(v.equals(btnEight))
        txtResult.append("8");
    if(v.equals(btnNine))
        txtResult.append("9");
    if(v.equals(btnZero))
        txtResult.append("0");
    if(v.equals(btnDot))
        txtResult.append(".");
    if(v.equals(btnClear))
        txtResult.setText("");
    if(v.equals(btnAdd))
```

```
        txtResult.append("+");
    if(v.equals(btnSub))
        txtResult.append("-");
    if(v.equals(btnMul))
        txtResult.append("*");
    if(v.equals(btnDiv))
        txtResult.append("/");

    if(v.equals(btnEqual))
    {
        try {

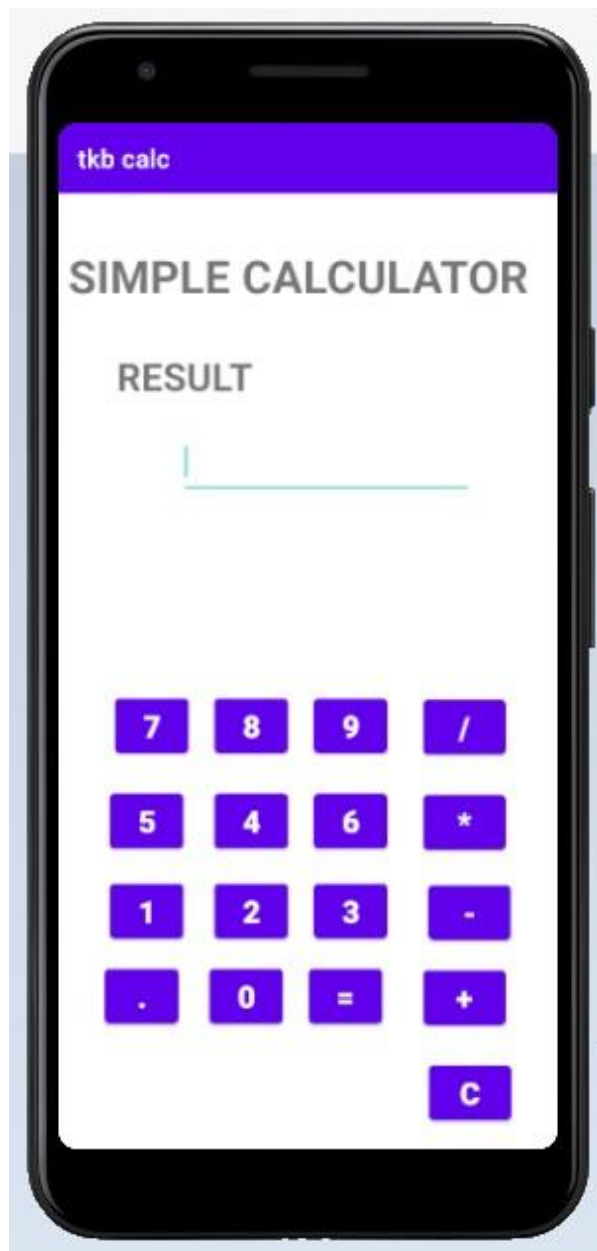
            String data = txtResult.getText().toString();
            if (data.contains("/")) {
                String[] operands = data.split("/");
                if(operands.length==2) {
                    double operand1 = Double.parseDouble(operands[0]);
                    double operand2 = Double.parseDouble(operands[1]);
                    double result = operand1 / operand2;
                    txtResult.setText(String.valueOf(result));
                }
                else
                {
                    Toast.makeText(getBaseContext(), "Invalid Input", Toast.LENGTH_LONG).show();
                }
            }
            else if (data.contains("*")) {
                String[] operands = data.split(Pattern.quote("*"));
                if(operands.length==2) {
                    double operand1 = Double.parseDouble(operands[0]);
                    double operand2 = Double.parseDouble(operands[1]);
                    double result = operand1 * operand2;
                    txtResult.setText(String.valueOf(result));
```

```
}  
else  
{  
    Toast.makeText(getBaseContext(), "Invalid Input", Toast.LENGTH_LONG).show();  
}  
}  
  
else if (data.contains("+")) {  
    String[] operands = data.split(Pattern.quote("+"));  
    if(operands.length==2) {  
        double operand1 = Double.parseDouble(operands[0]);  
        double operand2 = Double.parseDouble(operands[1]);  
        double result = operand1 + operand2;  
        txtResult.setText(String.valueOf(result));  
    }  
    else  
    {  
        Toast.makeText(getBaseContext(), "Invalid Input", Toast.LENGTH_LONG).show();  
    }  
}  
  
else if (data.contains("-")) {  
    String[] operands = data.split("-");  
    if(operands.length==2) {  
        double operand1 = Double.parseDouble(operands[0]);  
        double operand2 = Double.parseDouble(operands[1]);  
        double result = operand1 - operand2;  
        txtResult.setText(String.valueOf(result));  
    }  
    else  
    {  
        Toast.makeText(getBaseContext(), "Invalid Input", Toast.LENGTH_LONG).show();  
    }  
}  
}
```



```
catch(Exception e) {  
    Toast.makeText(getBaseContext(),"Invalid Input", Toast.LENGTH_LONG).show();  
}  
}  
}
```

Output:



Program 3:

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

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

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

The image shows two side-by-side UI designs for an Android application. The left design is for the 'SIGNUP ACTIVITY' and features a title 'SIGNUP ACTIVITY' at the top. Below it are two input fields: 'Username:' and 'Password:'. At the bottom is a button labeled 'SIGN UP'. The right design is for the 'LOGIN ACTIVITY' and features a title 'LOGIN ACTIVITY' at the top. Below it are two input fields: 'Username:' and 'Password:'. At the bottom is a button labeled 'SIGN IN'. Both designs are set against a light gray grid background.

UI Design (activity_main.xml):

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
    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/SignUp"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Sign Up"
    android:textAppearance="@style/TextAppearance.AppCompat.Large"
    android:textSize="30sp"
    android:textStyle="bold"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintHorizontal_bias="0.441"
    app:layout_constraintLeft_toLeftOf="parent"
    app:layout_constraintRight_toRightOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintVertical_bias="0.162" />
```

```
<TextView
    android:id="@+id/uname"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginStart="32dp"
    android:layout_marginLeft="32dp"
    android:layout_marginTop="76dp"
    android:layout_marginEnd="8dp"
    android:layout_marginRight="8dp"
    android:layout_marginBottom="20dp"
    android:text="User Name:"
    android:textSize="24sp"
    app:layout_constraintBottom_toTopOf="@+id/passw"
    app:layout_constraintEnd_toStartOf="@+id/signuname"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/SignUp" />
<EditText android:id="@+id/signuname"
    android:layout_width="wrap_content"
```

```
    android:layout_height="wrap_content"
    android:layout_marginStart="9dp"
    android:layout_marginLeft="9dp"
    android:layout_marginTop="58dp"
    android:layout_marginEnd="37dp"
    android:layout_marginRight="37dp"
    android:layout_marginBottom="44dp" android:ems="10"
    android:inputType="textPersonName"
    app:layout_constraintBottom_toTopOf="@+id/signpass"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toEndOf="@+id/uname"
    app:layout_constraintTop_toBottomOf="@+id/SignUp" />
<TextView
    android:id="@+id/passw"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginStart="41dp"
    android:layout_marginLeft="41dp"
    android:layout_marginTop="39dp"
    android:layout_marginEnd="8dp"
    android:layout_marginRight="8dp"
    android:layout_marginBottom="80dp"
    android:text="PassWord:"
    android:textSize="24sp"
    app:layout_constraintBottom_toTopOf="@+id/signbut"
    app:layout_constraintEnd_toStartOf="@+id/signpass"
    app:layout_constraintHorizontal_bias="0.0"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/uname"
    app:layout_constraintVertical_bias="1.0" />
<EditText
    android:id="@+id/signpass"
```

```
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_marginStart="9dp"
android:layout_marginLeft="9dp"
android:layout_marginTop="20dp"
android:layout_marginEnd="37dp"
android:layout_marginRight="37dp"
android:layout_marginBottom="77dp"
android:ems="10"
android:inputType="textPassword"
app:layout_constraintBottom_toTopOf="@+id/signbut"
app:layout_constraintEnd_toEndOf="parent"
app:layout_constraintStart_toEndOf="@+id/passw"
app:layout_constraintTop_toBottomOf="@+id/signuname" />
```

<Button

```
android:id="@+id/signbut"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_marginStart="129dp"
android:layout_marginLeft="129dp"
android:layout_marginTop="421dp"
android:layout_marginEnd="189dp"
android:layout_marginRight="189dp"
android:layout_marginBottom="262dp"
android:text="Sign Up"
app:layout_constraintBottom_toBottomOf="parent"
app:layout_constraintEnd_toEndOf="parent"
app:layout_constraintStart_toStartOf="parent"
app:layout_constraintTop_toTopOf="parent" />
```

</androidx.constraintlayout.widget.ConstraintLayout>

UI Design : login_layout.xml

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
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:id="@+id/linearLayout" android:layout_width="match_parent"
    android:layout_height="match_parent">

    <TextView
        android:id="@+id/textView"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginStart="160dp"
        android:layout_marginLeft="160dp"
        android:layout_marginTop="89dp"
        android:layout_marginEnd="160dp"
        android:layout_marginRight="160dp"
        android:layout_marginBottom="90dp"
        android:text="LogIn"
        android:textSize="36sp"
        android:textStyle="bold"
        app:layout_constraintBottom_toTopOf="@+id/textView2"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.0"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent" />

    <TextView android:id="@+id/textView2"
        android:layout_width="wrap_content"
```

```
android:layout_height="wrap_content"
android:layout_marginStart="25dp"
android:layout_marginLeft="25dp"
android:layout_marginTop="231dp"
android:layout_marginEnd="240dp"
android:layout_marginRight="240dp"
android:layout_marginBottom="460dp"
android:text="Username:" android:textSize="30sp"
android:textStyle="bold"
app:layout_constraintBottom_toBottomOf="parent"
app:layout_constraintEnd_toEndOf="parent"
app:layout_constraintStart_toStartOf="parent"
app:layout_constraintTop_toTopOf="parent" />
```

```
<TextView android:id="@+id/textView3"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginStart="37dp"
    android:layout_marginLeft="37dp"
    android:layout_marginTop="320dp"
    android:layout_marginEnd="232dp"
    android:layout_marginRight="232dp"
    android:layout_marginBottom="371dp"
    android:text="Password:" android:textSize="30sp"
    android:textStyle="bold"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent" />
```

```
<EditText android:id="@+id/loguname"
    android:layout_width="194dp"
    android:layout_height="49dp"
    android:layout_marginStart="195dp"
    android:layout_marginLeft="195dp"
    android:layout_marginTop="231dp"
    android:layout_marginEnd="23dp"
    android:layout_marginRight="23dp"
    android:layout_marginBottom="451dp"
    android:ems="10"
    android:inputType="textPersonName"
    android:textSize="24sp" android:textStyle="bold"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent" />
```

```
<EditText
    android:id="@+id/logpass"
    android:layout_width="184dp"
    android:layout_height="56dp"
    android:layout_marginStart="194dp"
    android:layout_marginLeft="194dp"
    android:layout_marginTop="317dp"
    android:layout_marginEnd="33dp"
    android:layout_marginRight="33dp"
    android:layout_marginBottom="358dp"
    android:ems="10"
    android:inputType="textPassword"
    android:textSize="24sp" android:textStyle="bold"
```



```
app:layout_constraintBottom_toBottomOf="parent"  
app:layout_constraintEnd_toEndOf="parent"  
app:layout_constraintStart_toStartOf="parent"  
app:layout_constraintTop_toTopOf="parent" />
```

<Button

```
android:id="@+id/logbut"  
android:layout_width="wrap_content"  
android:layout_height="wrap_content"  
android:layout_marginStart="155dp"  
android:layout_marginLeft="155dp"  
android:layout_marginTop="444dp"  
android:layout_marginEnd="124dp"  
android:layout_marginRight="124dp"  
android:layout_marginBottom="227dp"  
android:text="Login"  
android:textSize="30sp"  
android:textStyle="bold"  
app:layout_constraintBottom_toBottomOf="parent"  
app:layout_constraintEnd_toEndOf="parent"  
app:layout_constraintStart_toStartOf="parent"  
app:layout_constraintTop_toTopOf="parent" />
```

</androidx.constraintlayout.widget.ConstraintLayout>

MainActivity.java

```
package com.example.a3rdprogramlogin;

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.TextView;
import android.widget.Toast;


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


public class MainActivity extends AppCompatActivity implements View.OnClickListener {
    EditText signunname;
    EditText signpass;
    Button signbut;

    String regularExpression = "^(?=.*[0-9])(?=.*[A-Z])(?=.*[@#$%^&+=!])(?=\\S+$).{4,}$";
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        signunname = (EditText) findViewById(R.id.signunname);
        signpass = (EditText) findViewById(R.id.signpass);
        signbut = (Button) findViewById(R.id.signbut);
        signbut.setOnClickListener(this);

    }
```

```
public void onClick(View v)
{
    String uname=signuname.getText().toString();
    String pass=signpass.getText().toString();
    if(pass.length()>=8 && validatePassword(pass)) {
        Bundle bundle = new Bundle();
        bundle.putString("username", uname);
        bundle.putString("password", pass);

        Intent it = new Intent(this, logactivity.class);
        it.putExtra("data", bundle);
        startActivity(it);
    }
    else
    {
        Toast.makeText(getBaseContext(), "Invalid Password", Toast.LENGTH_LONG).show();
    }
}

public boolean validatePassword(String password)
{
    Pattern pattern= Pattern.compile(regularExpression);
    Matcher matcher=pattern.matcher(password);
    return matcher.matches();
}
}
```

logactivity.java

```
package com.example.a3rdprogramlogin;
```

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

```
import android.os.Bundle;
```

```
import android.view.View;
```

```
import android.widget.Button;
```

```
import android.widget.EditText;
```

```
import android.widget.Toast;
```

```
public class logactivity extends AppCompatActivity implements View.OnClickListener {
```

```
    EditText loguname;
```

```
    EditText logpass;
```

```
    Button logbut;
```

```
    String user,pass;
```

```
    int count=0;
```

```
@Override
```

```
protected void onCreate(Bundle savedInstanceState) {
```

```
    super.onCreate(savedInstanceState);
```

```
    setContentView(R.layout.login_layout);
```

```
        loguname=(EditText) findViewById(R.id.loguname);
```

```
        logpass=(EditText) findViewById(R.id.logpass);
```

```
        logbut=(Button)findViewById(R.id.logbut);
```

```
        logbut.setOnClickListener(this);
```

```
        Bundle bundle=getIntent().getBundleExtra("data");
```

```
        user=bundle.getString("username");
```

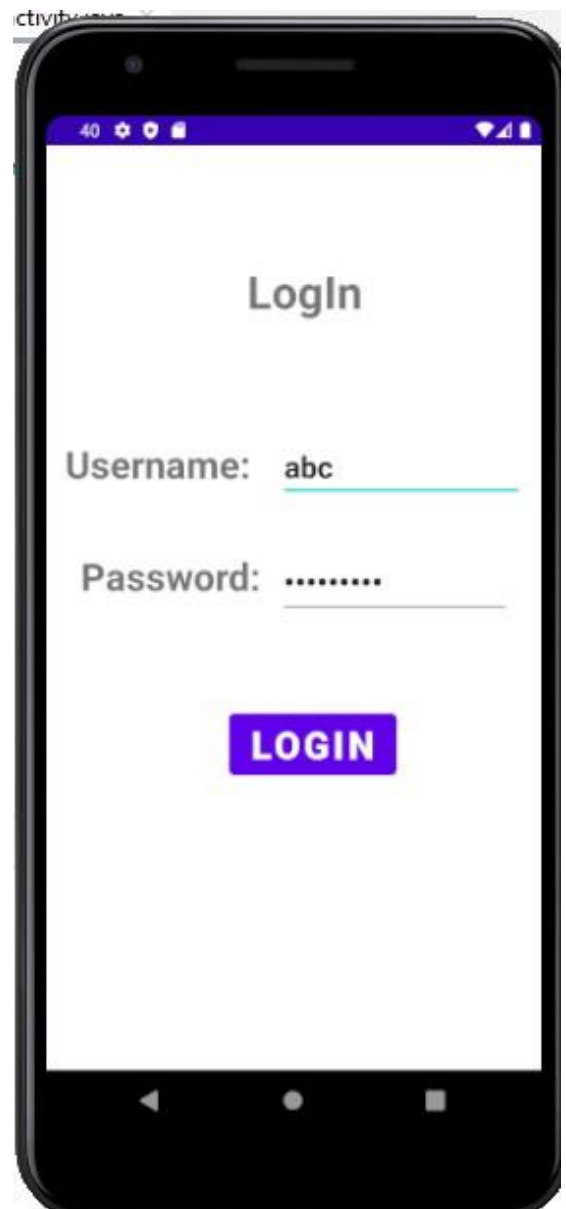
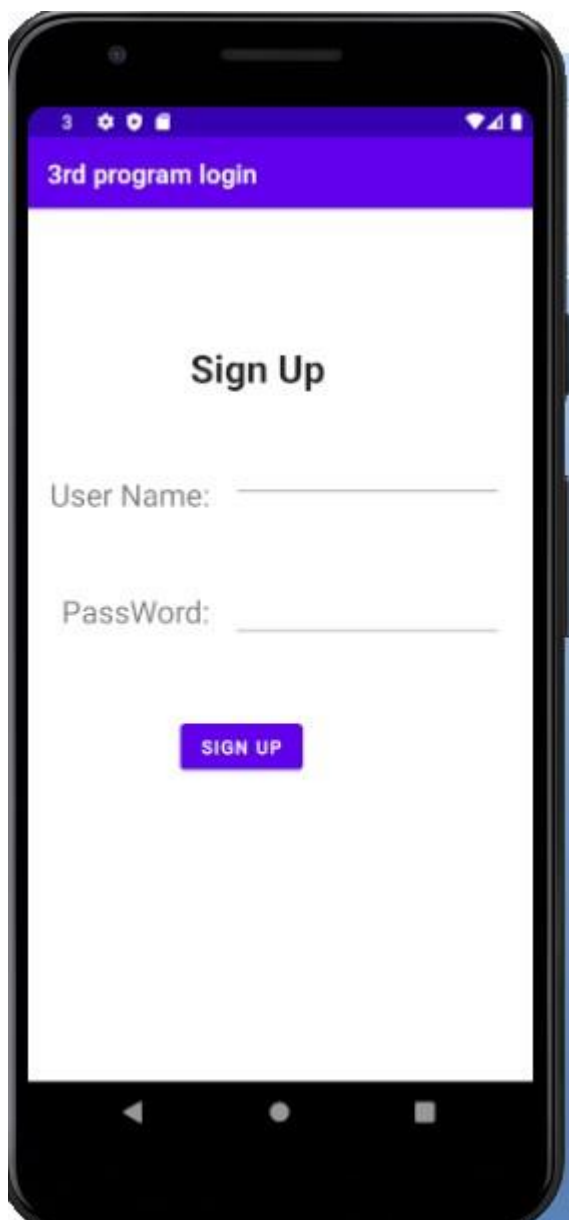
```
        pass=bundle.getString("password");
```

```
    }
```

```
public void onClick(View v)
{
    String user1=loguname.getText().toString();
    String pass1=logpass.getText().toString();

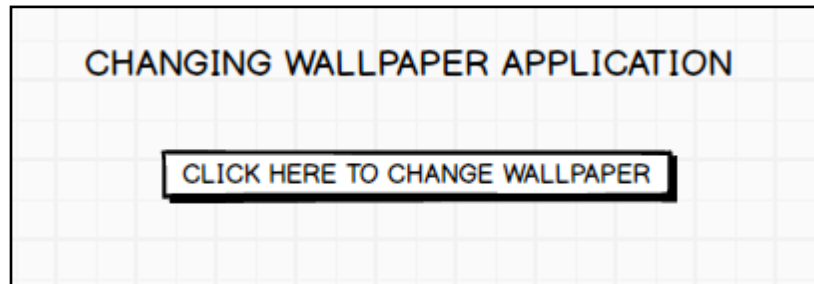
    if(user.equals(user1) && pass.equals(pass1))
    {
        Toast.makeText(this,"Login Successful",Toast.LENGTH_LONG).show();
    }
    else
    {
        count++;
        if(count==3)
        {
            logbut.setEnabled(false);
            Toast.makeText(this, "Failed Login Attempts",Toast.LENGTH_LONG).show();
        }
        else
        {
            Toast.makeText(this,"Login Failed "+count ,Toast.LENGTH_LONG).show();
        }
    }
}
```

output:



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.

**UI Design (activity_main.xml):**

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
    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:layerType="none" tools:context=".MainActivity">

    <TextView
        android:id="@+id/textView"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginStart="10dp" android:layout_marginLeft="10dp"
        android:layout_marginTop="110dp" android:layout_marginEnd="11dp"
        android:layout_marginRight="11dp"
        android:layout_marginBottom="591dp"
        android:text="@string/changing_wallpaper_application"
        android:textSize="22sp"
```

```
        android:textStyle="bold"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent" />

<Button android:id="@+id/btn_ch_wp"
        android:layout_width="195dp"
        android:layout_height="60dp"
        android:layout_marginStart="14dp"
        android:layout_marginLeft="14dp"
        android:layout_marginTop="203dp"
        android:layout_marginEnd="13dp"
        android:layout_marginRight="13dp"
        android:layout_marginBottom="480dp"
        android:text="@string/click_here_to_change_wallpaper"
        android:textSize="18sp"
        app:elevation="@dimen/cardview_default_radius"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent" />
</androidx.constraintlayout.widget.ConstraintLayout>
```


AndroidManifest.xml:

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.example.program4wallpaper">

    <uses-permission android:name="android.permission.SET_WALLPAPER"/>

    <application android:allowBackup="true"
        android:icon="@mipmap/ic_launcher"
        android:label="@string/app_name"
        android:roundIcon="@mipmap/ic_launcher_round"
        android:supportsRtl="true"
        android:theme="@style/Theme.Program4wallpaper">
        <activity android:name=".MainActivity">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />
                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
    </application>
</manifest>
```

MainActivity.java

```
package com.example.program4wallpaper;

import androidx.appcompat.app.AppCompatActivity;
import android.app.WallpaperManager;
import android.graphics.Bitmap;
import android.graphics.BitmapFactory;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
```

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

public class MainActivity extends AppCompatActivity implements View.OnClickListener {
    Button btnchwp;
    int[] imagesArray = new int[] {R.drawable.w1, R.drawable.w2, R.drawable.w3, R.drawable.w4,
                                    R.drawable.w5, R.drawable.w6};

    int i = 0;

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

        btnchwp = (Button) findViewById(R.id.butn_ch_wp);
        btnchwp.setOnClickListener(this);
    }

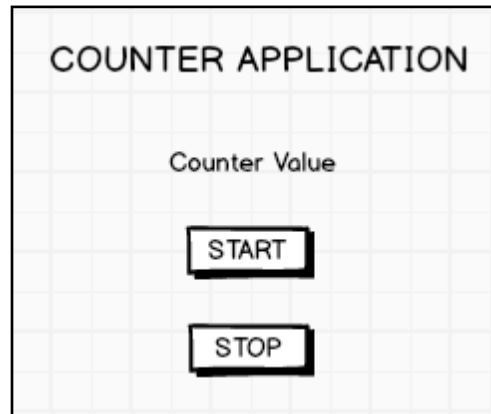
    public void onClick(View v) {
        new Timer().schedule(new MyTimer(), 0, 3000);
    }

    class MyTimer extends TimerTask {
        public void run() {
            try {
                WallpaperManager wpmng = WallpaperManager.getInstance(getBaseContext());
                wpmng.setBitmap(BitmapFactory.decodeResource(getResources(), imagesArray[i]));
                i++;
                if(i==6)
                {i=0;}
            } catch (IOException e) {
                e.printStackTrace();
            }
        }
    }
}
```

Output:

Program 5:

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

**UI Design (activity_main.xml):**

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
    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">

    <Button android:id="@+id/button3"
        android:layout_width="137dp"
        android:layout_height="59dp"
        android:text="Button"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
```

```
app:layout_constraintTop_toTopOf="parent" />
```

```
<TextView
```

```
    android:id="@+id/textView"  
    android:layout_width="365dp"  
    android:layout_height="100dp"  
    android:gravity="center_horizontal|center_vertical"  
    android:text="COUNTER APPLICATION" android:textSize="30sp"  
    android:textStyle="bold"  
    app:layout_constraintBottom_toBottomOf="parent"  
    app:layout_constraintEnd_toEndOf="parent"  
    app:layout_constraintStart_toStartOf="parent"  
    app:layout_constraintTop_toTopOf="parent"  
    app:layout_constraintVertical_bias="0.092" />
```

```
<EditText
```

```
    android:id="@+id/counter"  
    android:layout_width="72dp"  
    android:layout_height="79dp"  
    android:ems="10"  
    android:inputType="textPersonName"  
    android:textSize="30sp"  
    app:layout_constraintBottom_toBottomOf="parent"  
    app:layout_constraintEnd_toEndOf="parent"  
    app:layout_constraintStart_toStartOf="parent"  
    app:layout_constraintTop_toTopOf="parent"  
    app:layout_constraintVertical_bias="0.315" />
```

```
<Button
```

```
    android:id="@+id/startbutn"  
    android:layout_width="137dp"  
    android:layout_height="59dp"
```

```
android:clickable="true"
android:text="START"
android:textSize="24sp"
android:textStyle="bold"
app:layout_constraintBottom_toBottomOf="parent"
app:layout_constraintEnd_toEndOf="parent"
app:layout_constraintStart_toStartOf="parent"
app:layout_constraintTop_toTopOf="parent" />
```

<Button

```
android:id="@+id/stopbutn"
android:layout_width="137dp"
android:layout_height="59dp"
android:text="STOP"
android:textSize="24sp"
android:textStyle="bold"
app:layout_constraintBottom_toBottomOf="parent"
app:layout_constraintEnd_toEndOf="parent"
app:layout_constraintHorizontal_bias="0.489"
app:layout_constraintStart_toStartOf="parent"
app:layout_constraintTop_toTopOf="parent"
app:layout_constraintVertical_bias="0.615" />
```

</androidx.constraintlayout.widget.ConstraintLayout>

MainActivity.java

```
package com.example.program5counter;
import androidx.appcompat.app.AppCompatActivity;
import android.annotation.SuppressLint;
import android.os.Bundle;
import android.os.Handler;
import android.os.Message;
import android.view.View;
import android.widget.Button;
```

```
import android.widget.TextView;
```

```
public class MainActivity extends AppCompatActivity implements View.OnClickListener {
```

```
    TextView counter;
```

```
    Button start_butn, stop_butn;
```

```
    int count=0;
```

```
    boolean running=false;
```

```
@Override
```

```
protected void onCreate(Bundle savedInstanceState) {
```

```
    super.onCreate(savedInstanceState);
```

```
    setContentView(R.layout.activity_main);
```

```
    counter=(TextView)findViewById(R.id.counter);
```

```
    start_butn=(Button)findViewById(R.id.startbutn);
```

```
    stop_butn=(Button)findViewById(R.id.stopbutn);
```

```
    stop_butn.setOnClickListener(this);
```

```
    start_butn.setOnClickListener(this);
```

```
}
```

```
public void onClick(View v)
```

```
{
```

```
    if(v.equals(start_butn))
```

```
{
```

```
        count=0;
```

```
        running=true;
```

```
        start_butn.setClickable(false);
```

```
        new MyCounter().start();
```

```
}
```

```
    else if(v.equals(stop_butn))
```

```
{
```

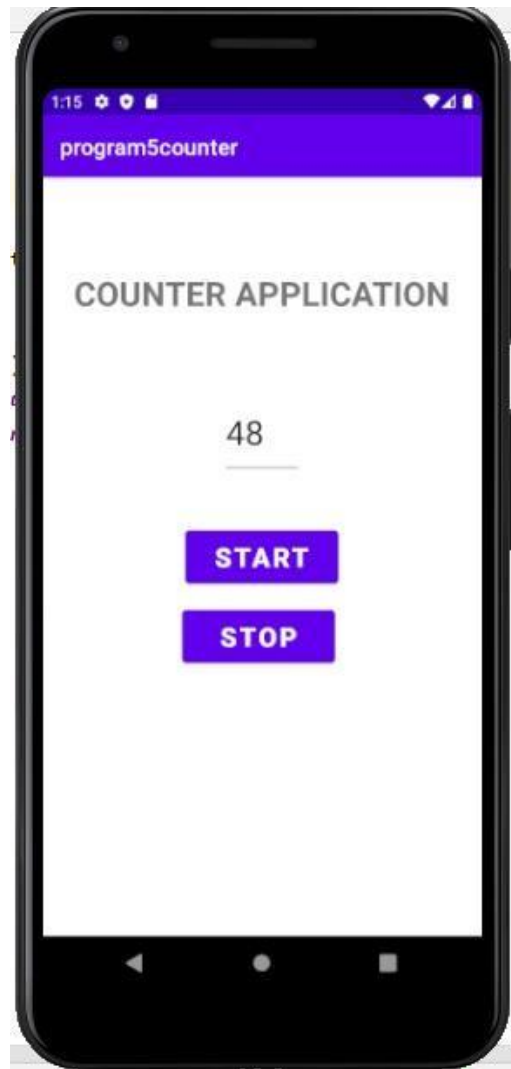
```
        running=false;
```

```
        start_butn.setClickable(true);
```

```
    }  
}
```

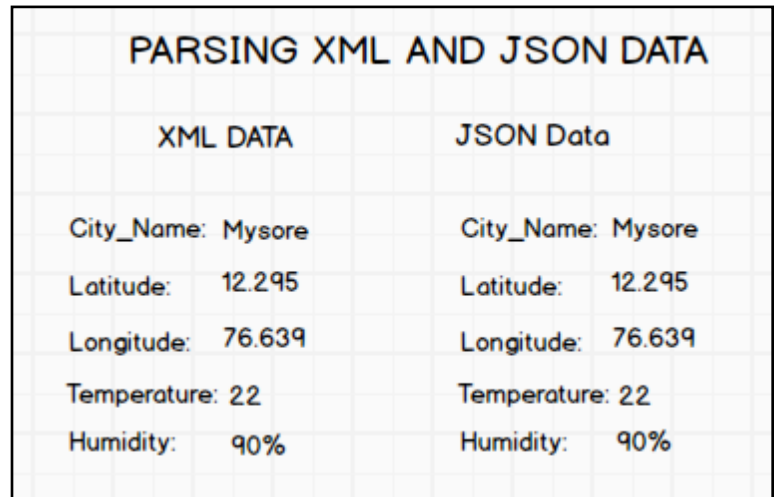
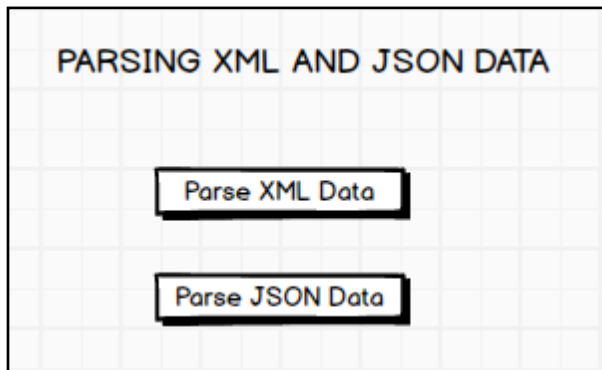
```
Handler handler=new Handler()  
{  
    @SuppressWarnings("HandlerLeak")  
    public void handleMessage(Message m)  
    {  
        counter.setText(String.valueOf(m.what));  
    }  
  
};
```

```
class MyCounter extends Thread  
{  
    public void run()  
    {  
        while(running)  
        {  
            count++; handler.sendMessage(count);  
  
            try { Thread.sleep(1000);  
            }  
            catch(Exception e) { }  
        }  
    }  
}  
  
}
```

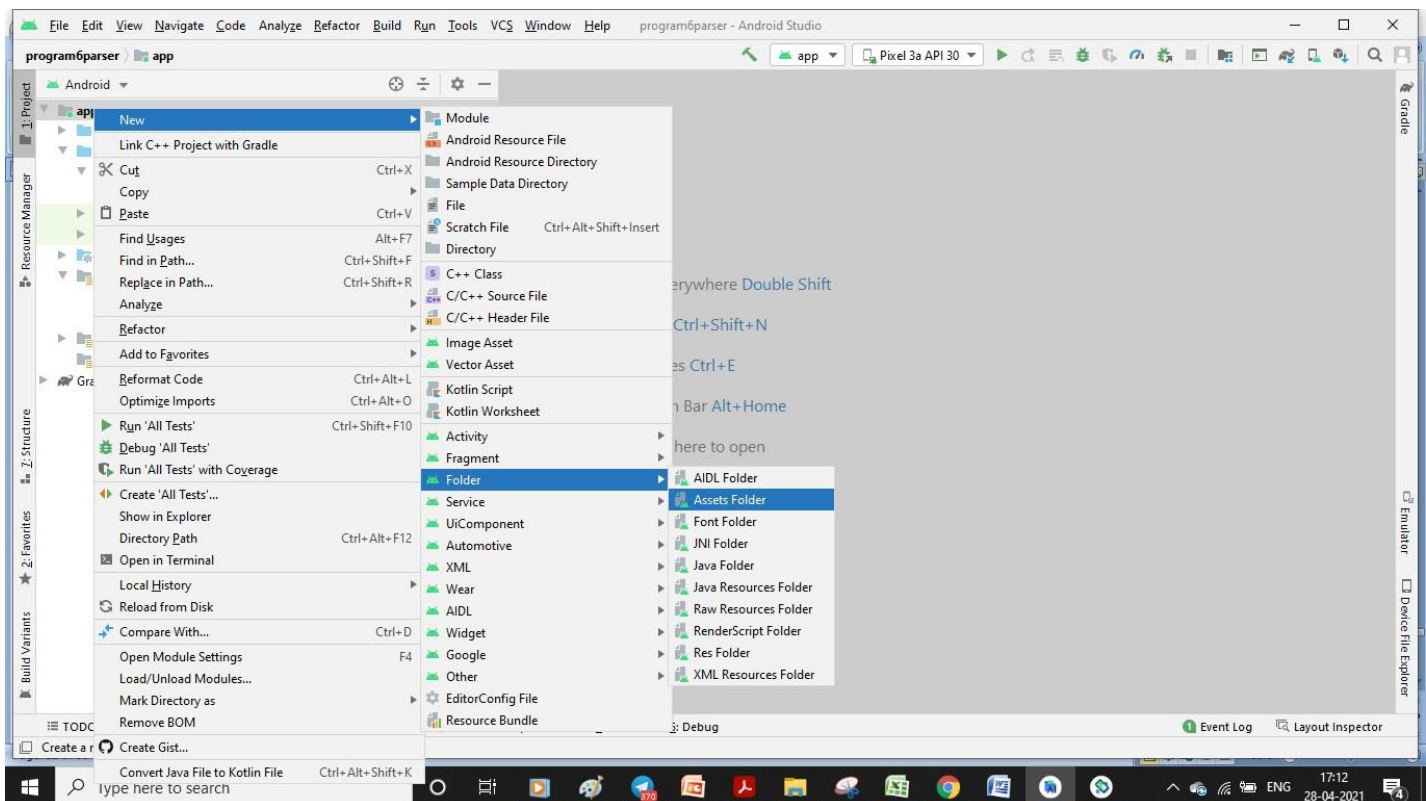

Output:

Program 6:

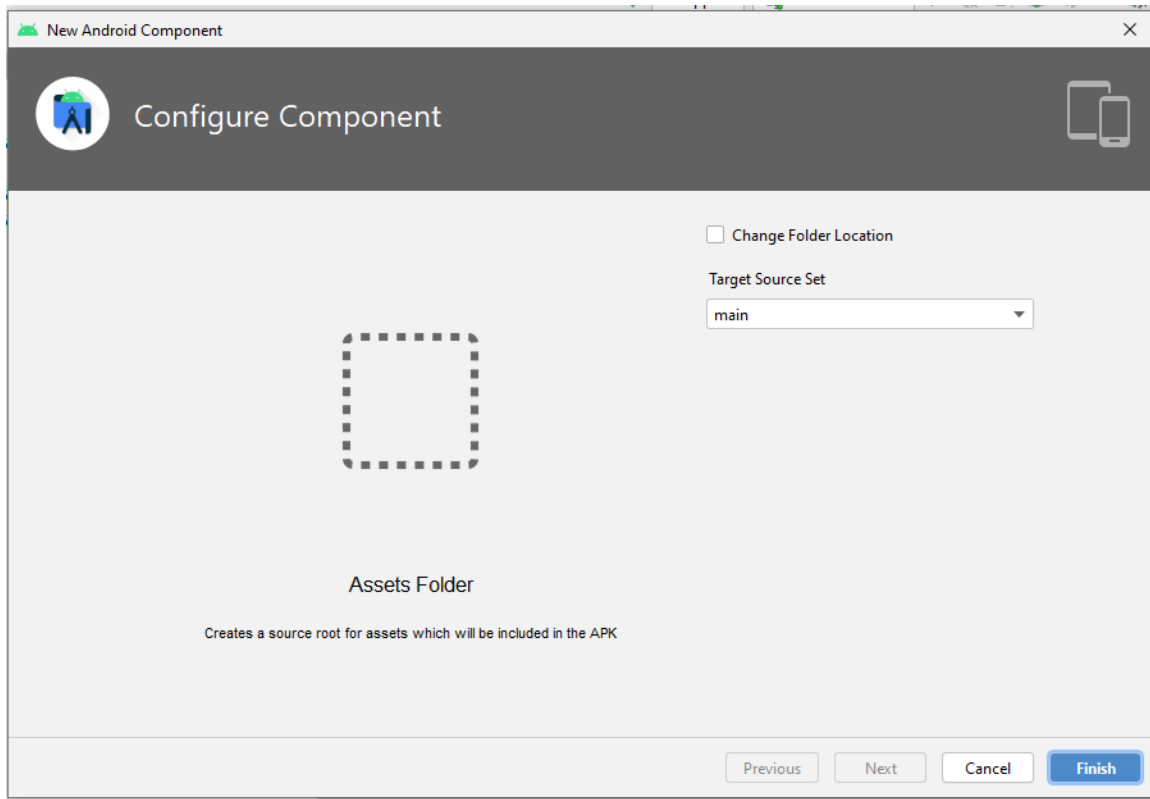
Create two files of XML and JSON type with values for City_Name, Latitude, Longitude, Temperature and Humidity. Develop an application to create an activity with two buttons to parse the XML and JSON files which when clicked should display the data in their respective layouts side by side.

**Step 1: creating xml and json files in asset folder.**

Right click on app folder, and navigate as shown below and click on “assets folder”

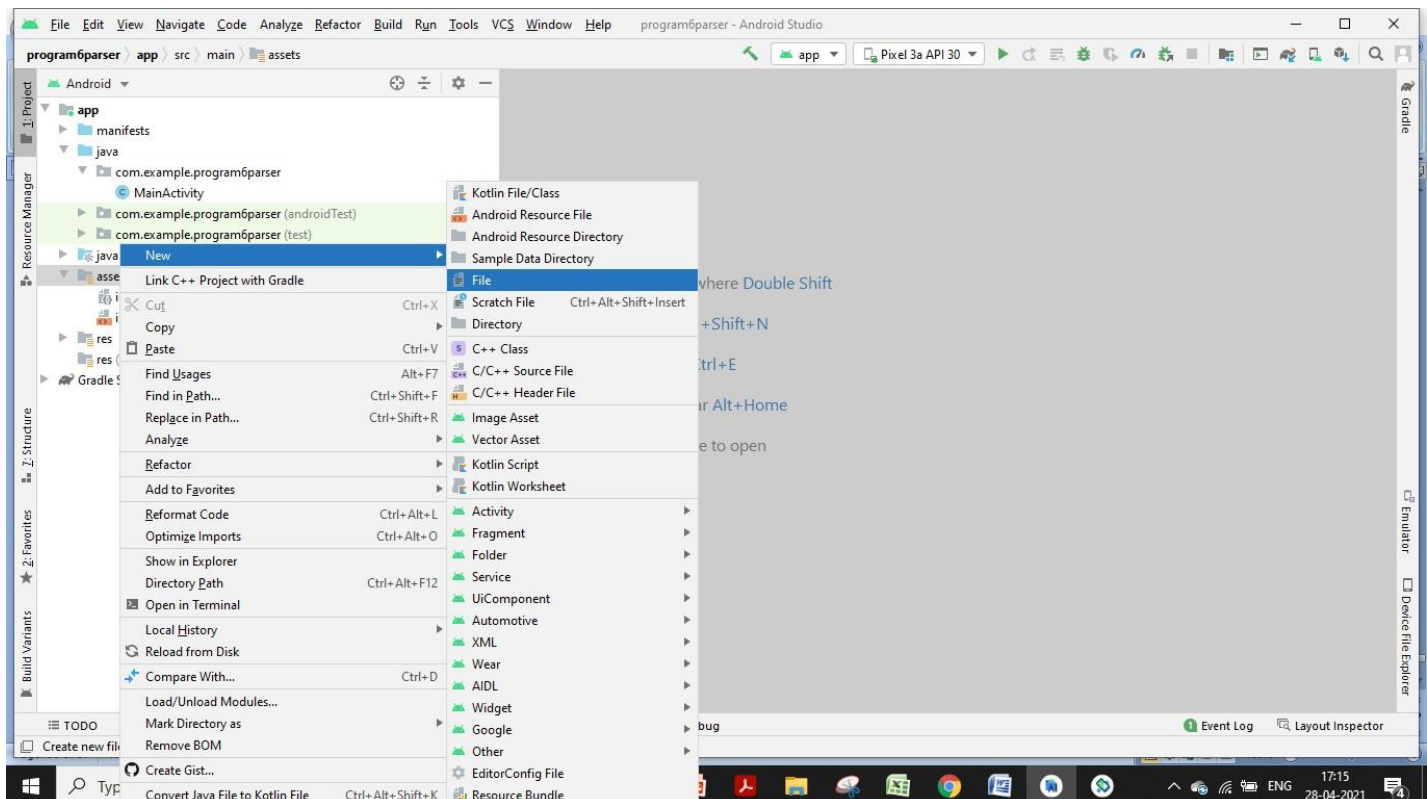


Click finish:

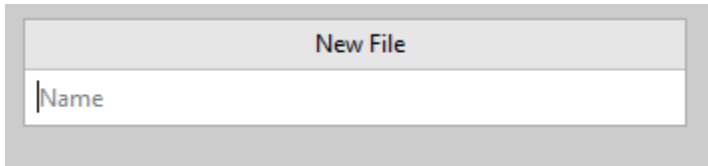


This will create assets folder.

Now to create files inside the assets folder, right click on assets folder and navigate as shown below



Click on file,



Now you can create “input.xml” and “input.json” files.

input.xml

```
<?xml version="1.0"?>
<records>
  <place>
    <city_name>Bengaluru</city_name>
    <Latitude>12.97</Latitude>
    <Longitude>77.59</Longitude>
    <Temperature>30</Temperature>
    <Humidity>63%</Humidity>
  </place>
  <place>
    <city_name>Mysuru</city_name>
    <Latitude>12.29</Latitude>
    <Longitude>76.63</Longitude>
    <Temperature>30</Temperature>
    <Humidity>63%</Humidity>
  </place>
</records>
```

input.json

```
[
  {
    "city_name": "Bengaluru",
    "Latitude": "12.97",
    "Longitude": "77.59",
    "Temperature": 30,
    "Humidity": "63%"
  }
]
```

```
},  
{  
  "city_name": "Mysuru",  
  "Latitude": "12.29",  
  "Longitude": "76.63",  
  "Temperature": 30,  
  "Humidity": "63%"  
}  
]
```

UI Design (activity_main.xml):

```
<?xml version="1.0" encoding="utf-8"?>  
<androidx.constraintlayout.widget.ConstraintLayout  
  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:scrollbars="vertical" tools:context=".MainActivity">  
  
  <TextView android:id="@+id/textView"  
    android:layout_width="348dp"  
    android:layout_height="41dp"  
    android:layout_marginStart="27dp"  
    android:layout_marginLeft="27dp"  
    android:layout_marginEnd="36dp"  
    android:layout_marginRight="36dp"  
    android:gravity="center_horizontal"  
    android:text="PARSING XML AND JSON DATA"  
    android:textSize="24sp"  
    android:textStyle="bold"
```

```
app:layout_constraintBottom_toBottomOf="parent"
app:layout_constraintEnd_toEndOf="parent"
app:layout_constraintStart_toStartOf="parent"
app:layout_constraintTop_toTopOf="parent"
app:layout_constraintVertical_bias="0.062" />
```

<Button

```
android:id="@+id/jsonbutn"
android:layout_width="127dp"
android:layout_height="77dp"
android:layout_marginStart="203dp"
android:layout_marginLeft="203dp"
android:layout_marginTop="154dp"
android:layout_marginEnd="81dp"
android:layout_marginRight="81dp"
android:layout_marginBottom="500dp"
android:text="PARSE JSON"
android:textSize="24sp" android:textStyle="bold"
app:layout_constraintBottom_toBottomOf="parent"
app:layout_constraintEnd_toEndOf="parent"
app:layout_constraintStart_toStartOf="parent"
app:layout_constraintTop_toTopOf="parent" />
```

<Button

```
android:id="@+id/xmlbutn"
android:layout_width="127dp"
android:layout_height="77dp"
android:layout_marginStart="46dp"
android:layout_marginLeft="46dp"
android:layout_marginTop="154dp"
android:layout_marginEnd="238dp"
android:layout_marginRight="238dp"
```

```
android:layout_marginBottom="500dp"
android:text="PARSE XML"
android:textSize="24sp" android:textStyle="bold"
app:layout_constraintBottom_toBottomOf="parent"
app:layout_constraintEnd_toEndOf="parent"
app:layout_constraintStart_toStartOf="parent"
app:layout_constraintTop_toTopOf="parent" />
```

<TextView

```
android:id="@+id/display"
android:layout_width="318dp"
android:layout_height="402dp"
android:layout_marginStart="38dp"
android:layout_marginLeft="38dp"
android:layout_marginTop="270dp"
android:layout_marginEnd="33dp"
android:layout_marginRight="33dp"
android:layout_marginBottom="100dp"
android:scrollbarSize="6dp"
android:scrollbarStyle="outsideInset"
android:scrollbars="vertical"
android:text="data....."
android:textColor="#0E0E0E" android:textSize="20sp"
app:layout_constraintBottom_toBottomOf="parent"
app:layout_constraintEnd_toEndOf="parent"
app:layout_constraintHorizontal_bias="0.363"
app:layout_constraintStart_toStartOf="parent"
app:layout_constraintTop_toTopOf="parent"
app:layout_constraintVertical_bias="0.0" />
```

</androidx.constraintlayout.widget.ConstraintLayout>

MainActivity.java

```
package com.example.program6parser;
import androidx.annotation.RequiresApi;
import androidx.appcompat.app.AppCompatActivity;
import android.content.Intent;
import android.os.Build;
import android.os.Bundle;
import android.text.method.ScrollingMovementMethod;
import android.view.View;
import android.widget.Button;
import android.widget.TextView;
import android.widget.Toast;

import org.json.JSONArray;
import org.json.JSONObject;

import org.w3c.dom.Document;
import org.w3c.dom.Element;
import org.w3c.dom.Node;
import org.w3c.dom.NodeList;

import java.io.InputStream;
import java.nio.charset.StandardCharsets;

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

public class MainActivity extends AppCompatActivity implements View.OnClickListener {
    Button btnxml, btnjson;
    TextView parsed_data;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
```



```
super.onCreate(savedInstanceState);
setContentView(R.layout.activity_main);
btnxml=(Button)findViewById(R.id.xmlbutn);
btnjson=(Button)findViewById(R.id.jsonbutn);
btnxml.setOnClickListener(this);
btnjson.setOnClickListener(this);
parsed_data=(TextView)findViewById(R.id.display);

}

@RequiresApi(api = Build.VERSION_CODES.KITKAT)
@Override
public void onClick(View v) {

    if(v.equals(btnjson))
    {
        parseJson();
    }
    else if(v.equals(btnxml))
    {
        parseXmlDocument();
    }
}
```

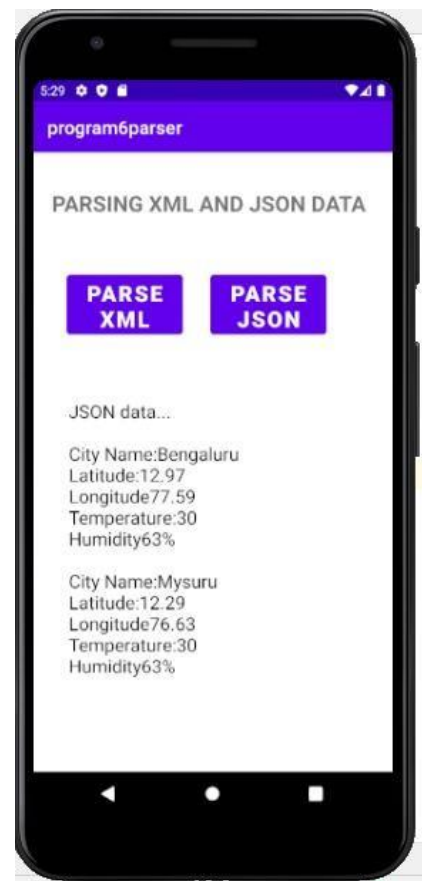
```
public void parseXmlDocument()
{
    try {
        InputStream is = getAssets().open("input.xml");
        DocumentBuilderFactory dbFactory = DocumentBuilderFactory.newInstance();
        DocumentBuilder dBuilder = dbFactory.newDocumentBuilder();
        Document doc = dBuilder.parse(is);
        parsed_data.setMovementMethod(new ScrollingMovementMethod());
        parsed_data.setText("XML data..."+"\n\n");
        Element element=doc.getDocumentElement();
        element.normalize();
        NodeList nList = doc.getElementsByTagName("place");

        for (int i=0; i<nList.getLength(); i++) {
            Node node = nList.item(i);
            if (node.getNodeType() == Node.ELEMENT_NODE) {
                Element element2 = (Element) node;
                parsed_data.append("City Name : " + getValue("city_name", element2)+"\n");
                parsed_data.append("Latitude : " + getValue("Latitude", element2)+"\n");
                parsed_data.append("Longitude : " + getValue("Longitude", element2)+"\n");
                parsed_data.append("Temperature : " + getValue("Temperature", element2)+"\n");
                parsed_data.append("Humidity : " + getValue("Humidity", element2)+"\n\n");
            }
        }
    }
    catch (Exception e) {
        Toast.makeText(MainActivity.this, "Error Parsing XML file", Toast.LENGTH_SHORT).show();
        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();
}

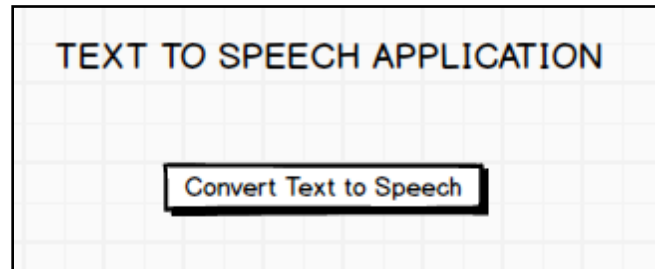
@RequiresApi(api = Build.VERSION_CODES.KITKAT)
public void parseJson()
{
    try {
        InputStream inputStream=getAssets().open("input.json");
        byte[] data=new byte[inputStream.available()];
        inputStream.read(data);
        String jsondata = new String(data,StandardCharsets.UTF_8);
        JSONArray jsonlist = new JSONArray(jsondata);
        parsed_data.setMovementMethod(new ScrollingMovementMethod());
        parsed_data.setText("JSON data..."+"\n\n");

        for (int i = 0; i < jsonlist.length(); i++) {
            JSONObject jObject = jsonlist.getJSONObject(i);
            parsed_data.append("City Name:" + jObject.getString("city_name") + "\n");
            parsed_data.append("Latitude:" + jObject.getString("Latitude") + "\n");
            parsed_data.append("Longitude" + jObject.getString("Longitude") + "\n");
            parsed_data.append("Temperature:" + jObject.getInt("Temperature") + "\n");
            parsed_data.append("Humidity" + jObject.getString("Humidity") + "\n\n");
        }
    }
    catch (Exception e) {
        Toast.makeText(MainActivity.this, "Error Parsing JSON file", Toast.LENGTH_SHORT).show();
        e.printStackTrace();
    }
}
```

Output:

Program 7:

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



Note: Use Emulator API level 28.

UI Design (activity_main.xml):

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
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:id="@+id/hint"
android:layout_width="match_parent"
android:layout_height="match_parent"
tools:context=".MainActivity">

<TextView
    android:id="@+id/textView2"
    android:layout_width="387dp"
    android:layout_height="50dp"
    android:text="TEXT TO SPEECH APPLICATION"
    android:textColor="@color/black"
```

```
android:textSize="26sp"
android:textStyle="bold"
app:layout_constraintBottom_toBottomOf="parent"
app:layout_constraintLeft_toLeftOf="parent"
app:layout_constraintRight_toRightOf="parent"
app:layout_constraintTop_toTopOf="parent"
app:layout_constraintVertical_bias="0.192" />
```

<TextView

```
android:id="@+id/textView"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:text="Enter the text:"
android:textColor="@color/black"
android:textSize="24sp"
android:textStyle="bold"
app:layout_constraintBottom_toBottomOf="parent"
app:layout_constraintEnd_toEndOf="parent"
app:layout_constraintHorizontal_bias="0.134"
app:layout_constraintStart_toStartOf="parent"
app:layout_constraintTop_toTopOf="parent"
app:layout_constraintVertical_bias="0.32" />
```

<EditText android:id="@+id/texttoconv"

```
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:ems="10" android:hint="Enter the
text here..."
android:inputType="textPersonName"
android:textColor="@color/black"
android:textColorHighlight="@color/black"
android:textColorLink="@color/white"
```

```
android:textSize="24sp"
android:textStyle="italic"
app:layout_constraintBottom_toBottomOf="parent"
app:layout_constraintEnd_toEndOf="parent"
app:layout_constraintHorizontal_bias="0.593"
app:layout_constraintStart_toStartOf="parent"
app:layout_constraintTop_toTopOf="parent"
app:layout_constraintVertical_bias="0.423" />
```

<Button

```
android:id="@+id/convert"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:text="Convert Text to Speech"
android:textColor="#FFFFFF"
android:textSize="18sp"
android:textStyle="bold"
app:layout_constraintBottom_toBottomOf="parent"
app:layout_constraintEnd_toEndOf="parent"
app:layout_constraintStart_toStartOf="parent"
app:layout_constraintTop_toTopOf="parent"
app:layout_constraintVertical_bias="0.614" />
```

</androidx.constraintlayout.widget.ConstraintLayout>

MainActivity.java

```
package com.example.program7texttospeech;
```

```
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 implements View.OnClickListener {
```

```
    EditText editText;
```

```
    Button sbutn;
```

```
    TextToSpeech t1;
```

```
@Override
```

```
protected void onCreate(Bundle savedInstanceState) {
```

```
    super.onCreate(savedInstanceState);
```

```
    setContentView(R.layout.activity_main);
```

```
    editText = (EditText) findViewById(R.id.texttoconv);
```

```
    sbutn = (Button) findViewById(R.id.convert);
```

```
    sbutn.setOnClickListener(this);
```

```
    Toast.makeText(getApplicationContext(), "before t1 obj", Toast.LENGTH_LONG).show();
```

```
    t1 = new TextToSpeech(MainActivity.this, new TextToSpeech.OnInitListener() {
```

```
        @Override
```

```
        public void onInit(int status) {
```

```
            if (status != TextToSpeech.ERROR) {
```

```
                Toast.makeText(getApplicationContext(), "SUCCESS", Toast.LENGTH_LONG).show();
```



```
        t1.setLanguage(Locale.ENGLISH);
    }
} //end of onInit
} //end of OnInitListener
);
} //end of onCreate
```

@Override

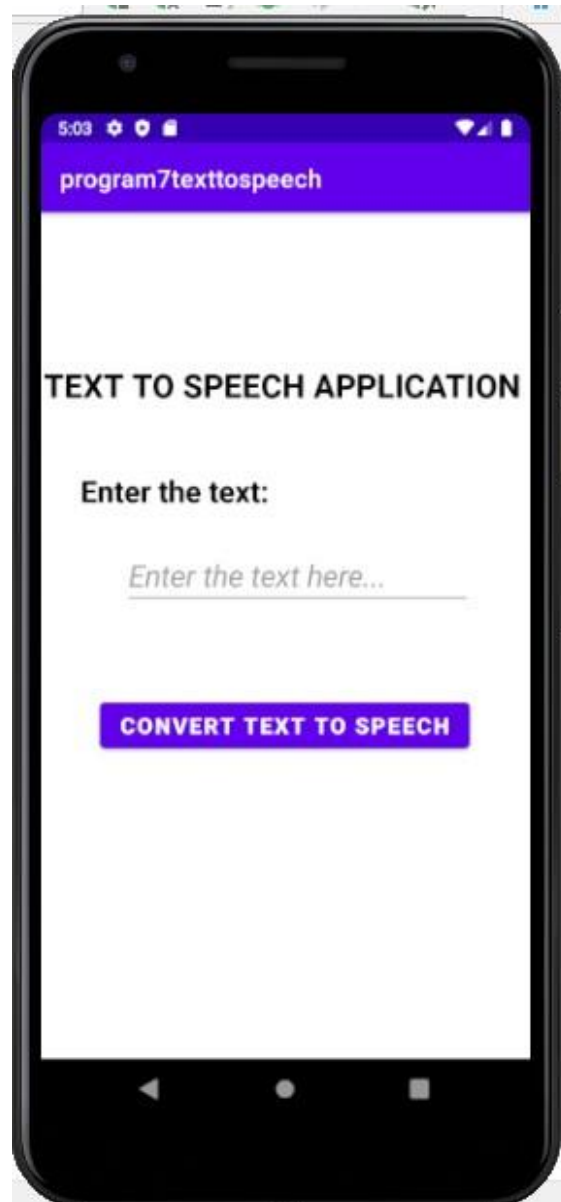
```
public void onClick(View v) {
    Toast.makeText(getApplicationContext(), "inside click", Toast.LENGTH_LONG).show();
    String text = editText.getText().toString();
    t1.speak(text, TextToSpeech.QUEUE_FLUSH, null);
} //end of onCreate
```

@Override

```
protected void onPause() {
    if (t1 != null) {
        t1.stop();
        t1.shutdown();
    }
    super.onPause();
} //end of onPause
```

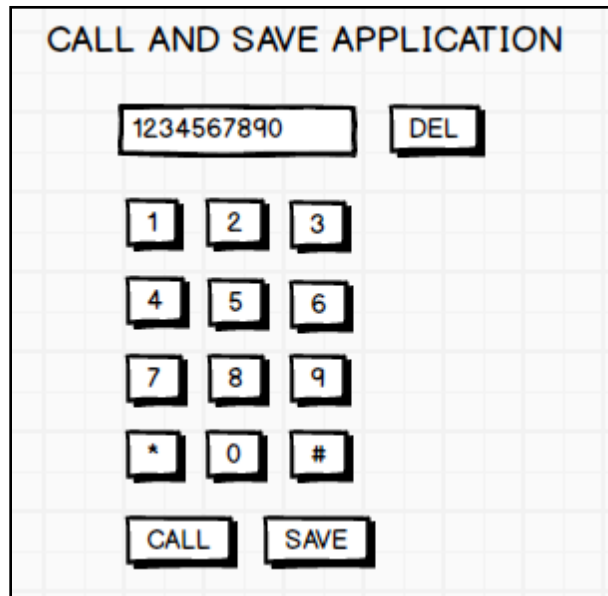
```
} //end of class MainActivity
```

Output:



Program 8:

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

**UI Design (activity_main.xml):**

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
    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:onClick="inputNumber"
    tools:context=".MainActivity">

    <TextView
        android:id="@+id/text1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
```

```
android:background="#03A9F4" android:text="CALL  
AND SAVE APPLICATION"  
android:textColor="@color/black"  
android:textSize="26sp"  
android:textStyle="bold"  
app:layout_constraintBottom_toBottomOf="parent"  
app:layout_constraintHorizontal_bias="0.405"  
app:layout_constraintLeft_toLeftOf="parent"  
app:layout_constraintRight_toRightOf="parent"  
app:layout_constraintTop_toTopOf="parent"  
app:layout_constraintVertical_bias="0.05" />
```

<EditText

```
android:id="@+id/phnum"  
android:layout_width="257dp"  
android:layout_height="49dp"  
android:ems="10"  
android:inputType="textPersonName"  
android:textColor="@color/black"  
android:textSize="24sp"  
app:layout_constraintBottom_toBottomOf="parent"  
app:layout_constraintEnd_toEndOf="parent"  
app:layout_constraintHorizontal_bias="0.103"  
app:layout_constraintStart_toStartOf="parent"  
app:layout_constraintTop_toTopOf="@+id/text1"  
app:layout_constraintVertical_bias="0.108" />
```

<Button

```
android:id="@+id/butndel"  
android:layout_width="80dp"  
android:layout_height="45dp"  
android:text="DEL "  
android:textColor="@color/white"
```

```
android:textSize="20sp"
android:textStyle="bold"
app:layout_constraintBottom_toBottomOf="parent"
app:layout_constraintEnd_toEndOf="parent"
app:layout_constraintHorizontal_bias="0.915"
app:layout_constraintStart_toStartOf="parent"
app:layout_constraintTop_toTopOf="parent"
app:layout_constraintVertical_bias="0.153" />
```

```
<Button android:id="@+id/btn1"
    android:layout_width="63dp"
    android:layout_height="60dp"
    android:onClick="inputNumber"
    android:text="1" android:textSize="24sp"
    android:textStyle="bold"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.071"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintVertical_bias="0.369" />
```

```
<Button android:id="@+id/btn2"
    android:layout_width="63dp"
    android:layout_height="60dp"
    android:onClick="inputNumber"
    android:text="2" android:textSize="24sp"
    android:textStyle="bold"
    app:layout_constraintBottom_toBottomOf="parent"
```

```
app:layout_constraintEnd_toEndOf="parent"
app:layout_constraintHorizontal_bias="0.37"
app:layout_constraintStart_toStartOf="parent"
app:layout_constraintTop_toTopOf="parent"
app:layout_constraintVertical_bias="0.369" />
```

```
<Button android:id="@+id/btn3"
    android:layout_width="63dp"
    android:layout_height="60dp"
    android:onClick="inputNumber"
    android:text="3" android:textSize="24sp"
    android:textStyle="bold"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.66"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintVertical_bias="0.369" />
```

```
<Button android:id="@+id/btn4"
    android:layout_width="63dp"
    android:layout_height="60dp"
    android:onClick="inputNumber"
    android:text="4" android:textSize="24sp"
    android:textStyle="bold"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.071"
    app:layout_constraintStart_toStartOf="parent"
```

```
app:layout_constraintTop_toTopOf="parent"  
app:layout_constraintVertical_bias="0.499" />
```

```
<Button android:id="@+id/btn5"  
    android:layout_width="63dp"  
    android:layout_height="60dp"  
    android:onClick="inputNumber"  
    android:text="5" android:textSize="24sp"  
    android:textStyle="bold"  
    app:layout_constraintBottom_toBottomOf="parent"  
    app:layout_constraintEnd_toEndOf="parent"  
    app:layout_constraintHorizontal_bias="0.37"  
    app:layout_constraintStart_toStartOf="parent"  
    app:layout_constraintTop_toTopOf="parent"  
    app:layout_constraintVertical_bias="0.499" />
```

```
<Button android:id="@+id/btn6"  
    android:layout_width="63dp"  
    android:layout_height="60dp"  
    android:onClick="inputNumber"  
    android:text="6" android:textSize="24sp"  
    android:textStyle="bold"  
    app:layout_constraintBottom_toBottomOf="parent"  
    app:layout_constraintEnd_toEndOf="parent"  
    app:layout_constraintHorizontal_bias="0.66"  
    app:layout_constraintStart_toStartOf="parent"  
    app:layout_constraintTop_toTopOf="parent"  
    app:layout_constraintVertical_bias="0.499" />
```

```
<Button android:id="@+id/btn7"
    android:layout_width="63dp"
    android:layout_height="60dp"
    android:onClick="inputNumber"
    android:text="7" android:textSize="24sp"
    android:textStyle="bold"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.071"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintVertical_bias="0.63" />
```

```
<Button android:id="@+id/btn8"
    android:layout_width="63dp"
    android:layout_height="60dp"
    android:onClick="inputNumber"
    android:text="8" android:textSize="24sp"
    android:textStyle="bold"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.37"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintVertical_bias="0.63" />
```

```
<Button android:id="@+id/btn9"
    android:layout_width="63dp"
```



```
android:layout_height="60dp"
android:onClick="inputNumber"
android:text="9" android:textSize="24sp"
android:textStyle="bold"
app:layout_constraintBottom_toBottomOf="parent"
app:layout_constraintEnd_toEndOf="parent"
app:layout_constraintHorizontal_bias="0.66"
app:layout_constraintStart_toStartOf="parent"
app:layout_constraintTop_toTopOf="parent"
app:layout_constraintVertical_bias="0.63" />
```

```
<Button android:id="@+id/butnstar"
    android:layout_width="63dp"
    android:layout_height="60dp"
    android:onClick="inputNumber"
    android:text="*"
    android:textSize="24sp"
    android:textStyle="bold"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.071"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintVertical_bias="0.773" />
```

```
<Button android:id="@+id/butn0"
    android:layout_width="63dp"
    android:layout_height="60dp"
    android:onClick="inputNumber"
    android:text="0"
```

```
android:textSize="24sp"
android:textStyle="bold"
app:layout_constraintBottom_toBottomOf="parent"
app:layout_constraintEnd_toEndOf="parent"
app:layout_constraintHorizontal_bias="0.37"
app:layout_constraintStart_toStartOf="parent"
app:layout_constraintTop_toTopOf="parent"
app:layout_constraintVertical_bias="0.773" />
```

```
<Button android:id="@+id/butnhash"
    android:layout_width="63dp"
    android:layout_height="60dp"
    android:onClick="inputNumber"
    android:text="#"
    android:textSize="24sp"
    android:textStyle="bold"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.66"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintVertical_bias="0.773" />
```

```
<Button
    android:id="@+id/butncall"
    android:layout_width="140dp"
    android:layout_height="62dp"
    android:text="CALL"
    android:textSize="24sp"
    android:textStyle="bold"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
```

```
app:layout_constraintHorizontal_bias="0.066"
app:layout_constraintStart_toStartOf="parent"
app:layout_constraintTop_toTopOf="parent"
app:layout_constraintVertical_bias="0.922" />

<Button
    android:id="@+id/butnsave"
    android:layout_width="140dp"
    android:layout_height="62dp"
    android:text="SAVE"
    android:textSize="24sp"
    android:textStyle="bold"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.699"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintVertical_bias="0.922" />

</androidx.constraintlayout.widget.ConstraintLayout>
```

MainActivity.java

```
package com.example.program8callsave;

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 implements View.OnClickListener{
```

```
Button btnDel,btnStar,btnHash,btnCall,btnSave;  
EditText txtPhonenumber;  
@Override  
protected void onCreate(Bundle savedInstanceState) {  
    super.onCreate(savedInstanceState);  
    setContentView(R.layout.activity_main);  
  
    btnCall=(Button)findViewById(R.id.butncall);  
    btnCall.setOnClickListener(this);  
  
    btnSave=(Button)findViewById(R.id.butnsave);  
    btnSave.setOnClickListener(this);  
  
    btnDel=(Button)findViewById(R.id.butndel);  
    btnDel.setOnClickListener(this);  
  
    txtPhonenumber=(EditText)findViewById(R.id.phnum);  
    txtPhonenumber.setText('');  
}
```

```
public void onClick(View v)
{
    if(v.equals(btnSave))
    {
        Intent contactIntent= new Intent
        (ContactsContract.Intents.Insert.ACTION);
        contactIntent.setType(ContactsContract.RawContacts.CONTENT_TYPE);
        contactIntent.putExtra(ContactsContract.Intents.Insert.NAME, "Unknown");
        contactIntent.putExtra(ContactsContract.Intents.Insert.PHONE, txtPhonenumber.getText().toString());
        startActivity(contactIntent);
    }

    else if(v.equals(btnDel))
    {
        String data=txtPhonenumber.getText().toString();
        if(data.length()>0)
        {
            txtPhonenumber.setText(data.substring(0,data.length()-1));
        }
        else
        {
            txtPhonenumber.setText("");
        }
    }

    else if(v.equals(btnCall))
    {
        String data = txtPhonenumber.getText().toString();
        Intent intent=new Intent(Intent.ACTION_DIAL);
        intent.setData(Uri.parse("tel:" + data));
        startActivity(intent);
    }
} //end of on click
```

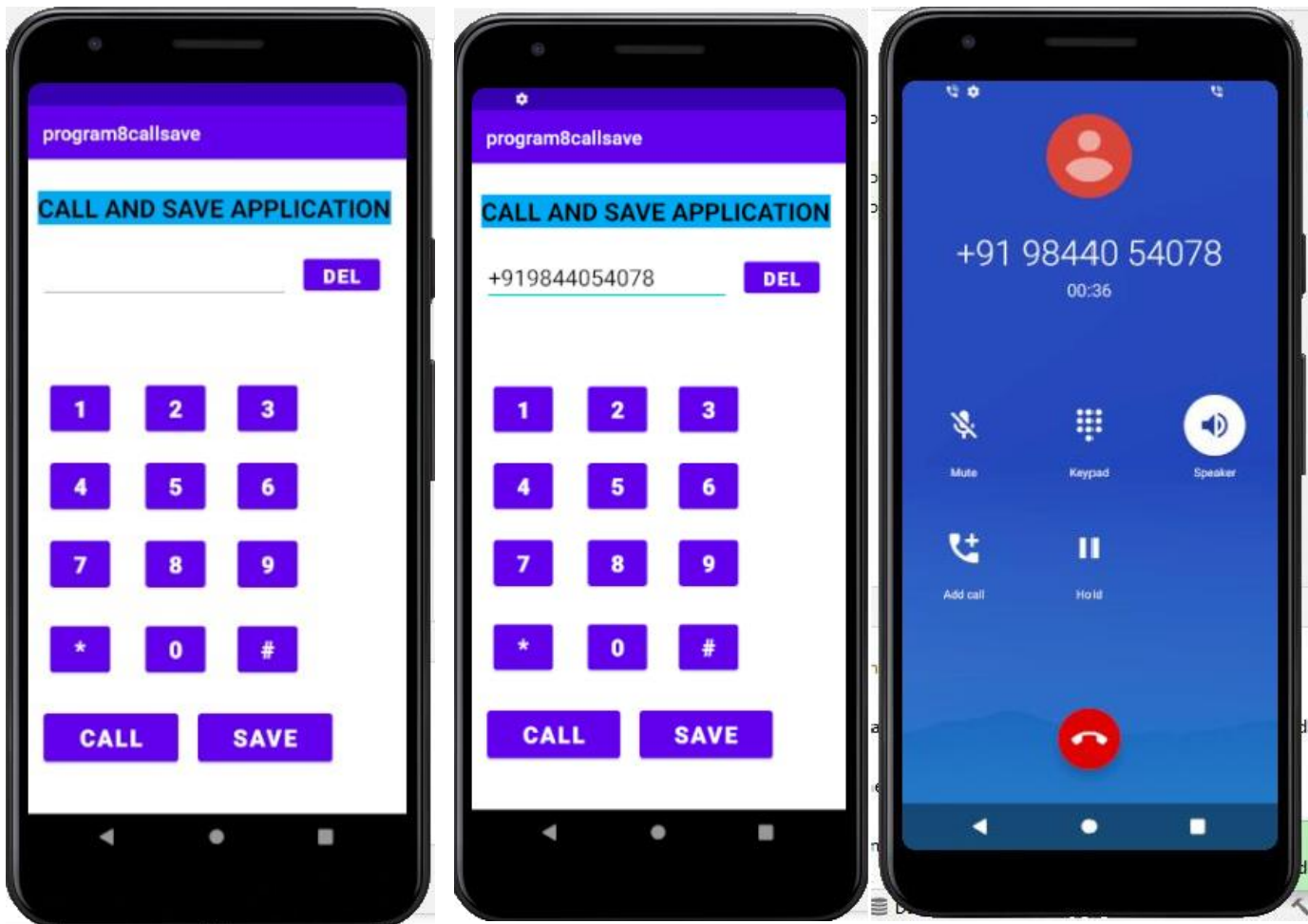
```
public void inputNumber(View V){
    Button btn=(Button)V;
    String digit=btn.getText().toString();
    String phoneNumber=txtPhonenumber.getText().toString();
    txtPhonenumber.setText(phoneNumber +digit);
}

}//end of MainActivity
```

AndroidManifest.xml:

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.example.program8callsave">
    <uses-permission android:name="android.permission.CALL_PHONE"/>
    <application android:allowBackup="true"
        android:icon="@mipmap/ic_launcher"
        android:label="@string/app_name"
        android:roundIcon="@mipmap/ic_launcher_round"
        android:supportsRtl="true"
        android:theme="@style/Theme.Program8callsave">
        <activity android:name=".MainActivity">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />

                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
    </application>
</manifest>
```

Output:

Sample viva questions:

Q1. Android Is Developed By

- a) Apple
- b) Microsoft
- c) Google
- d) Android Inc**

Q2. Android Web Browser Is Based On

- a) Chrome
- b) Open-source Webkit**
- c) Safari
- d) Firefox

Q3. Android Is Based on Which Kernel

- a) Linux**
- b) Windows
- c) Mac
- d) Redhat

Q4. Which Media Format Is Not Supported by Android

- a) MP4
- b) AVI**
- c) MIDI
- d) MPEG

Q5. In Which Directory XML Layout Files Are Stored

- a) /assets
- b) /src
- c) /res/values
- d) /res/layout**

Q6. Which Code Used by Android is not an Open Source.

- A. Video Driver
- B. WiFi Driver**
- C. Device Driver
- D. Bluetooth Driver

Q7. How Many Levels of Securities Are In Android?

- Android Level Security
- App And Kernel Level Security**
- Java Level Security
- None Of The Above

Q8. Which Of The Following Does Not Belong To Transitions?

- A. ViewFlipper
- B. ViewAnimator
- C. ViewSwitcher
- D. ViewSlider**

Q9. What Are The Functionalities In AsyncTask In Android?

- A. OnPreExecution()

B. OnPostExecute()

C. DoInBackground()

D. OnProgressUpdate()

Q10. What Does AAPT Stands For?

a) Android Asset Processing Tool.

b) Android Asset Providing Tool.

c) Android Asset Packaging Tool.

d) Android Asset Packaging Technique

Q11. Which Broadcast In Android Includes Information About Battery State Level

A. Android.intent.action.BATTERY_CHANGED

B. Android.intent.action.BATTERY_LOW

C. Android.intent.action.BATTERY_OKAY

D. Android.intent.action.CALL_BUTTON

Q12. View Pager Is Used For

A. Swiping Activities

B. Swiping Fragments

C. Paging Down List Items

D. View Pager Is Not Supported By Android SDK

Q13. What Is JNI In Android?

a) Java Interface

b) Java Native Interface

c) Java Network Interface

d) Image Editable Tool

Q14. Adb Stands For

a) Android Debug Bridge.

b) Android Drive Bridge.

c) Android Delete Bridge

d) Android Destroy Bridge.

Q15. What Is LastKnownLocation In Android?

A. To Find The Last Location Of A Phone

B. To Find Known Location Of A Phone

C. To Find The Last Known Location Of A Phone.

D. None Of The Above

Q16. Which Programming Language Is Used For Android Application Development?

a) NodeJs

b) PHP

c) JSX

d) Java

Q17. Action Bar Can Be Associated To

A. Only Fragments

B. Only Activities

C. Both Activities And Fragments

D. None Of The Above

Q18. What is Manifest.xml in android?

- a) It has information about layout in an application
- b) It has all the information about an application**
- c) It has the information about activities in an application
- d) None of the above

Q19. In which technique, we can refresh the dynamic content in android?

A. Ajax

B. Java

C. Android

D. None of These

Q20. What is an interface in android?

- a) Interface is a class.
- b) Interface acts as a bridge between class and the outside world.**
- c) Interface is a layout file
- d) All of the above

What is APK in android?

- a) Android pack
- b) Android packages
- c) Android Package Kit**
- d) None of above

Q22. What is the name of the program that converts Java byte code into Dalvik byte code?

- a) Dex compiler**
- b) Dalvik Converter
- c) Android Interpretive Compiler (AIC)
- d) Mobile Interpretive Compiler (MIC)**

Q23. Jelly Bean is an incremental update, with the primary aim of improving the

- A. functionality**
- B. user interface**
- C. performance**
- D. All of the Above**

Q24. What does API stand for?

- a) Application programming interface**
- b) Algorithmic Protocol Interface
- c) Accelerated Programming Interface
- d) None of above

Q25. What is the latest version of android?

- a) Android 8
- b) Android 9
- c) Android 10
- d) Android 11**