MOBILE APPLICATION DEVELOPMENT

(EFFECTIVE FROM THE ACADEMIC YEAR 2021 -2022)

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

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 SQL lite database.
- Inspect different methods of sharing data using services.

Descriptions:

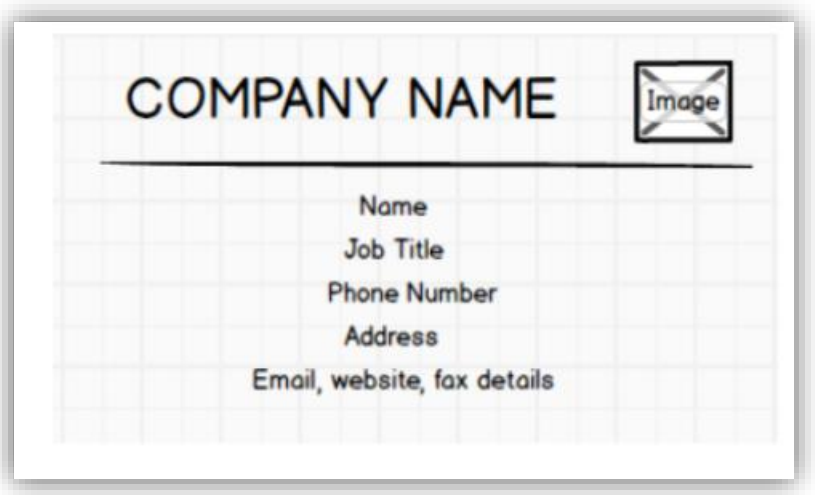
1. The installation procedure of the Android Studio/Java software must be demonstrated and carried out in groups.
2. Students should use the latest version of Android Studio/Java/Kotlin to execute these
3. programs. Diagrams given are for representational purposes only, students are expected to improvise on them.
4. **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).**

GitHub Link: https://github.com/jaideeppoojary/Mobile_Application_Development_Lab

PART – A

1. Create an application to design a Visiting Card. The Visiting card should have a company logo at the top right corner. The company name should be displayed in Capital letters, aligned to the centre. 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.

PROTOTYPE DESIGN



Step 1: Open Android studio, click on 'New Project' and select 'Empty Activity'.

Step 2: Set project name, location, language (Java/ Kotlin) and minimum SDK version.

Step 3: Open activity_main.xml file from Res > Layout folder.

Step 4: Change default root layout to Linear Layout.

Step 5: Create a Relative Layout, add Image and Text View.

Step 6: Use 'View' draw a line.

Step 7: Add Image to drawable folder and use it as Company Icon image.

Step 8: Design the xml to create a Visiting card as given in problem statement.

Step 9: Run the project.

CODE:

activity_main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    tools:context=".MainActivity">
    <RelativeLayout
        android:layout_width="match_parent"
        android:layout_height="120dp">

        <TextView
            android:id="@+id/textView"
            android:layout_width="match_parent"
            android:layout_height="37dp"
            android:layout_marginTop="5dp"
            android:gravity="center"
            android:text="Deep's LTD"
            android:textSize="25dp" />

        <ImageView
            android:id="@+id/imageView"
            android:layout_width="120dp"
            android:layout_height="100dp"
            android:layout_alignParentRight="true"
            android:layout_marginLeft="270dp"
            android:layout_marginTop="10dp"
            android:layout_marginRight="5dp"
            app:srcCompat="@drawable/jlogo" />
    </RelativeLayout>

    <View
        android:id="@+id/view2"
        android:layout_width="match_parent"
        android:layout_height="1dp"
        android:background="@color/black" />

    <TextView
        android:id="@+id/textView3"
        android:layout_width="match_parent"
        android:layout_height="32dp"
        android:layout_marginTop="15dp"
        android:gravity="center"
        android:text="Name: Jaideep"
        android:textSize="22dp" />

    <TextView
        android:id="@+id/textView5"
        android:layout_width="match_parent"
```

```
        android:layout_height="32dp"
        android:layout_marginTop="15dp"
        android:gravity="center"
        android:text="Job: Devoleper"
        android:textSize="22dp" />

    <View
        android:id="@+id/view3"
        android:layout_width="250dp"
        android:layout_height="1dp"
        android:layout_gravity="center"
        android:background="@color/black" />

    <TextView
        android:id="@+id/textView6"
        android:layout_width="match_parent"
        android:layout_height="32dp"
        android:layout_marginTop="15dp"
        android:gravity="center"
        android:text="Phone: 9777777777"
        android:textSize="22dp" />
    <TextView
        android:id="@+id/textView7"
        android:layout_width="match_parent"
        android:layout_height="32dp"
        android:layout_marginTop="15dp"
        android:gravity="center"
        android:text="Address: Udupi"
        android:textSize="22dp" />

    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:orientation="horizontal">
        <TextView
            android:id="@+id/textView8"
            android:layout_width="match_parent"
            android:layout_height="32dp"
            android:layout_marginTop="15dp"
            android:layout_weight="1"
            android:gravity="center"
            android:text="Email: Jaideep@deeps.com"
            android:textSize="12dp" />
        <TextView
            android:id="@+id/textView9"
            android:layout_width="match_parent"
            android:layout_height="32dp"
            android:layout_marginTop="15dp"
            android:layout_weight="1"
            android:gravity="center"
            android:text="Website: www.Jaideep.com"
            android:textSize="12dp" />
    </LinearLayout>
</LinearLayout>
```

MainActivity.java

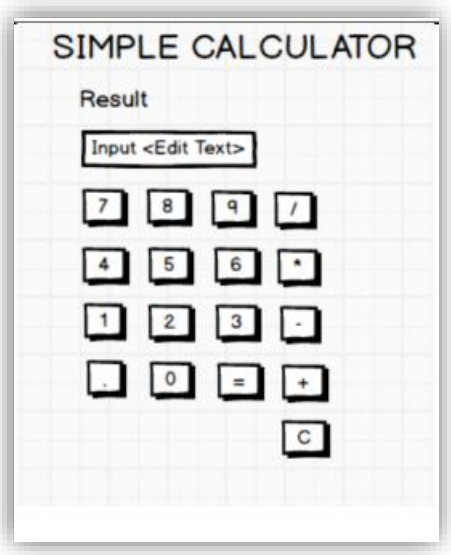
```java
package com.example.visitingcard;
import androidx.appcompat.app.AppCompatActivity;
import android.os.Bundle;

public class MainActivity extends AppCompatActivity {

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

Observed Output:

GitHub Link: https://github.com/jaideeppoojary/Mobile_Application_Development_Lab

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

PROTOTYPE DESIGN



Step 1: Open Android studio, click on 'New Project' and select 'Empty Activity'.

Step 2: Set project name, location, language (Java/ Kotlin) and minimum SDK version.

Step 3: Open activity_main.xml file from Res > Layout folder.

Step 4: Change default root layout to Constraint Layout.

Step 5: Add Edit View to accept input numbers for calculation from user.

Step 6: Add four buttons for Addition, Subtraction, Multiplication and Division Operation.

Step 7: Add Button logic in MainActivity.java to do calculation and display output.

Step 8: Run the project.

CODE:

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/textView"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Simple Calculator"
        android:textSize="28dp"
        android:layout_marginTop="20dp"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent" />

    <EditText
        android:id="@+id/editTextNumber"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginTop="20dp"
        android:ems="10"
        android:inputType="number"
        android:minHeight="48dp"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/textView" />

    <EditText
        android:id="@+id/editTextNumber2"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginTop="20dp"
        android:ems="10"
        android:inputType="number"
        android:minHeight="48dp"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.502"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/editTextNumber" />
    <EditText
        android:id="@+id/editTextNumberResult"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginTop="30dp"
        android:ems="10"
        android:inputType="number"
        android:minHeight="48dp"
        app:layout_constraintEnd_toEndOf="parent"
```

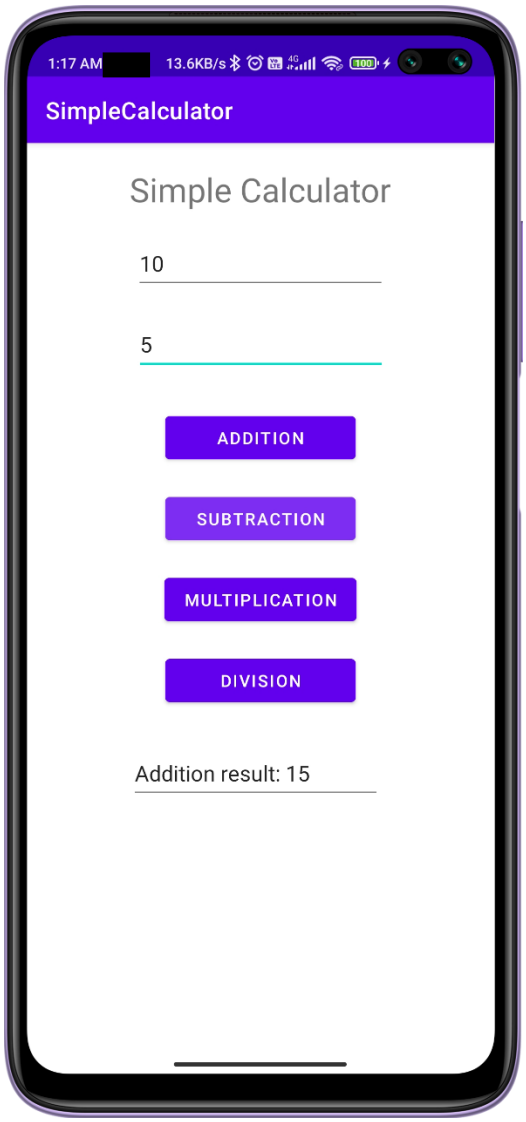
```
        app:layout_constraintHorizontal_bias="0.477"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/buttonToDivide" />

    <Button
        android:id="@+id/buttonToAdd"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginTop="30dp"
        android:onClick="Add"
        android:text="Addition"
        android:minWidth="160dp"
        app:layout_constraintHorizontal_bias="0.5"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/editTextNumber2" />
    <Button
        android:id="@+id/buttonToSub"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginTop="20dp"
        android:onClick="Sub"
        android:text="Subtraction"
        android:minWidth="160dp"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.5"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/buttonToAdd" />

    <Button
        android:id="@+id/buttonToMul"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginTop="20dp"
        android:onClick="Mul"
        android:text="Multiplication"
        android:minWidth="160dp"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.5"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/buttonToSub" />

    <Button
        android:id="@+id/buttonToDivide"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginTop="20dp"
        android:onClick="Div"
        android:text="Division"
        android:minWidth="160dp"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.5"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/buttonToMul" />

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

MainActivity.java

```
package com.example.simplecalculator;
import androidx.appcompat.app.AppCompatActivity;
import android.os.Bundle;
import android.view.View;
import android.widget.EditText;

public class MainActivity extends AppCompatActivity {
    private EditText et1;
    private EditText et2;
    private EditText et3;

    public void InitValues(){
        et1 = (EditText) findViewById(R.id.editTextNumber);
        et2 = (EditText) findViewById(R.id.editTextNumber2);
        et3 = (EditText) findViewById(R.id.editTextNumberResult);
    }
    public void Add(View v){
        int n1 = Integer.parseInt(et1.getText().toString());
        int n2 = Integer.parseInt(et2.getText().toString());

        int result = n1 + n2;
        et3.setText("Addition result: " + result);
    }
    public void Sub(View v){
        int n1 = Integer.parseInt(et1.getText().toString());
        int n2 = Integer.parseInt(et2.getText().toString());

        int result = n1 - n2;
        et3.setText("Subtraction result: " + result);
    }
    public void Mul(View v){
        int n1 = Integer.parseInt(et1.getText().toString());
        int n2 = Integer.parseInt(et2.getText().toString());

        int result = n1 * n2;
        et3.setText("Multiplication result: " + result);
    }
    public void Div(View v){
        int n1 = Integer.parseInt(et1.getText().toString());
        int n2 = Integer.parseInt(et2.getText().toString());
        double result;
        if( n2!=0){
            result = n1/n2;
            et3.setText("Division result: " + result);
        }
        else{
            et3.setText("Zero Division Not allowed!!");
```

```
        }

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

//      Initializing values before calling any functions
        InitValues();
    }
}
```

Observed Output:



GitHub Link: https://github.com/jaideeppoojary/Mobile_Application_Development_Lab

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.

Step 1: Open Android studio, click on 'New Project' and select 'Empty Activity'.

Step 2: Set project name, location, language (Java/ Kotlin) and minimum SDK version.

Step 3: Open activity_main.xml file from Res > Layout folder.

Step 4: Create Signup layout using drag and drop framework.

Step 5: Create one more 'Empty Activity' called 'Login Activity'.

Step 6: Open activity_login.xml file from Layout folder and make root layout as Constraint Layout.

Step 7: Create Login layout using drag and drop framework.

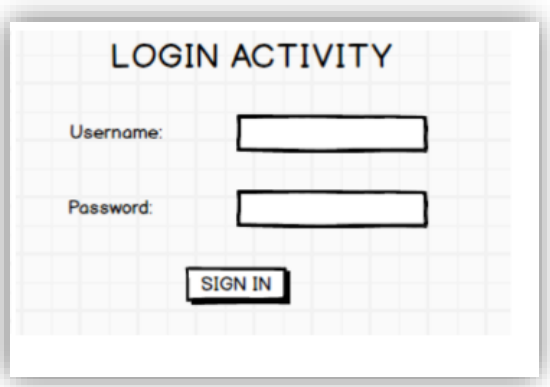
Step 8: Add button click listener to button and use Intent to change Activity.

Step 9: Use Regular Expression –

"^(?=.*[A-Z]) (?=.*[a-z]) (?=.*\\d) (?=.*[*&@$+])(?=.*[A-Za-z0-9@$#*]){8, }$"

Step 8: Run the project.

PROTOTYPE DESIGN



CODE:

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/textView"
        android:layout_width="250dp"
        android:layout_height="53dp"
        android:layout_marginTop="88dp"
        android:text="Sign Up Page"
        android:textAlignment="center"
        android:textColor="#1C1B1B"
        android:textSize="34sp"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.5"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent" />

    <EditText
        android:id="@+id/signupUserPassword"
        android:layout_width="226dp"
        android:layout_height="50dp"
        android:layout_marginTop="44dp"
        android:ems="10"
        android:hint="Password"
        android:inputType="textPassword"
        android:textSize="20sp"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.491"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/signupUserName" />
    <Button
        android:id="@+id/signupButton"
        android:layout_width="228dp"
        android:layout_height="53dp"
        android:layout_marginTop="40dp"
        android:text="Sign Up"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/signupUserPassword" />
```

```
    <EditText
        android:id="@+id/signupUserName"
        android:layout_width="225dp"
        android:layout_height="50dp"
        android:layout_marginTop="120dp"
        android:ems="10"
        android:hint="User Name"
        android:inputType="textPersonName"
        android:textSize="20sp"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/textView" />
</androidx.constraintlayout.widget.ConstraintLayout>
```

Activity_signin.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=".Signin">
    <TextView
        android:id="@+id/textView"
        android:layout_width="250dp"
        android:layout_height="53dp"
        android:layout_marginTop="88dp"
        android:text="Login Page"
        android:textAlignment="center"
        android:textColor="#1C1B1B"
        android:textSize="34sp"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.496"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent" />

    <EditText
        android:id="@+id/loginUserName"
        android:layout_width="225dp"
        android:layout_height="50dp"
        android:layout_marginTop="120dp"
        android:ems="10"
        android:hint="User Name"
        android:inputType="textPersonName"
        android:textSize="20sp"
```

```
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintHorizontal_bias="0.497"
        app:layout_constraintTop_toBottomOf="@+id/textView" />


    <EditText
        android:id="@+id/loginUserPassword"
        android:layout_width="226dp"
        android:layout_height="50dp"
        android:layout_marginTop="44dp"
        android:ems="10"
        android:hint="Password"
        android:inputType="textPassword"
        android:textSize="20sp"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.5"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/loginUserName" />

    <Button
        android:id="@+id/loginButton"
        android:layout_width="228dp"
        android:layout_height="53dp"
        android:layout_marginTop="40dp"
        android:text="Login"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/loginUserPassword" />

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

MainActivity.java

```
package com.example.signuploginproject;

import androidx.appcompat.app.AppCompatActivity;

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

import java.util.regex.Pattern;

public class MainActivity extends AppCompatActivity {
```

```
    public EditText signupUserName;
    public EditText signupUserPassword;
    public Button signupButton;
    public String regularExpression;
    public void myInit(){
        signupUserName = (EditText) findViewById(R.id.signupUserName);
        signupUserPassword = (EditText) findViewById(R.id.signupUserPassword);
        signupButton = (Button) findViewById(R.id.signupButton);

        regularExpression = "^(?=.*[a-z])(?=.*[A-Z])(?=.*\\d)(?=.*[@$!%*?&])[A-Za-
z\\d@$!%*?&]{8,19}$";
    }

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

        myInit();

        signupButton.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                String userName = signupUserName.getText().toString();
                String userPassword = signupUserPassword.getText().toString();

                if(validatePassword(userPassword)){
                    //like wrapping values
                    Bundle bundle = new Bundle();
                    bundle.putString("username", userName);
                    bundle.putString("password", userPassword);

                    Intent intent = new Intent(MainActivity.this, Signin.class);
                    intent.putExtras(bundle);

                    startActivity(intent);
                }
                else{
                    Toast.makeText(MainActivity.this, "Invalid Password",
Toast.LENGTH_SHORT).show();
                }
            }
        });
    }

    public  Boolean validatePassword(String pass){
        Pattern pattern = Pattern.compile(regularExpression);
```

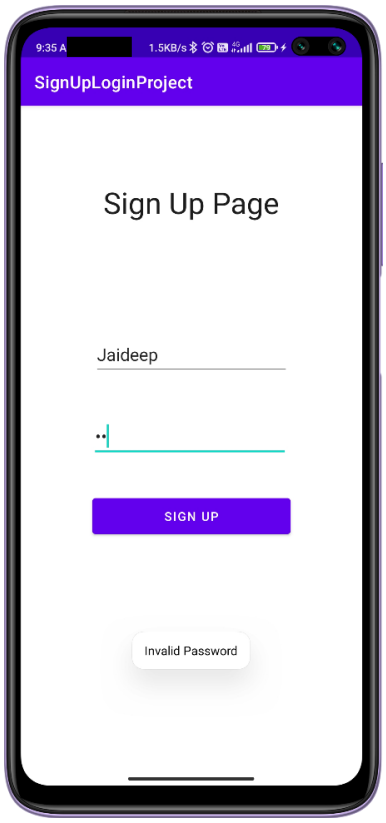
```
        if(pattern.matcher(pass).matches())
            return true;
        else
            return  false;
    }
}
```

Signin.java

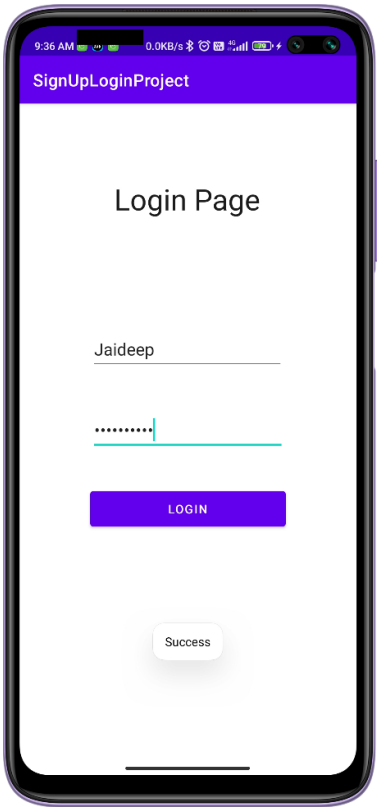
```
package com.example.signuploginproject;

import androidx.appcompat.app.AppCompatActivity;

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

public class Signin extends AppCompatActivity {

    public EditText loginUserName;
    public EditText loginUserPass;
    public Button loginButton;
    int count=0;

    public void myInit(){
        loginUserName = (EditText) findViewById(R.id.loginUserName);
        loginUserPass = (EditText) findViewById(R.id.loginUserPassword);
        loginButton = (Button) findViewById(R.id.loginButton);
    }

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

        myInit();
        Bundle bundle = getIntent().getExtras();
        String id = bundle.getString("username");
        String pass = bundle.getString("password");
```

```
        loginButton.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                String user = loginUserName.getText().toString();
                String password = loginUserPass.getText().toString();

                if( id.equals(user) && pass.equals(password)){
                    Toast.makeText(Signin.this, "Success",
Toast.LENGTH_SHORT).show();
                }
                else{
                    count++;

                    if( count>=3){
                        loginButton.setEnabled(false);
                        Toast.makeText(Signin.this, "Failed Login in Attempts",
Toast.LENGTH_SHORT).show();
                    }
                    Toast.makeText(Signin.this, "Failed",
Toast.LENGTH_SHORT).show();
                }
            }
        });


    }
}
```

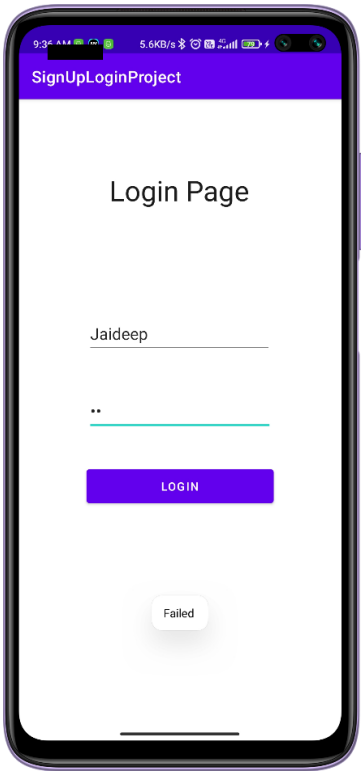
Observed Output:



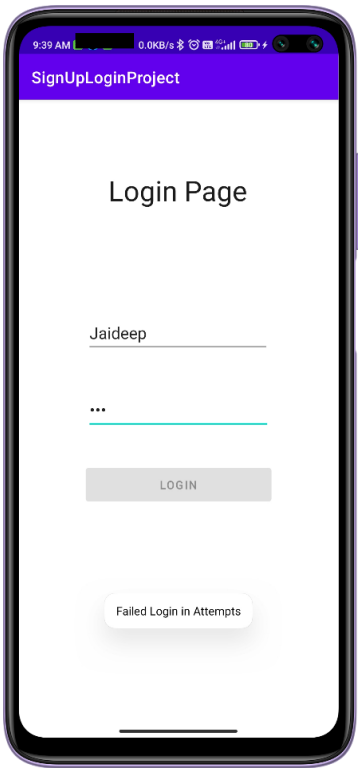
Signup page- If password pattern not followed.



Login page – On successful login.



Login page – If password is wrong.



Login page – Three times wrong credential.

GitHub Link: https://github.com/jaideeppoojary/Mobile_Application_Development_Lab

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.

Step 1: Open Android studio, click on 'New Project' and select 'Empty Activity'.

Step 2: Set project name, location, language (Java/ Kotlin) and minimum SDK version.

Step 3: Open activity_main.xml file from Res > Layout folder.

Step 4: Change default root layout to Linear Layout.

Step 5: Add three or more images to drawable folder.

Step 6: Mention the permission used in AndroidManifest.xml.

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

Step 7: Use TimerTask class to change wallpaper on every 30 seconds time interval.

Step 8: Initialize and use WallpaperManager.setBitmap() method to change wallpaper.

Step 9: Run the project.

PROTOTYPE DESIGN

CODE:

activity_main.xml

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

    <TextView
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_marginTop="80dp"
```

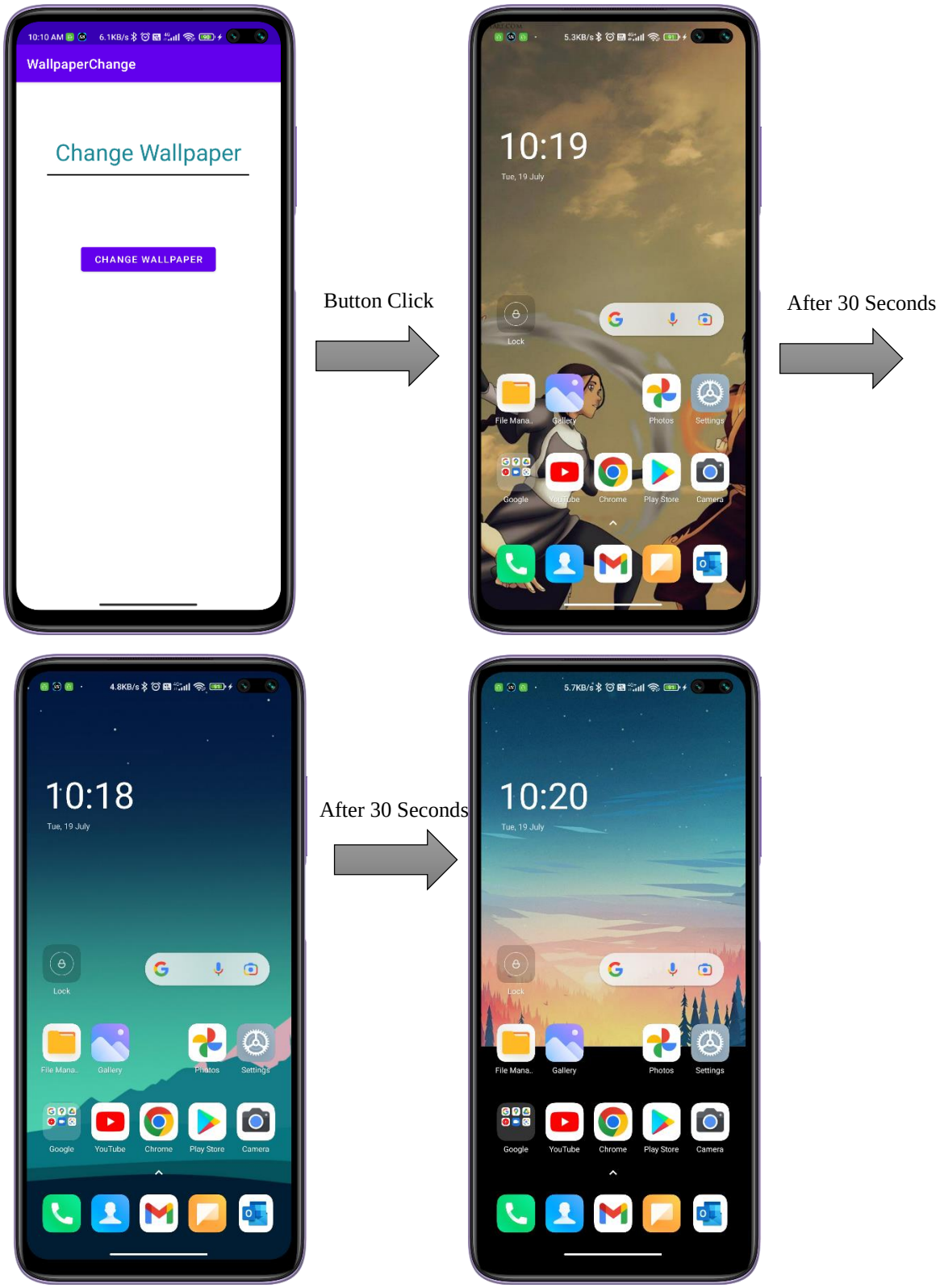
```
        android:gravity="center"
        android:text="Change Wallpaper"
        android:textColor="#1F8B98"
        android:textSize="34sp"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintLeft_toLeftOf="parent"
        app:layout_constraintRight_toRightOf="parent"
        app:layout_constraintTop_toTopOf="parent" />

    <View
        android:id="@+id/view2"
        android:layout_width="300dp"
        android:layout_height="2dp"
        android:layout_gravity="center"
        android:layout_marginTop="10dp"
        android:background="@color/black" />

    <Button
        android:id="@+id/wallpaperButton"
        android:layout_width="200dp"
        android:layout_height="wrap_content"
        android:layout_gravity="center"
        android:layout_marginTop="100dp"
        android:text="change Wallpaper" />

</LinearLayout>
```

MainActivity.java

```
package com.example.wallpaperchange;

import androidx.appcompat.app.AppCompatActivity;

import android.app.WallpaperManager;
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 {

    public Button changeButton;
    public int images[] ;
    public int i;
```

```
    public  void myInit(){
        images= new int[]{
                R.drawable.img1,
                R.drawable.img2,
                R.drawable.img3,
                R.drawable.img4,
                R.drawable.img5,
                R.drawable.img6,
        };
        changeButton = (Button) findViewById(R.id.wallpaperButton);
        i=0;
    }
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        myInit();
        changeButton.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                new Timer().schedule(new ChangeWallpaper(), 0, 30000);
            }
        });


    }
    class ChangeWallpaper extends TimerTask{

        @Override
        public void run() {
            WallpaperManager wallpaperManager =
WallpaperManager.getInstance(getBaseContext());

            try{
                wallpaperManager.setBitmap(
BitmapFactory.decodeResource(getResources(), images[i]) );
                i++;
                if( i>= images.length)
                    i=0;
            }
            catch (IOException ex){
                ex.printStackTrace();
            }
        }
    }

}
```

Observed Output:



GitHub Link: https://github.com/jaideeppoojary/Mobile_Application_Development_Lab

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.

Step 1: Open Android studio, click on 'New Project' and select 'Empty Activity'.

Step 2: Set project name, location, language (Java/ Kotlin) and minimum SDK version.

Step 3: Open activity_main.xml file from Res > Layout folder.

Step 4: Create layout which contains title called 'Counter'.

Step 5: Add 'Start' and 'Stop' button to start and stop counter. And a TextView to display counter value.

Step 6: Write logic and java code to have a counter. (Hint: Use Thread class).

Step 7: Use TimerTask class to change wallpaper on every 30 seconds time interval.

Step 8: Run the project.

PROTOTYPE DESIGN

CODE:

activity_main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="#B1EAF1"
    android:orientation="vertical"
    tools:context=".MainActivity">

    <TextView
```

```
        android:id="@+id/textView"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_marginTop="80dp"
        android:gravity="center"
        android:text="Counter Application"
        android:textColor="#000000"
        android:textSize="28sp" />

    <View
        android:id="@+id/view"
        android:layout_width="250dp"
        android:layout_height="4dp"
        android:layout_gravity="center"
        android:background="#35515E" />

    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_marginTop="80dp"
        android:orientation="horizontal">

        <TextView
            android:id="@+id/textView5"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:layout_weight="1"
            android:gravity="center"
            android:text="Counter :"
            android:textColor="#000000"
            android:textSize="20sp" />

        <TextView
            android:id="@+id/textViewCounter"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:layout_weight="1"
            android:gravity="center"
            android:text="1"
            android:textColor="#000000"
            android:textSize="20sp" />
    </LinearLayout>

    <Switch
        android:id="@+id/switchToReverse"
        android:layout_width="150dp"
        android:layout_height="50dp"
        android:layout_gravity="center"
```

```
        android:layout_marginTop="20dp"
        android:text="Reverse"
        android:textSize="16sp" />

    <Button
        android:id="@+id/buttonToStart"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_gravity="center"
        android:layout_marginTop="30dp"
        android:backgroundTint="#F34CAF50"
        android:minWidth="180dp"
        android:text="Start" />

    <Button
        android:id="@+id/buttonToStop"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_gravity="center"
        android:layout_marginTop="20dp"
        android:backgroundTint="#F44336"
        android:minWidth="180dp"
        android:text="Stop" />

</LinearLayout>
```

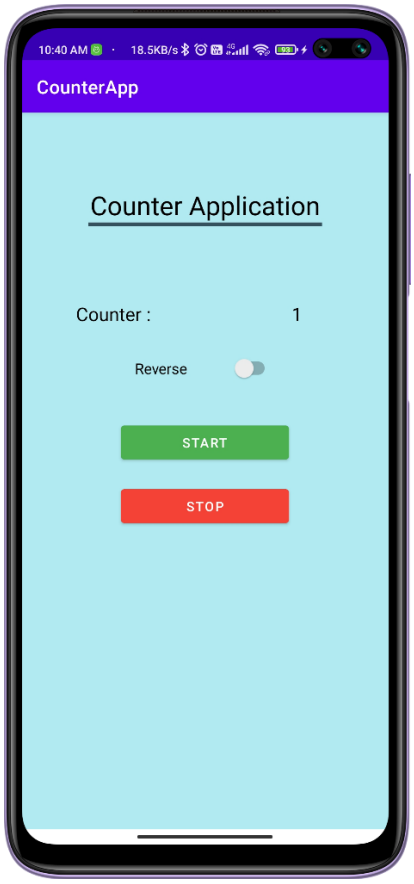
MainActivity.java

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

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

public class MainActivity extends AppCompatActivity {

    private static final String TAG = "Thread";
    Button start;
    Button stop;
    TextView counterText;
    Switch reverseSwitch;
    int counter;
    int counterStep;
```

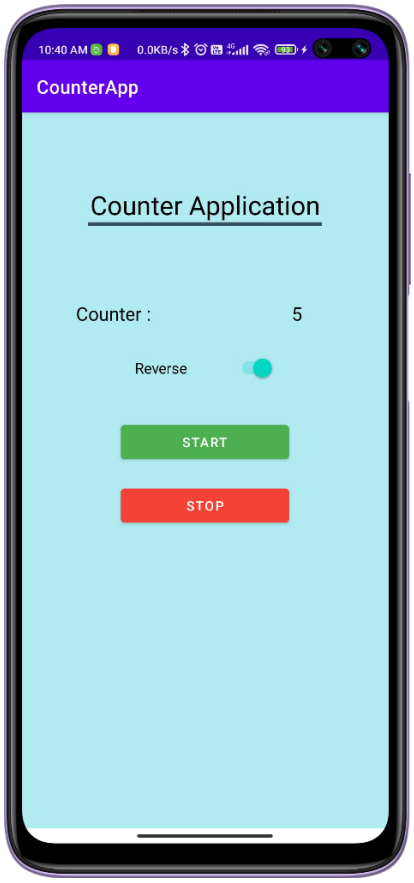
```
    boolean running = true;

    Handler mainHandler = new Handler();


    public void myInit(){
        start= (Button) findViewById(R.id.buttonToStart);
        stop= (Button) findViewById(R.id.buttonToStop);
        counterText= (TextView) findViewById(R.id.textViewCounter);
        reverseSwitch = (Switch) findViewById(R.id.switchToReverse);
        counter = 1;
        counterStep =1;
    }
    public void startThread(){
        myThread obj = new myThread();
        obj.start();
    }

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

        myInit();

        start.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                counter =0;
                running = true;
                startThread();
            }
        });
        stop.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                running = false;
            }
        });
    }
    class myThread extends  Thread {
        @Override
        public  void run(){
            while(running){
                counterStep=1;
                if(reverseSwitch.isChecked())
                    counterStep=-1;
                counter+=counterStep;
```

```
                mainHandler.post(new Runnable() {
                    @Override
                    public void run() {
                        counterText.setText(counter+"");
                    }
                });

                try {
                    Thread.sleep(1000);
                } catch (InterruptedException e) {
                    e.printStackTrace();
                }

            }
        }
    }

}
```

Observed Output:



Incremental Counter



Decremental Counter

GitHub Link: https://github.com/jaideeppoojary/Mobile_Application_Development_Lab

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: Open Android studio, click on 'New Project' and select 'Empty Activity'.

Step 2: Set project name, location, language (Java/ Kotlin) and minimum SDK version.

Step 3: Open activity_main.xml file from Res > Layout folder.

Step 4: Create layout which contains Two buttons one for parsing XML data and another for JSON data.

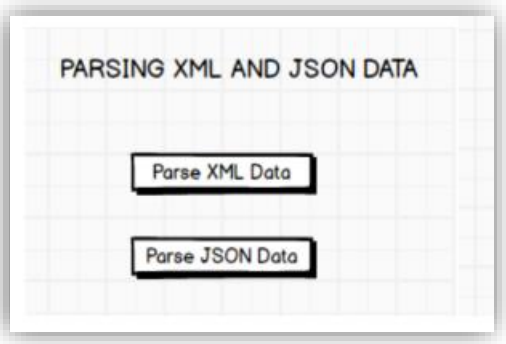
Step 5: Create a 'Empty Activity' and design layout for displaying parsed data.

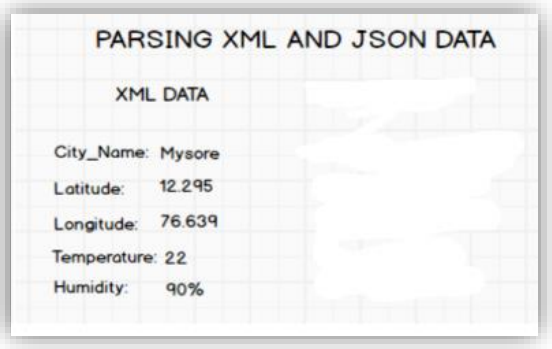
Step 6: Create a 'Assets' folder and add XML, and JSON file.

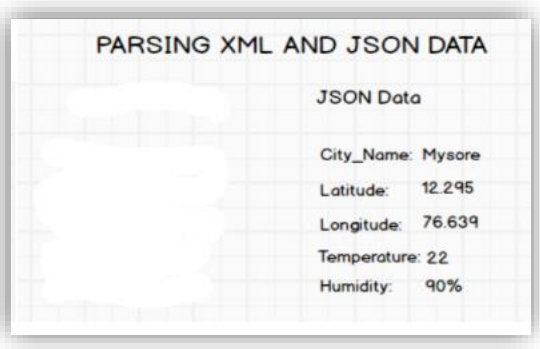
Step 7: Write Java code to parse and display data.

Step 8: Run the project.

PROTOTYPE DESIGN







ASSETS FOLDER FILES:

City.xml

```
<?xml version="1.0" ?>
<Records>
    <city id="1">
        <name>Udupi</name>
        <longitude>25.99554</longitude>
        <latitude>-85.569821</latitude>
        <temparature>34</temparature>
        <humidity>90%</humidity>
    </city>
</Records>
```

City.json

```
{
    "city": {
      "name": "Udupi",
      "logitute": 25.99554,
      "latitude": -85.569821,
      "temparature": 34,
      "humidity": "90%"
    }
  }
```

CODE:

activity_main.xml

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

    <TextView
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_marginTop="80dp"
        android:gravity="center"
        android:text="Data Parsing"
        android:textSize="34sp"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintLeft_toLeftOf="parent"
```

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

    <View
        android:id="@+id/view3"
        android:layout_width="300dp"
        android:layout_height="3dp"
        android:layout_gravity="center_horizontal"
        android:background="#2D393E" />

    <Button
        android:id="@+id/buttonXml"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_gravity="center_horizontal"
        android:layout_marginTop="50dp"
        android:text="Parse XML Data" />

    <Button
        android:id="@+id/buttonJson"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_gravity="center_horizontal"
        android:layout_marginTop="50dp"
        android:text="Parse JSON Data" />

</LinearLayout>
```

activity_view.xml

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


    <TextView
        android:id="@+id/textView"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_marginTop="50dp"
        android:gravity="center_horizontal"
        android:text="Parsing XML And JSON Data"
        android:textSize="24sp" />
```

```
    <TextView
        android:id="@+id/textViewTitle"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_marginTop="50dp"
        android:gravity="center_horizontal"
        android:text="XML Data"
        android:textSize="24sp" />

    <TextView
        android:id="@+id/textViewParseData"
        android:layout_width="337dp"
        android:layout_height="310dp"
        android:layout_marginStart="30dp"
        android:layout_marginLeft="50dp"
        android:layout_marginTop="73dp"
        android:gravity="center_horizontal"
        android:text="Values"
        android:textSize="18sp" />

</LinearLayout>
```

MainActivity.java

```
package com.example.dataparsing;

import androidx.appcompat.app.AppCompatActivity;

import android.content.Intent;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;

public class MainActivity extends AppCompatActivity {

    Button xmlParseBtn;
    Button jsonParseBtn;

    public void myInit(){
        xmlParseBtn = (Button) findViewById(R.id.buttonXml);
        jsonParseBtn = (Button) findViewById(R.id.buttonJson);
    }
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        myInit();
```

```
        xmlParseBtn.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                Intent intent = new Intent(MainActivity.this, ViewActivity.class);
                intent.putExtra("mode", 1);
                startActivity(intent);
            }
        });

        jsonParseBtn.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                Intent intent = new Intent(MainActivity.this, ViewActivity.class);
                intent.putExtra("mode", 2);
                startActivity(intent);
            }
        });
    }
}
```

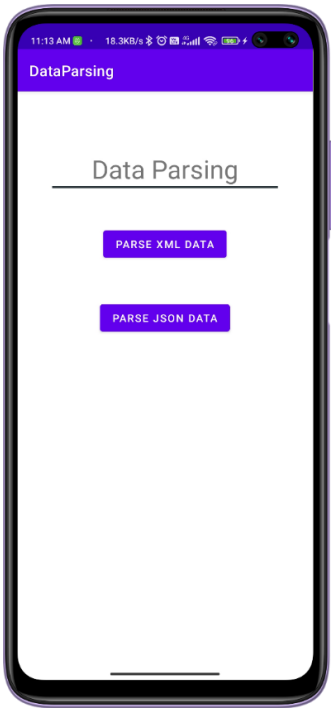
ViewActivity.java

```
package com.example.dataparsing;

import androidx.appcompat.app.AppCompatActivity;

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

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

public class ViewActivity extends AppCompatActivity {

    TextView dataDisplayView, parseDataTitle;
    int mode=0;

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

```
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_view);

        dataDisplayView = (TextView) findViewById(R.id.textViewParseData);
        parseDataTitle = (TextView) findViewById(R.id.textViewTitle);

        mode = getIntent().getIntExtra("mode",0);
        if(mode==1)
            ParseXMLDocument();
        else
            ParseJSON();
    }

    private void ParseJSON() {
        parseDataTitle.setText("JSON Data");
        try{
            InputStream inputStream=getAssets().open("city.json");
            byte[] data=new byte[inputStream.available()];
            inputStream.read(data);
            String readData=new String(data);

           JSONObject jsonObject=new JSONObject(readData);
           JSONObject jsonCityDetails=jsonObject.getJSONObject("city");

 dataDisplayView.setText("City Name :"+ jsonCityDetails.getString("name")+"\n");
 dataDisplayView.append("Temperature :"+ jsonCityDetails.getInt("temparature")+
"\n");
 dataDisplayView.append("Longitude :"+ jsonCityDetails.getString("logitute")+
"\n");
 dataDisplayView.append("Latitude :"+ jsonCityDetails.getString("latitude")+"\n");
 dataDisplayView.append("Humidity :"+ jsonCityDetails.getString("humidity")+"\n");
        }
        catch (Exception e){
            e.printStackTrace();
        }
    }

    private String ParseXMLDocument() {
        parseDataTitle.setText("XML Data");
        try{
            InputStream inputStream = getAssets().open("city.xml");
          DocumentBuilderFactory dbFactory =DocumentBuilderFactory.newInstance();
          DocumentBuilder dBuilder=dbFactory.newDocumentBuilder();
          Document doc=dBuilder.parse(inputStream);
          Element element=doc.getDocumentElement();
            element.normalize();
            NodeList nList =doc.getElementsByTagName("city");
            for(int i=0;i<nList.getLength();i++)
```

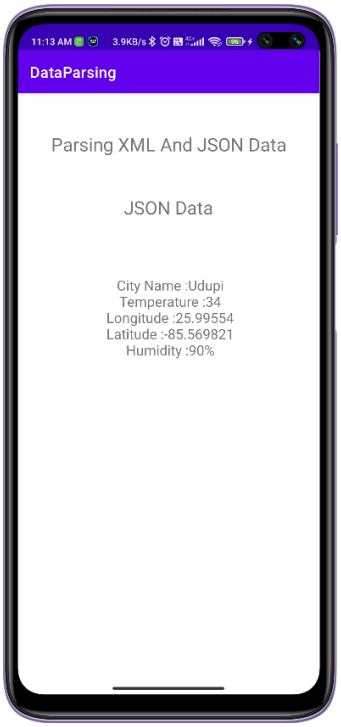
```
            {
                Node node=nList.item(i);
    if(node.getNodeType()==Node.ELEMENT_NODE){
        Element element2=(Element) node;
    dataDisplayView.setText("City Name :"+ getValue("name",element2)+"\n");
    dataDisplayView.append("Temperature :"+ getValue("temparature",element2)+"\n");
    dataDisplayView.append("Longitude :"+ getValue("longitude",element2)+"\n");
    dataDisplayView.append("Latitude :"+ getValue("latitude",element2)+"\n");
    dataDisplayView.append("Humidity :"+ getValue("humidity",element2)+"\n");
                }
            }
        }
        catch(Exception e) {
            e.printStackTrace();
        }
        return null;
    }

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

Observed Output:



Choose which data to parse.

Shows XML data parsed.



Shows JSON data parsed.

GitHub Link: https://github.com/jaideeppoojary/Mobile_Application_Development_Lab

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.

Step 1: Open Android studio, click on 'New Project' and select 'Empty Activity'.

Step 2: Set project name, location, language (Java/ Kotlin) and minimum SDK version.

Step 3: Open activity_main.xml file from Res > Layout folder.

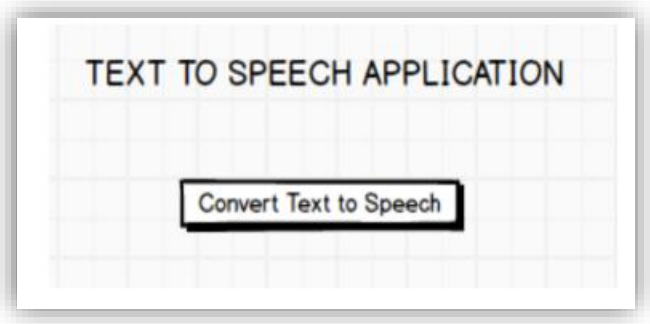
Step 4: Create layout which contains One EditText and a Button.

Step 5: On click of button read text in EditText.

Step 6: Use Android Text to Speech API to convert text to speech.

Step 7: Run the project.

PROTOTYPE DESIGN



CODE:

activity_main.xml

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

    <EditText
        android:id="@+id/editTextTextPersonName"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_gravity="center"
        android:ems="10"
```

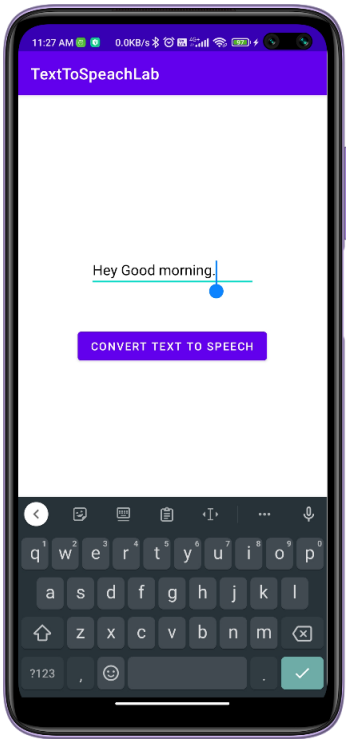
```
        android:hint="Enter the text to Speech"
        android:layout_marginTop="200dp"
        android:textColor="@color/black"
        android:inputType="textPersonName" />

    <Button
        android:id="@+id/button"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_gravity="center"
        android:layout_marginTop="50dp"
        android:text="Convert Text To Speech" />
</LinearLayout>
```

MainActivity.java

```
package com.example.texttospeachlab;

import androidx.appcompat.app.AppCompatActivity;

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

import java.util.Locale;

public class MainActivity extends AppCompatActivity {

    EditText enteredValue;
    Button speakBtn;

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

        speakBtn = findViewById(R.id.button);
        enteredValue = findViewById(R.id.editTextTextPersonName);


        textToSpeech = new TextToSpeech(MainActivity.this, new
TextToSpeech.OnInitListener() {
            @Override
            public void onInit(int status) {
```

```
                if(status == TextToSpeech.SUCCESS){
                    textToSpeech.setLanguage(Locale.ENGLISH);
                }
                else{
                    Log.e("failed", "onInit: Failed");
                }

            }
        });

        speakBtn.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                String text = enteredValue.getText().toString();

                textToSpeech.speak(text,TextToSpeech.QUEUE_FLUSH,null);

            }
        });



    }
}
```

Observed Output:



GitHub Link: https://github.com/jaideeppoojary/Mobile_Application_Development_Lab

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.

Step 1: Open Android studio, click on 'New Project' and select 'Empty Activity'.

Step 2: Set project name, location, language (Java/ Kotlin) and minimum SDK version.

Step 3: Open activity_main.xml file from Res > Layout folder.

Step 4: Create layout with buttons and EditText which looks like a dialler.

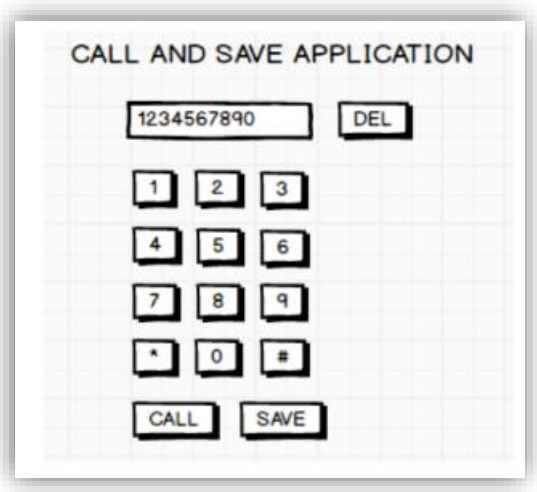
Step 5: Use 'Intent.ACTION_DIAL' to make call.

Step 6: Use 'ContactsContract.Intents.Insert.ACTION' to save the contact.

Step 7: Write logic and java code to match problem statement.

Step 7: Run the project.

PROTOTYPE DESIGN



CODE:

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"
```

```
    tools:ignore="MissingClass">

    <TextView
        android:id="@+id/textView"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Phone Dialer"
        android:textAlignment="center"
        android:textAllCaps="true"
        android:textColor="#FF1744"
        android:textSize="24sp"
        android:textStyle="bold"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintHorizontal_bias="0.443"
        app:layout_constraintLeft_toLeftOf="parent"
        app:layout_constraintRight_toRightOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        app:layout_constraintVertical_bias="0.022" />

    <EditText
        android:id="@+id/editText"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:ems="10"
        android:inputType="textPersonName"
        android:text="Enter Text"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.159"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        app:layout_constraintVertical_bias="0.09"
        tools:ignore="TouchTargetSizeCheck" />

    <Button
        android:id="@+id/btn1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="1"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.1"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        app:layout_constraintVertical_bias="0.206" />

    <Button
        android:id="@+id/btn4"
```

```
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="4"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.1"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        app:layout_constraintVertical_bias="0.323" />

    <Button
        android:id="@+id/btn7"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="7"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.1"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        app:layout_constraintVertical_bias="0.446" />

    <Button
        android:id="@+id/btnStar"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="*"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.1"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        app:layout_constraintVertical_bias="0.559" />

    <Button
        android:id="@+id/btnCall"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Call"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.1"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        app:layout_constraintVertical_bias="0.68" />

    <Button
        android:id="@+id/btn2"
```

```
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="2"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.498"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        app:layout_constraintVertical_bias="0.206" />

    <Button
        android:id="@+id/btn5"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="5"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.476"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        app:layout_constraintVertical_bias="0.323" />


    <Button
        android:id="@+id/btn8"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="8"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.498"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        app:layout_constraintVertical_bias="0.446" />

    <Button
        android:id="@+id/btn0"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="0"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.498"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        app:layout_constraintVertical_bias="0.559" />

    <Button
        android:id="@+id/btnDelete"
```

```
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Delete"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.848"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        app:layout_constraintVertical_bias="0.09" />

    <Button
        android:id="@+id/btn3"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="3"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.848"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        app:layout_constraintVertical_bias="0.206" />

    <Button
        android:id="@+id/btn6"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="6"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.848"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        app:layout_constraintVertical_bias="0.323" />

    <Button
        android:id="@+id/btn9"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="9"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.848"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        app:layout_constraintVertical_bias="0.446" />

    <Button
        android:id="@+id/btnHash"
```

```
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="#"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.848"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        app:layout_constraintVertical_bias="0.559" />

    <Button
        android:id="@+id/btnSave"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Save"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.848"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        app:layout_constraintVertical_bias="0.68" />

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

MainActivity.java

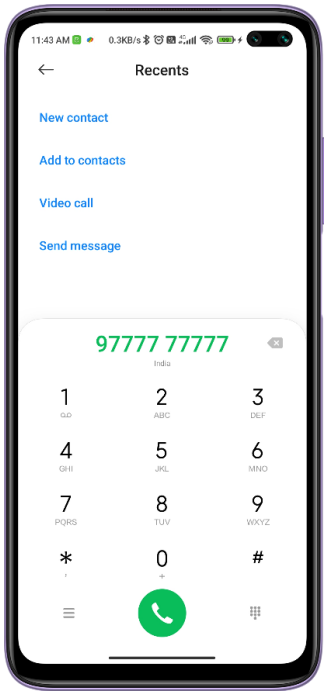
```
package com.example.phoneapp;

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 btn1, btn2, btn3, btn4, btn5, btn6, btn7, btn8, btn9, btn0;
    Button btndel, btnstar, btnhash, btncall, btnsave;
    EditText txtPhonenumber;

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

```
        btn1=(Button) findViewById(R.id.btn1);
        btn1.setOnClickListener(this);

        btn0=(Button) findViewById(R.id.btn0);
        btn0.setOnClickListener(this);

        btn2=(Button) findViewById(R.id.btn2);
        btn2.setOnClickListener(this);

        btn3=(Button) findViewById(R.id.btn3);
        btn3.setOnClickListener(this);

        btn4=(Button) findViewById(R.id.btn4);
        btn4.setOnClickListener(this);

        btn5=(Button) findViewById(R.id.btn5);
        btn5.setOnClickListener(this);

        btn6=(Button) findViewById(R.id.btn6);
        btn6.setOnClickListener(this);

        btn7=(Button) findViewById(R.id.btn7);
        btn7.setOnClickListener(this);

        btn8=(Button) findViewById(R.id.btn8);
        btn8.setOnClickListener(this);

        btn9=(Button) findViewById(R.id.btn9);
        btn9.setOnClickListener(this);

        btnhash=(Button) findViewById(R.id.btnHash);
        btnhash.setOnClickListener(this);

        btnstar=(Button) findViewById(R.id.btnStar);
        btnstar.setOnClickListener(this);

        btnsave=(Button) findViewById(R.id.btnSave);
        btnsave.setOnClickListener(this);

        btncall=(Button) findViewById(R.id.btnCall);
        btncall.setOnClickListener(this);

        btndel=(Button) findViewById(R.id.btnDelete);
        btndel.setOnClickListener(this);

        txtPhonenumber = (EditText) findViewById(R.id.editText);
        txtPhonenumber.setText("");
    }
```

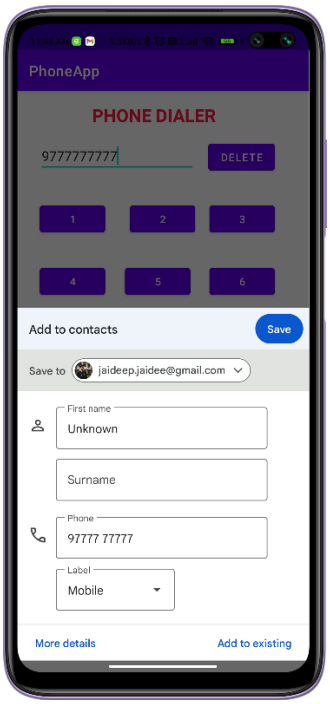
```
    @Override
    public void onClick(View v) {
        if(v.equals(btn1)){
            txtPhonenumber.append("1");
        }
        else if(v.equals(btn2))txtPhonenumber.append("2");
        else if(v.equals(btn3))txtPhonenumber.append("3");
        else if(v.equals(btn4))txtPhonenumber.append("4");
        else if(v.equals(btn5))txtPhonenumber.append("5");
        else if(v.equals(btn6))txtPhonenumber.append("6");
        else if(v.equals(btn7))txtPhonenumber.append("7");
        else if(v.equals(btn8))txtPhonenumber.append("8");
        else if(v.equals(btn9))txtPhonenumber.append("9");
        else if(v.equals(btn0))txtPhonenumber.append("0");
        else if(v.equals(btnstar))txtPhonenumber.append("*");
        else if(v.equals(btnhash))txtPhonenumber.append("#");
        else
      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,txtPhonenum
ber.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("");
        }
        btncall.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                String data=txtPhonenumber.getText().toString();
                Intent intent=new Intent(Intent.ACTION_DIAL);
                intent.setData(Uri.parse("tel:"+data));
                startActivity(intent);
            }
        });
    }
}
```

Observed Output:

Dialler to type phone number.



On click CALL button.



On click SAVE button.

GitHub Link: https://github.com/jaideeppoojary/Mobile_Application_Development_Lab

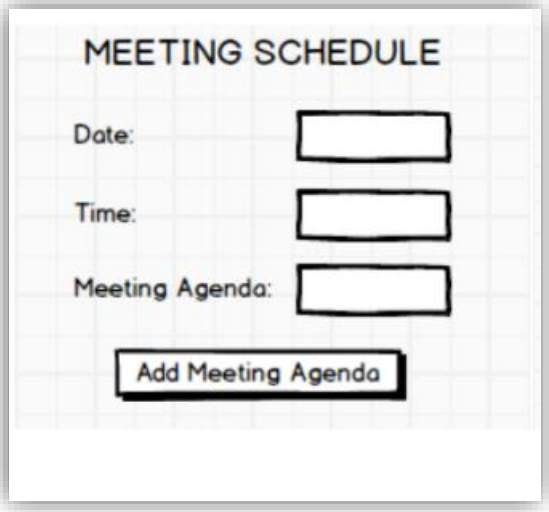
PART – B

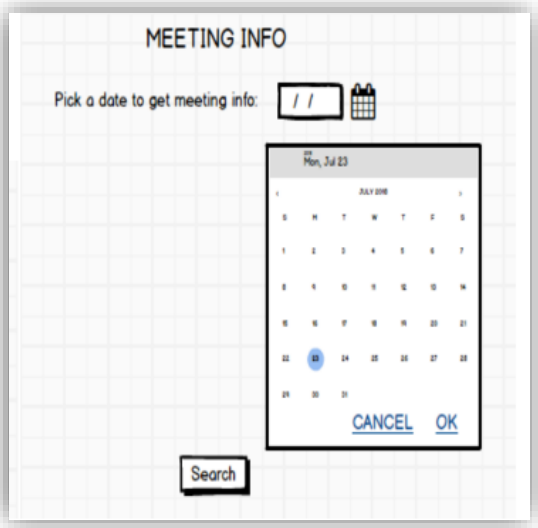
1. Write a program to enter Medicine Name, Date and Time of the Day as input from the user and store it in the SQLite database. Input for Time of the Day should be either Morning or Afternoon or Evening or Night. Trigger an alarm based on the Date and Time of the Day and display the Medicine Name.

PROTOTYPE DESIGN

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”.

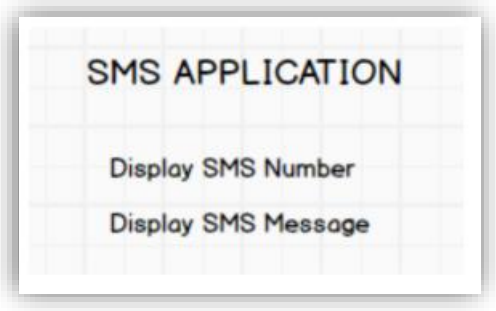
PROTOTYPE DESIGN





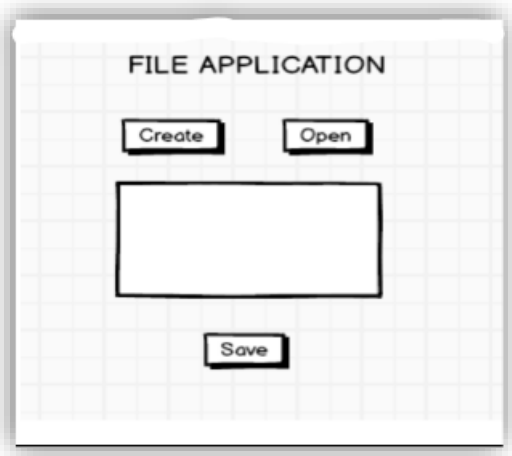
3. Create an application to receive an incoming SMS which is notified to the user. On clicking this SMS notification, the message content and the number should be displayed on the screen. Use appropriate emulator control to send the SMS message to your application.

PROTOTYPE DESIGN



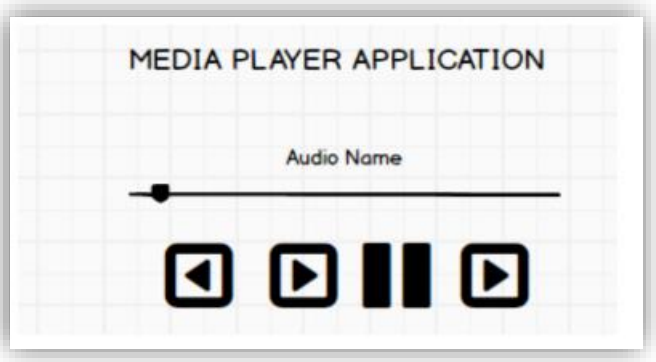
4. Write a program to create an activity having a Text box, and also Save, Open and Create buttons. The user has to write some text in the Text box. On pressing the Create button the text should be saved as a text file in MkSDcard. On subsequent changes to the text, the Save button should be pressed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

PROTOTYPE DESIGN



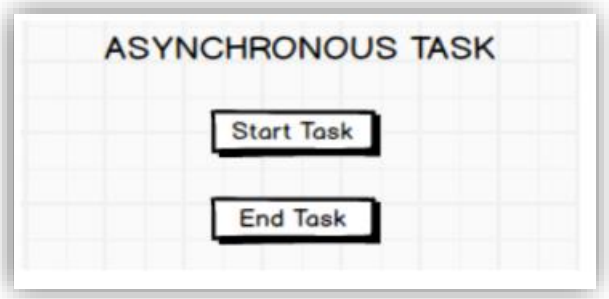
5. Create an application to demonstrate a basic media player that allows the user to Forward, Backward, Play and Pause an audio. Also, make use of the indicator in the seek bar to move the audio forward or backward as required.

PROTOTYPE DESIGN



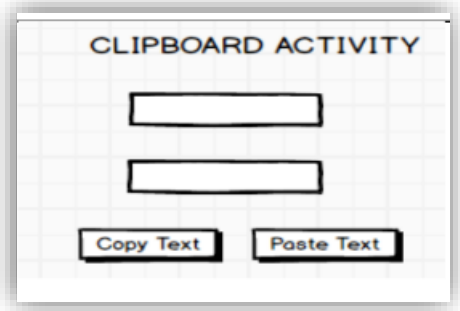
6. Develop an application to demonstrate the use of Asynchronous tasks in android. The asynchronous task should implement the functionality of a simple moving banner. On pressing the Start Task button, the banner message should scroll from right to left. On pressing the Stop Task button, the banner message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

PROTOTYPE DESIGN



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

PROTOTYPE DESIGN



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

$$E = P * (r(1+r)^n)/((1+r)^n - 1)$$

Where

E = The EMI payable on the car loan amount.

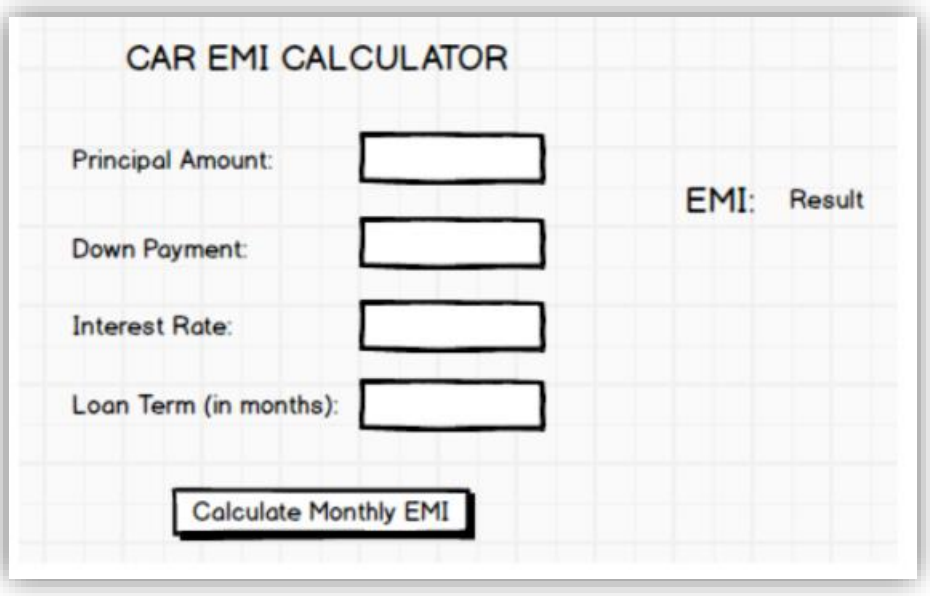
P = The Car Loan Principal Amount.

r = The interest rate value computed on a monthly basis.

n = The loan tenure in the form of months.

The down payment amount has to be deducted from the principal amount paid towards buying the Car. Develop an application that makes use of this AIDL service to calculate the EMI. This application should have four EditText to read the 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.

PROTOTYPE DESIGN



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.