Practical – 1

Aim :- Introduction to Android and Create "Custom Message" application. That will display "Custom Message" in the middle of the screen in the Black color with the Yellow background.

Source code :

MainActivity.java

```
package com.example.wcmc_pract117it086;

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);
    }
}
```

Layout File :-

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:gravity="center"
    android:background="#ffe633"
    android:orientation="vertical"
    tools:context=".MainActivity">

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="ID: 17IT086"
        android:textSize="30dp"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintLeft_toLeftOf="parent"
        app:layout_constraintRight_toRightOf="parent"
        app:layout_constraintTop_toTopOf="parent" />

    <TextView
        android:layout_width="wrap_content"
```

```
        android:layout_height="wrap_content"
        android:text="CSPIT"
        android:textSize="30dp"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintLeft_toLeftOf="parent"
        app:layout_constraintRight_toRightOf="parent"
        app:layout_constraintTop_toTopOf="parent" />

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Branch:IT"
        android:textSize="30dp"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintLeft_toLeftOf="parent"
        app:layout_constraintRight_toRightOf="parent"
        app:layout_constraintTop_toTopOf="parent" />

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="sem 6"
        android:textSize="30dp"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintLeft_toLeftOf="parent"
        app:layout_constraintRight_toRightOf="parent"
        app:layout_constraintTop_toTopOf="parent" />

</LinearLayout>
```

Output :-

Practical - 2

Aim :- Create an android application to calculate the sum of two numbers and gives result in Toast Message.

Source code :-

MainActivity.java

```
package com.example.a17it086;

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 MainActivity extends AppCompatActivity {
   EditText e1,e2;
   Button b1;
   @Override
   protected void onCreate(Bundle savedInstanceState) {
      super.onCreate(savedInstanceState);
      setContentView(R.layout.activity_main);
      e1=findViewById(R.id.num1);
      e2=findViewById(R.id.num2);
      b1=findViewById(R.id.submit);
      b1.setOnClickListener(new View.OnClickListener() {
         @Override
         public void onClick(View view) {
            String s1=e1.getText().toString();
            int i1,i2,i3;
            i1=Integer.parseInt(s1);
            String s2=e2.getText().toString();
            i2=Integer.parseInt(s2);
            i3=i1+i2;
            Toast.makeText(MainActivity.this,"ans:"+i3,Toast.LENGTH_LONG).show();
         }
      });
   }
}
```

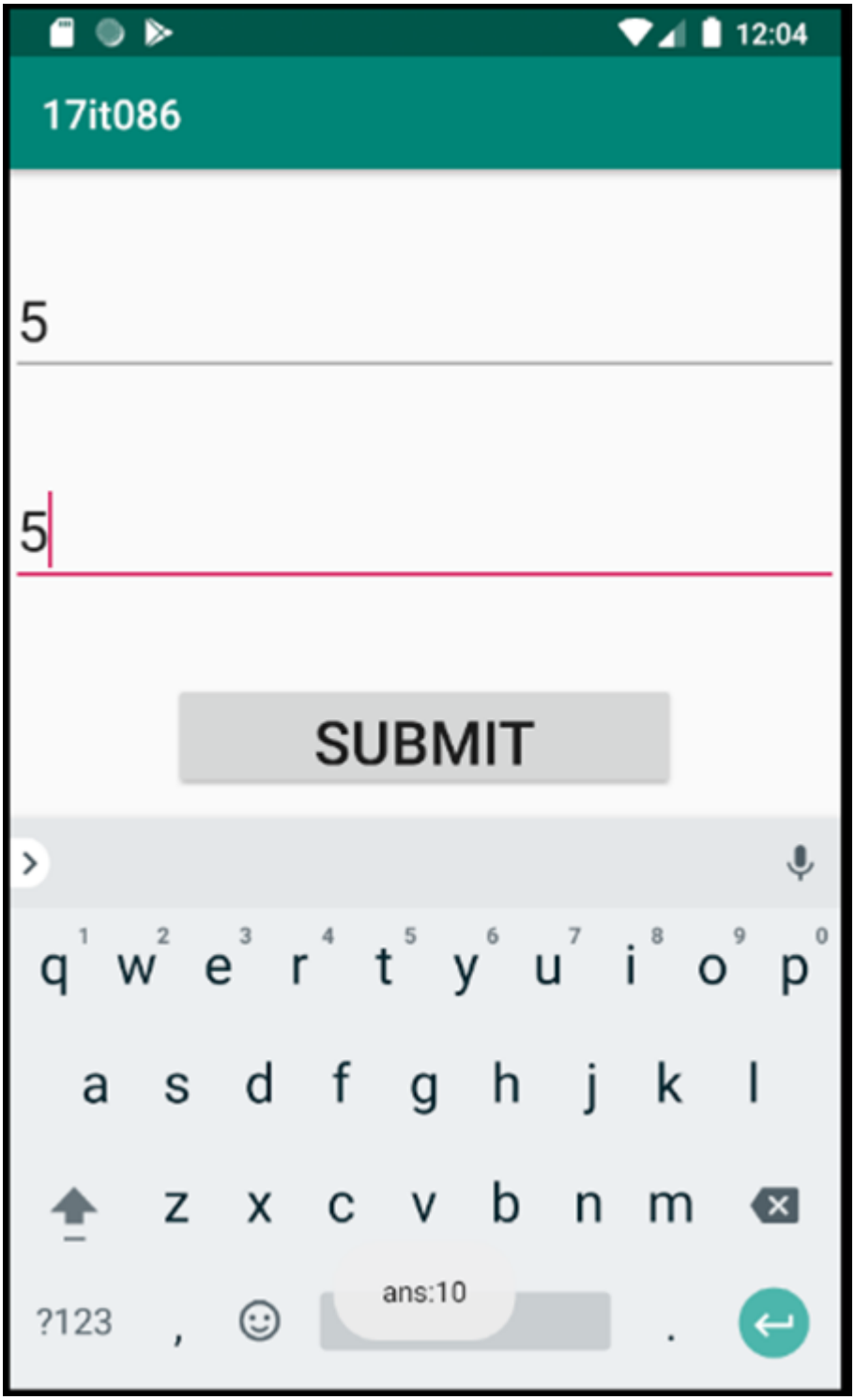
Layout File :-

activity_main.xml

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

```
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".MainActivity">
    <EditText
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:id="@+id/num1"
        android:hint="enter number1"
        android:textSize="28dp"
        android:layout_marginTop="45dp"
        ></EditText>
    <EditText
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:id="@+id/num2"
        android:hint="enter number2"
        android:layout_below="@+id/num1"
        android:textSize="28dp"
        android:layout_marginTop="45dp"
        ></EditText>
    <Button
        android:layout_width="250dp"
        android:layout_height="55dp"
        android:id="@+id/submit"
        android:text="submit"
        android:textSize="30dp"
        android:layout_below="@+id/num2"
        android:layout_marginTop="45dp"
        android:layout_centerHorizontal="true"></Button>
</RelativeLayout>
```

Output :-



Practical - 3

Aim :- Create an application that will display Toast (Message) on specific interval of time.

Source code :-

MainActivity.java

```
package com.example.a17it086_practical3;

import androidx.appcompat.app.AppCompatActivity;

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

public class MainActivity extends AppCompatActivity {
    Chronometer timer;
    int i=0;
    int Duration=11;
    TextView txt;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        setContentView(R.layout.activity_main);
        timer=(Chronometer)findViewById(R.id.chronometer);
        txt=(TextView)findViewById(R.id.textView);
        timer.start();
        timer.setOnChronometerTickListener(new Chronometer.OnChronometerTickListener() {

            @Override
            public void onChronometerTick(Chronometer arg0) {

                txt.setText("Next Massage Will Show After Some Time");
                i++;
                if(i>=Duration)
                {
                    Toast.makeText(MainActivity.this, "Message "+(i/10),Toast.LENGTH_LONG).show();
                    Duration+=10;
                }
                if(Duration>61)
                {
                    timer.stop();
                }
            }
        });
    }
```

```
}
```

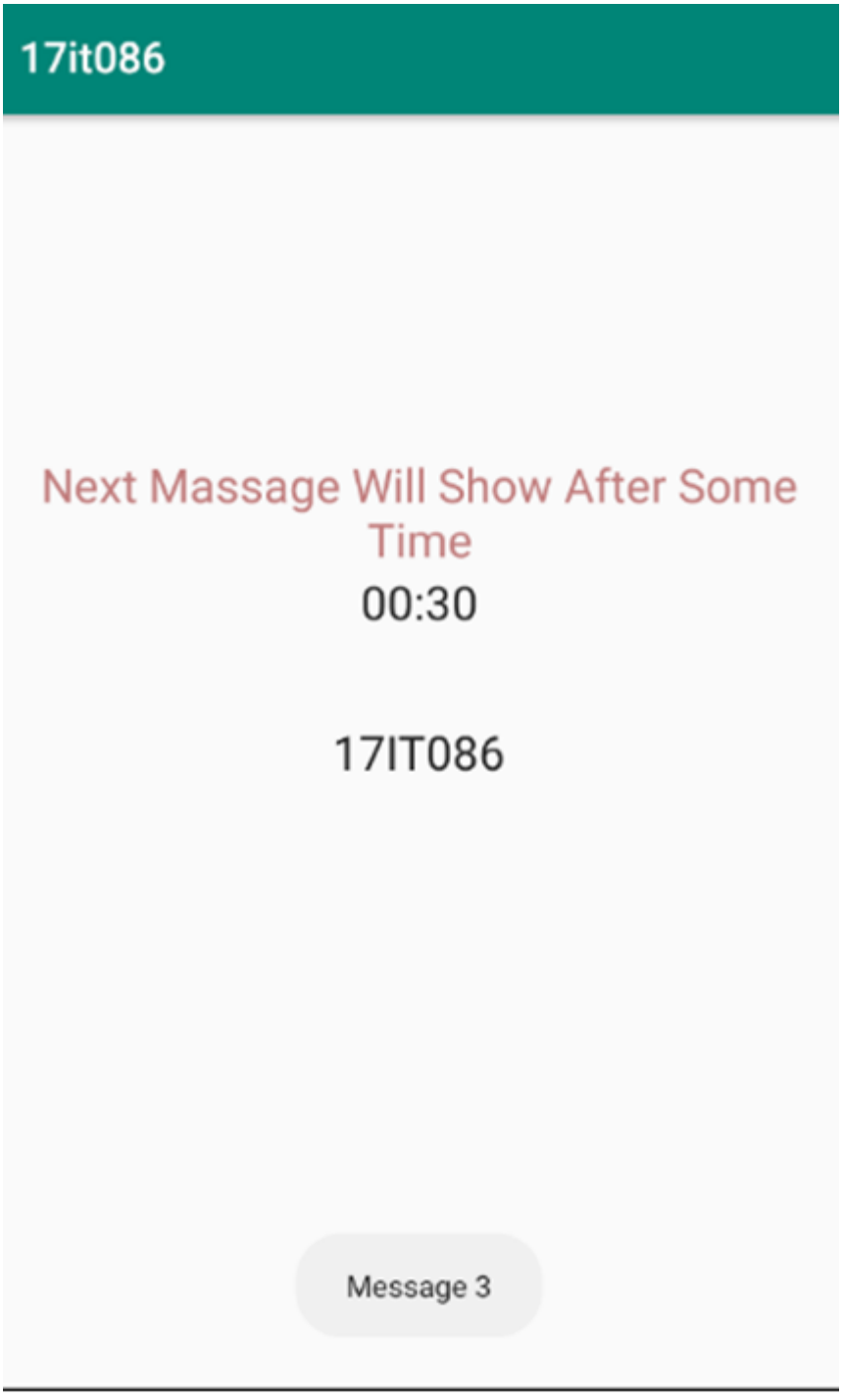
Layout File :-

activity_main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout
xmlns:android="http://schemas.android.com/apk/res/android"
   xmlns:app="http://schemas.android.com/apk/res-auto"
   xmlns:tools="http://schemas.android.com/tools"
   android:layout_width="match_parent"
   android:layout_height="match_parent"
   android:paddingBottom="16dp"
   android:paddingLeft="16dp"
   android:paddingRight="16dp"
   android:paddingTop="16dp"
   tools:context=".MainActivity">

   <TextView
      android:layout_width="wrap_content"
      android:layout_height="wrap_content"
      android:textAppearance="?android:attr/textAppearanceLarge"
      android:text=" PR3_TOAST MESSAGE"
      android:id="@+id/textView"
      android:textIsSelectable="true"
      android:textColor="#c17b7b"
      android:textAlignment="center"
      android:layout_marginTop="144dp"
      android:layout_alignParentTop="true"
      android:layout_centerHorizontal="true" />

   <Chronometer
      android:id="@+id/chronometer"
      android:layout_width="wrap_content"
      android:layout_height="wrap_content"
      android:layout_below="@+id/textView"
      android:layout_centerHorizontal="true"
      android:textAppearance="?android:attr/textAppearanceLarge"
      tools:layout_editor_absoluteX="141dp"
      tools:layout_editor_absoluteY="216dp" />

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

```
        android:layout_centerVertical="true"
        android:text="17IT086"
        android:textAppearance="?android:attr/textAppearanceLarge"
        tools:layout_editor_absoluteX="150dp"
        tools:layout_editor_absoluteY="336dp" />

</RelativeLayout>
```

Output :-



Practical - 4

Aim :- Create a temperature converter Application. (Fahrenheit-Celsius).

Source code :-

MainActivity.java

```
package com.example.a17it086_p4;

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


public class MainActivity extends AppCompatActivity {

    Button b1,b2;
    EditText e1;
    TextView result;
    double n1,n2;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        b1 = findViewById(R.id.b1);
        b2 = findViewById(R.id.b2);
        e1 = findViewById(R.id.e1);
        result = findViewById(R.id.result);

        b1.setOnClickListener(new View.OnClickListener(){
            public void onClick(View v){
                double temp = Double.parseDouble(e1.getText().toString());
                n1 = (temp*1.8)+32;
                String new1 = n1 + " F";
                result.setText(String.valueOf(new1));
            }

        });

        b2.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                double temp = Double.parseDouble(e1.getText().toString());
                n2 =  (temp-32)/1.8;
                String new2 = n2 + " C";
```

```
            result.setText(String.valueOf(new2));
        }
    });
  }
}
```

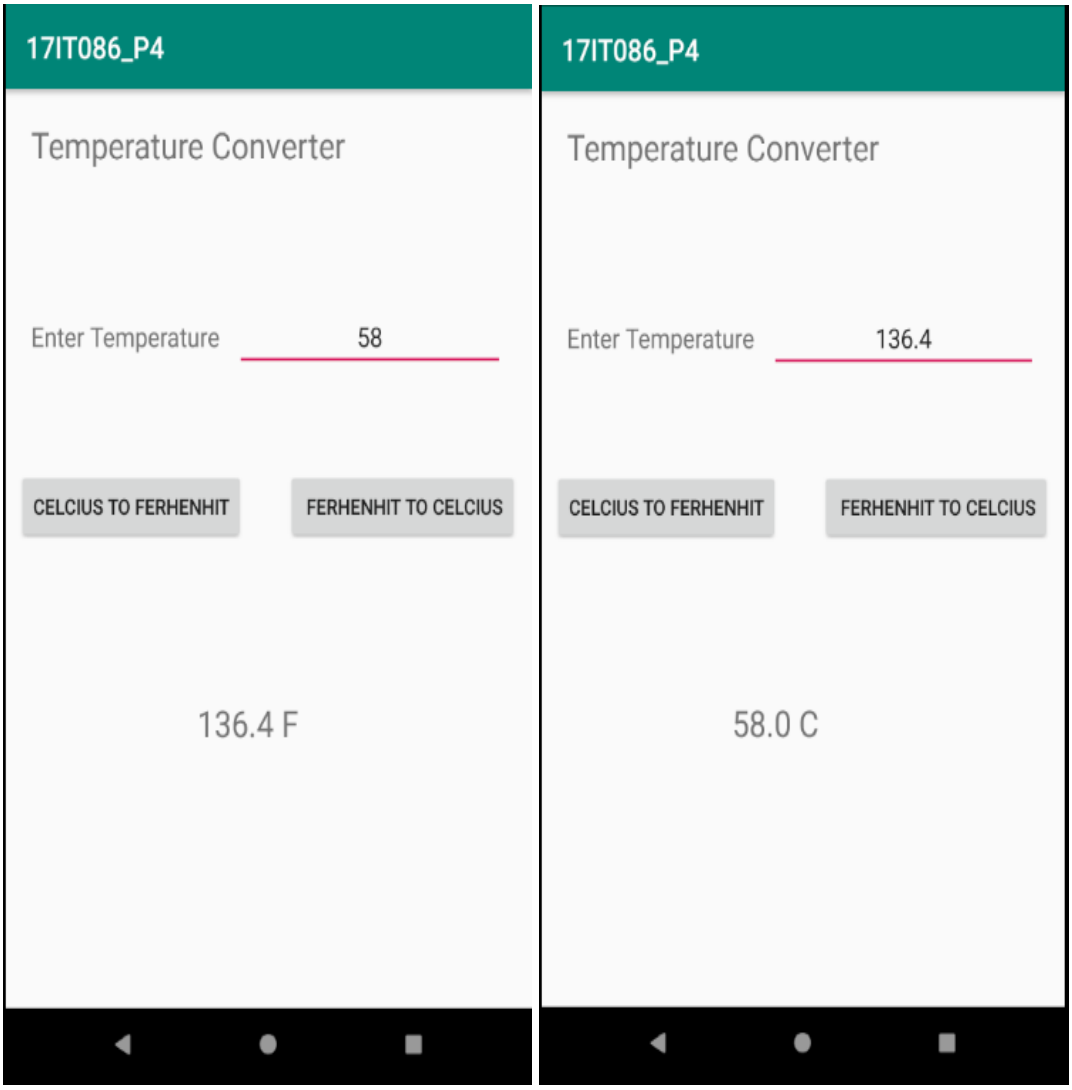
Layout File :-

activity_main.xml

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

   <TextView
      android:layout_width="wrap_content"
      android:layout_height="wrap_content"
      android:text="Temperature Converter"
      android:layout_margin="20dp"
      android:textSize="24sp"
      app:layout_constraintBottom_toBottomOf="parent"
      app:layout_constraintLeft_toLeftOf="parent"
      app:layout_constraintRight_toRightOf="parent"
      app:layout_constraintTop_toTopOf="parent" />

   <TextView
      android:id="@+id/textView"
      android:layout_width="wrap_content"
      android:layout_height="wrap_content"
      android:layout_marginTop="150dp"
      android:text="Enter Temperature"
      android:layout_marginLeft="20dp"
      android:textSize="18sp" />

   <EditText
      android:id="@+id/e1"
      android:layout_width="wrap_content"
      android:layout_height="wrap_content"
      android:layout_marginLeft="180dp"
      android:layout_marginTop="140dp"
      android:ems="10"
      android:hint="temperature"
      android:inputType="numberDecimal"
```

```
    android:textAlignment="center"/>

  <Button
    android:id="@+id/b1"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginLeft="10dp"
    android:layout_marginTop="250dp"
    android:text="celcius to Ferhenhit" />

  <Button
    android:id="@+id/b2"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginLeft="220dp"
    android:layout_marginTop="250dp"
    android:text=" ferhenhit to celcius" />

  <TextView
    android:id="@+id/result"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginLeft="150dp"
    android:layout_marginTop="400dp"
    android:gravity="center"
    android:text="0 degree"
    android:textSize="24sp" />

</RelativeLayout>
```

Output :-


17IT086_P4
Temperature Converter
Enter Temperature
58
CELCIUS TO FERHENHIT
FERHENHIT TO CELCIUS
136.4 F
17IT086_P4
Temperature Converter
Enter Temperature
136.4
CELCIUS TO FERHENHIT
FERHENHIT TO CELCIUS
58.0 C

Practical – 5

Aim :- Create a login application with following features:

1. **Successful Login message in TextView with Green background if Username & password is correct.**
2. **Failure message in TextView with Red background if Username or password is incorrect.**
3. **Disable Login Button after three wrong login attempts.**
4. **Close application if user selects Cancel Button.**

Source code :-

MainActivity.java

```
package com.example.a17it086_practical5;

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

public class MainActivity extends AppCompatActivity {
   EditText txt1,txt2,uname,password;
   TextView t;
   Button button,cancle;
   int counter=5;
   @Override
   protected void onCreate(Bundle savedInstanceState) {
      super.onCreate(savedInstanceState);
      setContentView(R.layout.activity_main);
      txt1 = (EditText) findViewById(R.id.editText);
      txt2 = (EditText) findViewById(R.id.editText2);
      button = (Button) findViewById(R.id.button);
      cancle = (Button) findViewById(R.id.button2);
      button.setOnClickListener(new View.OnClickListener()
      {
         public void onClick(View v)
         {
            try
            {

               if(txt1.getText().length()==0)
               {
                  txt1.setError("Field cannot be left blank.");
```

```
            }
            else if(txt2.getText().length()==0)
            {
               txt2.setError("Field cannot be left blank.");
            }
            else{
               if(txt1.getText().toString().equals("Dharmil")){
                  if(txt2.getText().toString().equals("123")){
                     String str = txt1.getText().toString();
                     Intent intent = new Intent(MainActivity.this, secondactivity.class);
                     intent.putExtra("username", str);
                     startActivity(intent);
                  }
                  else {
                     if (counter > 0) {
                        counter--;
                        txt2.setError("Password is incorrect.");
                     } else {

                        final Handler handler = new Handler();
                        handler.postDelayed(new Runnable()
                        {   @Override
                        public void run()
                        {   button.setEnabled(true);
                        }
                        }, 10000);
                     }
                  }
               }
               else{
                  if(counter>0){
                     counter--;
                     txt1.setError("Username is incorrect.");
                  }
                  else {
                     button.setEnabled(false);
                     Toast alert = Toast.makeText(MainActivity.this, "Login Disabled for 1 mins",
Toast.LENGTH_SHORT);
                     alert.show();

                     final Handler handler = new Handler();
                     handler.postDelayed(new Runnable()
                     {   @Override
                     public void run()
                     {   button.setEnabled(true);
                        counter = 2;
                     }
                     }, 10000);
```

```
                    }
                }
            }
        }
        catch(Exception e)
        {
            Toast.makeText(getBaseContext(), e.getMessage(),
                Toast.LENGTH_SHORT).show();
        }


    }
});

cancle.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        finish();
    }
});
}
}
```

secondactivity.java

```
package com.example.a17it086_practical5;

import androidx.appcompat.app.AppCompatActivity;

import android.content.Intent;
import android.os.Bundle;
import android.widget.TextView;
public class secondactivity extends AppCompatActivity {
    TextView t1;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_secondactivity);
        t1 = findViewById(R.id.textView);
        Intent intent = getIntent();

        String user = intent.getStringExtra("username");
        t1.setText(user);
    }

    public void previous(){
        startActivity(new Intent(secondactivity.this,MainActivity.class));
```

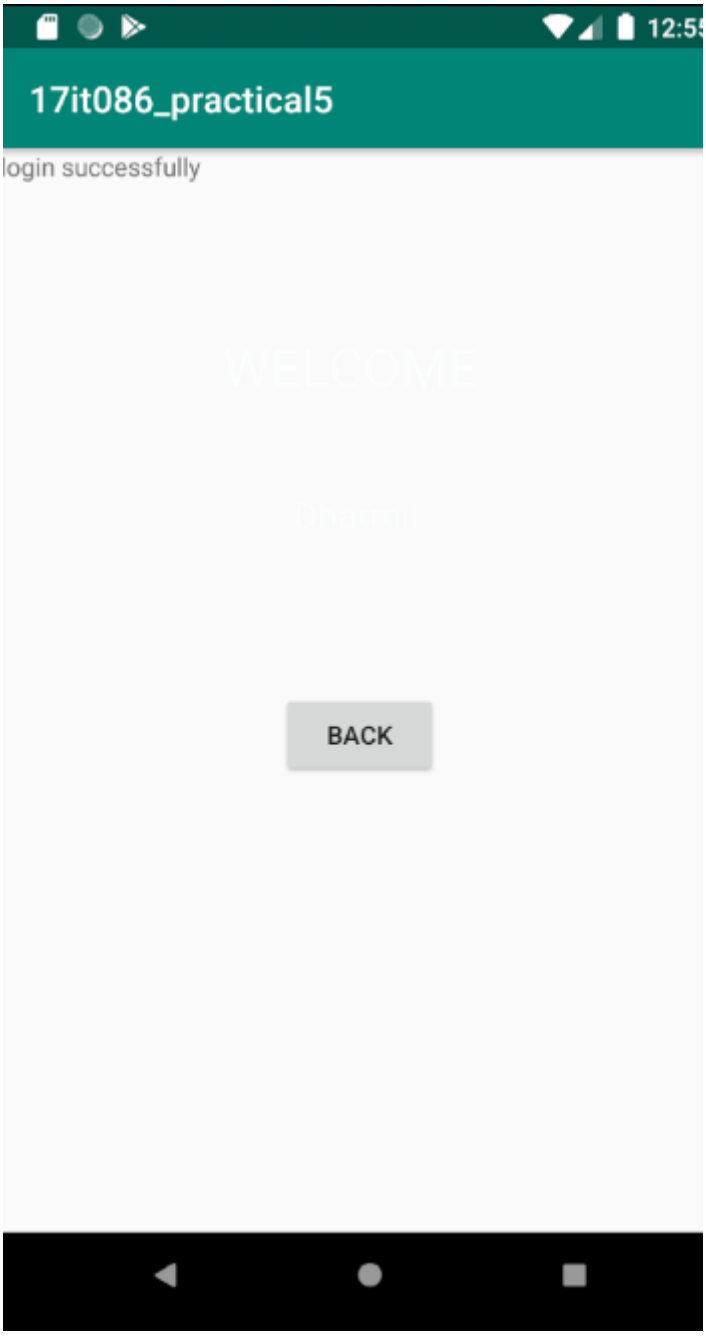
```
    }

}
```

Layout File :-
activity_main.xml

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

    <EditText
        android:id="@+id/editText"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignParentTop="true"
        android:layout_centerHorizontal="true"
        android:layout_marginTop="90dp"
        android:ems="10"
        android:hint="Username"
        android:inputType="textPersonName"
        />

    <EditText
        android:id="@+id/editText2"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignParentTop="true"
        android:layout_centerHorizontal="true"
        android:layout_marginTop="180dp"
        android:ems="10"
        android:hint="Password"
        android:inputType="textPassword"
        />

    <Button
        android:id="@+id/button"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignEnd="@+id/editText"
```

```
    android:layout_alignParentBottom="true"
    android:layout_marginBottom="270dp"
    android:text="Login" />

 <Button
    android:id="@+id/button2"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_alignStart="@+id/editText"
    android:layout_alignTop="@+id/button"
    android:layout_marginStart="0dp"
    android:text="Cancle" />
</RelativeLayout>
```

output :-


12:55
17it086_practical5
login successfully
WELCOME
Dharmil
BACK

Practical - 6

Aim :- Create an application which turns ON or OFF Torch/Flashlight of Camera.

Source code :-

MainActivity.java

```
package com.example.a17it086_p6;

import androidx.appcompat.app.AlertDialog;
import androidx.appcompat.app.AppCompatActivity;

import android.os.Bundle;
import android.content.Context;
import android.content.DialogInterface;
import android.content.pm.PackageManager;
import android.hardware.camera2.CameraAccessException;
import android.hardware.camera2.CameraManager;
import android.widget.CompoundButton;
import android.widget.ToggleButton;

public class MainActivity extends AppCompatActivity {
    private ToggleButton toggleButton;
    private CameraManager mCameraManager;
    private String mCameraId;

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

        boolean isFlashAvailable =
getApplicationContext().getPackageManager().hasSystemFeature(PackageManager.FEATURE_CAMERA
_FLASH);

        if (!isFlashAvailable) {
            showNOFlashError();
        }

        mCameraManager = (CameraManager) getSystemService(Context.CAMERA_SERVICE);
        try {
            mCameraId = mCameraManager.getCameraIdList()[0];
        } catch (CameraAccessException e) {
            e.printStackTrace();
        }


    toggleButton = findViewById(R.id.togglebutton);
```

```
        toggleButton.setOnCheckedChangeListener(new CompoundButton.OnCheckedChangeListener() {
            @Override
            public void onCheckedChanged(CompoundButton buttonView, boolean isChecked) {
                switchFlashLight(isChecked);
            }
        });
    }
    private void showNOFlashError() {
        AlertDialog alert = new AlertDialog.Builder(this).create();
        alert.setTitle("opps!");
        alert.setMessage("Flash not available in this device...");
        alert.setButton(DialogInterface.BUTTON_POSITIVE, "OK", new DialogInterface.OnClickListener()
{
            @Override
            public void onClick(DialogInterface dialog, int which) {
                finish();
            }
        });
        alert.show();


    }
    public void switchFlashLight(boolean status) {
        try {
            mCameraManager.setTorchMode(mCameraId, status);
        } catch (CameraAccessException e) {
            e.printStackTrace();
        }
    }
}
```

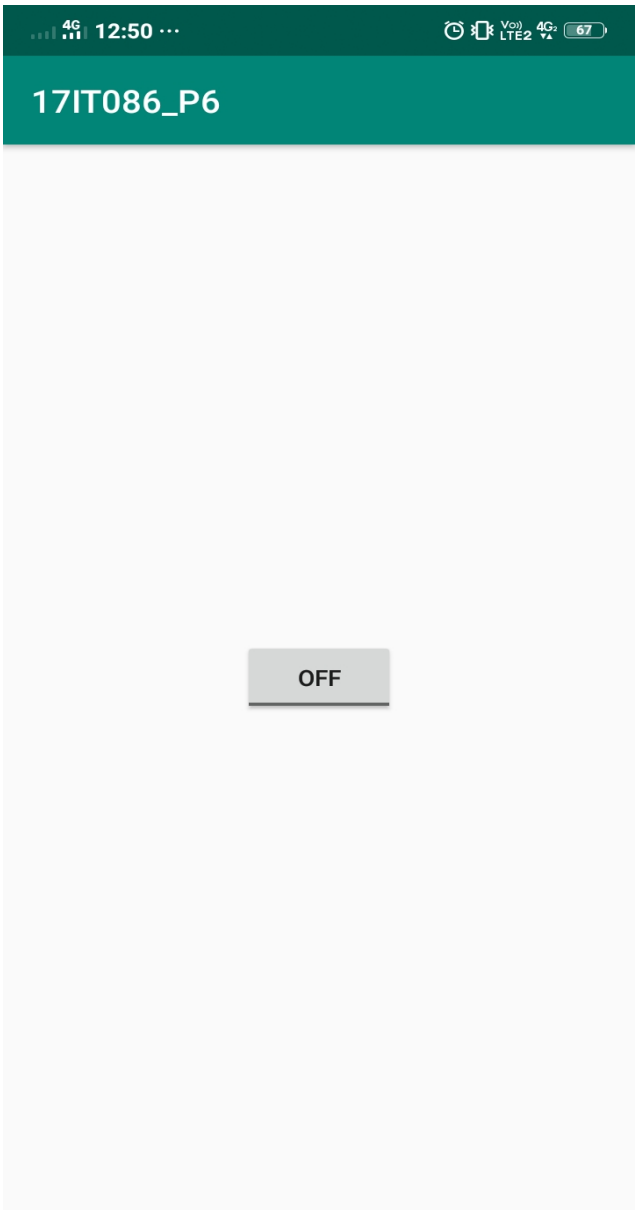
Layout File :-

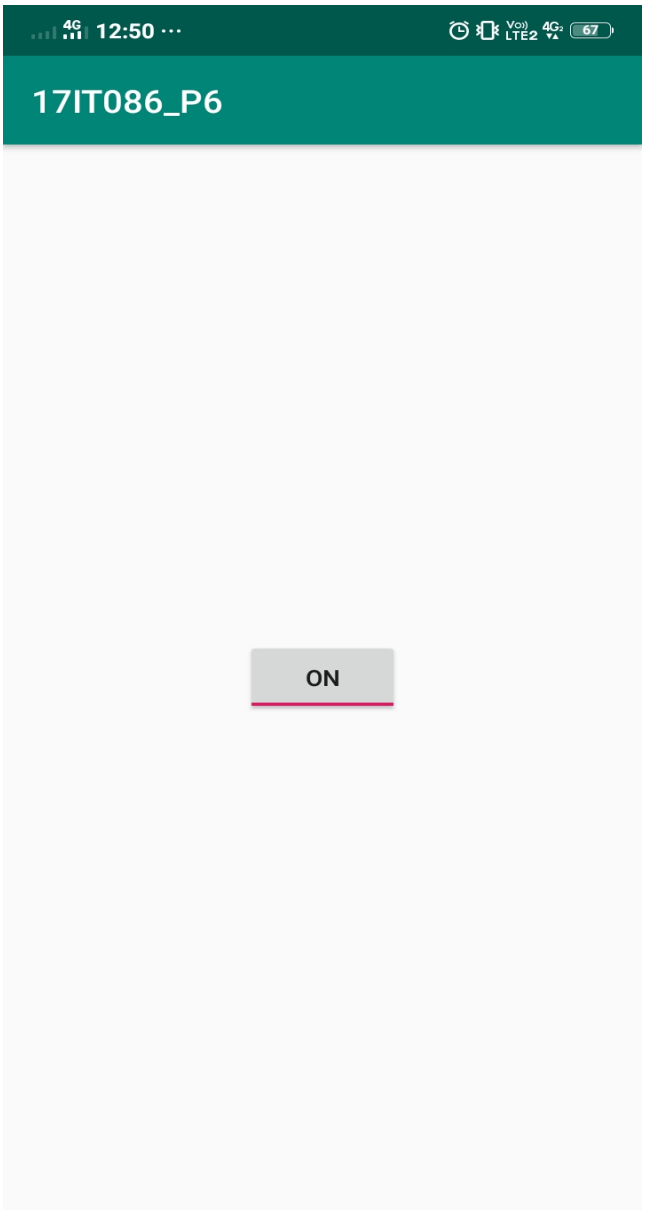
activity_main.xml

```
<?xml version="1.0" encoding="utf-8"?>

<RelativeLayout
xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".MainActivity">

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

```
    android:layout_centerInParent="true"
    android:id="@+id/togglebutton"
    android:textOff="off"
    android:textOn="on"
    ></ToggleButton>
</RelativeLayout>
```

Output :-





Practical-7

Aim:- Create an application that will change color of the screen, based on selected options from the menu.

Source Code:

MainActivity.java

```
package com.example.a17it086_p7;

import androidx.appcompat.app.AppCompatActivity;

import android.graphics.Color;
import android.os.Bundle;
import android.view.Menu;
import android.view.MenuItem;
import android.widget.LinearLayout;
import android.widget.RelativeLayout;
import android.widget.TextView;
import android.widget.Toast;

public class MainActivity extends AppCompatActivity {
    TextView text1,text2;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        text1 = findViewById(R.id.text1);
        text2 = findViewById(R.id.text2);

    }
    public boolean onCreateOptionsMenu(Menu menu){
        getMenuInflater().inflate(R.menu.p7,menu);
        return true;
    }

    public boolean onOptionsItemSelected(MenuItem item){
        text1.setText("");
        RelativeLayout r1 = (RelativeLayout)findViewById(R.id.bgrnd_id);
        switch(item.getItemId()){
            case R.id.blue1:
                r1.setBackgroundColor(Color.BLUE);
                text2.setText("BLUE");
                text2.setBackgroundColor(Color.WHITE);
                for (int i=0;i<2;i++) {
                    Toast.makeText(this, "BLUE", Toast.LENGTH_SHORT).show();
                }
                return true;
            case R.id.red1:
                r1.setBackgroundColor(Color.RED);
                text2.setText("RED");
```

```
            text2.setBackgroundColor(Color.WHITE);
            Toast.makeText(this, "RED", Toast.LENGTH_SHORT).show();
            return true;
        case R.id.green1:
            r1.setBackgroundColor(Color.GREEN);
            text2.setText("GREEN");
            text2.setBackgroundColor(Color.WHITE);
            Toast.makeText(this, "GREEN", Toast.LENGTH_SHORT).show();
            return true;
        case R.id.yellow1:
            text2.setText("YELLOW");
            r1.setBackgroundColor(Color.YELLOW);
            text2.setBackgroundColor(Color.WHITE);
            Toast.makeText(this, "YELLOW", Toast.LENGTH_SHORT).show();
            return true;

        default:
            Toast.makeText(this, "Nothing is selected", Toast.LENGTH_SHORT).show();
            return true;
    }
  }
}
```

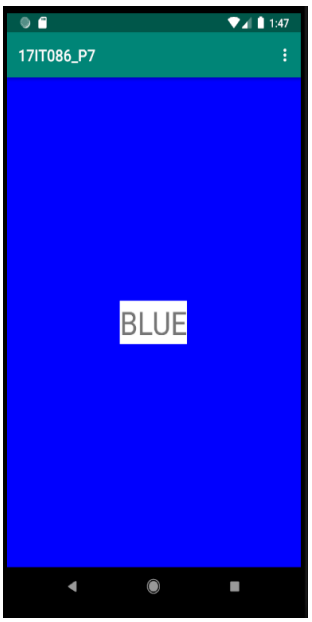
activity_main.xml

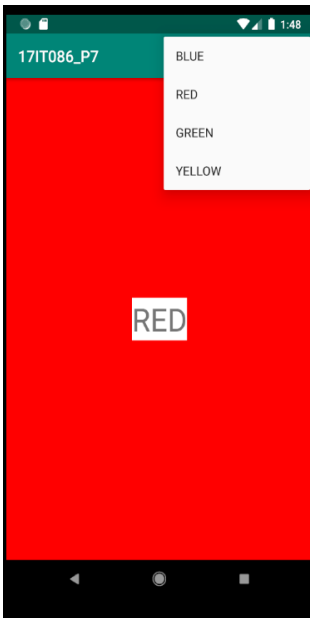
```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
   xmlns:app="http://schemas.android.com/apk/res-auto"
   xmlns:tools="http://schemas.android.com/tools"
   android:id="@+id/bgrnd_id"
   android:gravity="center"
   android:layout_width="match_parent"
   android:layout_height="match_parent"
   android:background="#FFFFFF"
   tools:context=".MainActivity">

   <TextView
      android:id="@+id/text1"
      android:layout_width="wrap_content"
      android:layout_height="wrap_content"
      android:text="CLICK the DOTS above in ActionBar"
      android:textSize="20dp"></TextView>
   <TextView
      android:id="@+id/text2"
      android:layout_width="wrap_content"
      android:layout_height="wrap_content"
      android:textSize="40dp"
      android:text="" />

</RelativeLayout>
```

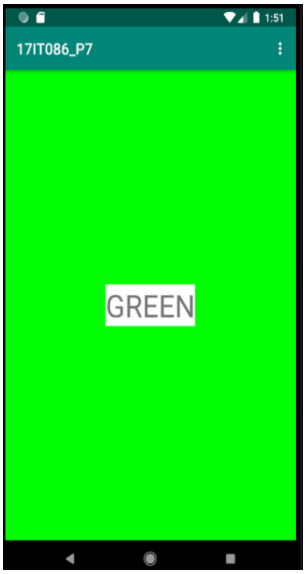
P7.xml

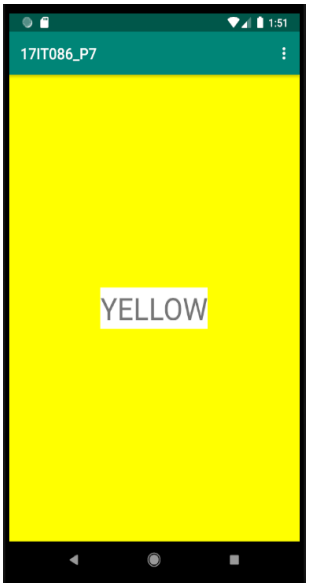
```
<?xml version="1.0" encoding="utf-8"?>
<menu xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:android="http://schemas.android.com/apk/res/android">
    <item
        android:id="@+id/blue1"
        android:title="BLUE" />
    <item android:id="@+id/red1"
        android:title="RED"></item>
    <item
        android:id="@+id/green1"
        android:title="GREEN" />
    <item
        android:id="@+id/yellow1"
        android:title="YELLOW" />
</menu>
```

Output:









Practical-8

Aim: Create an application with the help of fragment.

Activity_main.xml

```
<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">
  <LinearLayout
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:orientation="horizontal"
    android:weightSum="4">
    <Button android:id="@+id/button1"
      android:layout_width="wrap_content"
      android:layout_height="wrap_content"
      android:text="Fragment One"
      android:onClick="FragmentOneClick"
      android:layout_weight="2"/>

    <Button android:id="@+id/button2"
      android:layout_width="wrap_content"
      android:layout_height="wrap_content"
      android:text="Fragment Two"
      android:onClick="FragmentTwoClick"
      android:layout_weight="2"/>
  </LinearLayout>

  <fragment
    android:id="@+id/fragment_switch"
    android:name="com.example.admin1.fragment.FragmentOne"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:layout_marginLeft="5dp"
    android:layout_marginRight="5dp"
    android:layout_marginBottom="5dp"/>

</LinearLayout>
```

mainactivity.java

```
package com.example.admin1.fragment;
import android.app.Activity;
    import android.app.Fragment;
    import android.app.FragmentManager;
    import android.app.FragmentTransaction;
```

```
import android.os.Bundle;

import android.view.View;
public class MainActivity extends Activity {

  @Override
  protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);
  }
  public void FragmentOneClick(View view) {
    Fragment myfragment;
    myfragment = new FragmentOne();

    FragmentManager fm = getFragmentManager();
    FragmentTransaction fragmentTransaction =
    fm.beginTransaction();
    fragmentTransaction.replace(R.id.fragment_switch,
    myfragment); fragmentTransaction.commit();

  }
  public void FragmentTwoClick(View view) {
    Fragment myfragment;
    myfragment = new FragmentTwo();

    FragmentManager fm = getFragmentManager();
    FragmentTransaction fragmentTransaction =
    fm.beginTransaction();
    fragmentTransaction.replace(R.id.fragment_switch,
    myfragment); fragmentTransaction.commit();

  }
}
```

fragment_one_.xml

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

<TextView android:id="@+id/textView1"
  android:layout_width="match_parent"
  android:layout_height="match_parent"
  android:layout_weight="1"
  android:text="Fragment One"
  android:textStyle="bold"
  android:textSize="20dp"
  android:textColor="#ffffff"
  android:gravity="center"
  android:background="#367396"/>
 </LinearLayout>
```

fragment_two.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
  android:layout_width="match_parent"
  android:layout_height="match_parent"
  android:orientation="vertical">
  <TextView android:id="@+id/textView2"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="#563256"
    android:gravity="center" android:text="Fragment
    Two" android:textColor="#ffffff"
    android:textSize="20dp" android:textStyle="bold"
    />

</LinearLayout>
```

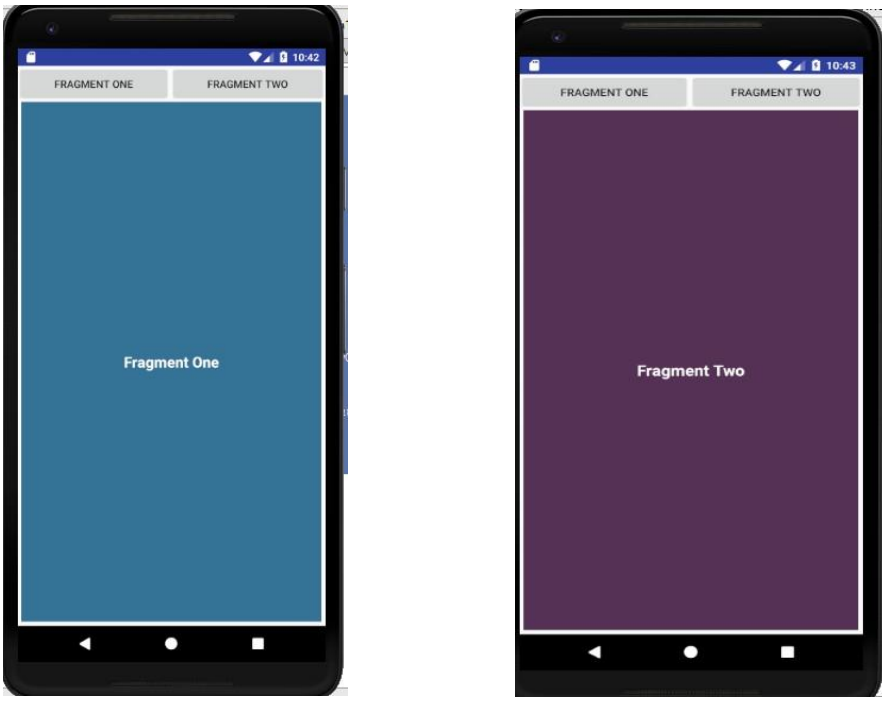
fragmentOne.java

```
package com.example.admin1.fragment;
import android.app.Fragment;
import android.os.Bundle;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;

public class FragmentOne extends Fragment {
  public View onCreateView(LayoutInflater inflater, ViewGroup container,
                Bundle savedInstanceState) {

    return inflater.inflate(R.layout.fragment_one, container, false);

  }
}
```

FragmentTwo.java

```
package com.example.admin1.fragment;

import android.app.Fragment;
import android.os.Bundle;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;


public class FragmentTwo extends Fragment {
  public View onCreateView(LayoutInflater inflater, ViewGroup container,
                Bundle savedInstanceState) {

    return inflater.inflate(R.layout.fragment_twolayout, container, false);

  }
}
```

OUTPUT:



Conclusion:

We successfully completed creating application using fragment.

Practical-9

Aim: Create an application with the help of web view.

Activity_main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context="com.example.administrator.prac9_webview.MainActivity">

    <TextView android:text="WebView" android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:id="@+id/textview"
        android:textSize="35dp"
        android:layout_alignParentTop="true"
        android:layout_centerHorizontal="true" />

    <TextView
        android:id="@+id/textView"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignParentTop="true"
        android:layout_centerHorizontal="true"
        android:layout_marginTop="35dp"
        android:text="charusat website"
```

```
    android:textColor="#ff7aff24"
    android:textSize="35dp" />

<EditText
    android:id="@+id/editText"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_alignParentStart="true"
    android:layout_below="@+id/textView"
    android:layout_marginTop="11dp"
    android:focusable="true"
    android:hint="Enter Text"
    android:textColorHighlight="#ff7eff15"
    android:textColorHint="#ffff25e6" />

<ImageView
    android:id="@+id/imageView"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"

    android:layout_alignEnd="@+id/textView"
    android:layout_below="@+id/button"
    android:src="@drawable/abc" />

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

```
    android:layout_width="wrap_content"

    android:layout_height="wrap_content"

    android:layout_below="@+id/editText"

    android:layout_centerHorizontal="true"

    android:text="Enter" />


  <WebView

    android:id="@+id/webView"

    android:layout_width="match_parent"

    android:layout_height="wrap_content"

    android:layout_alignStart="@+id/textview"

    android:layout_marginBottom="0dp"

    android:layout_marginLeft="0dp"

    android:layout_marginRight="100dp"

    android:layout_marginTop="404dp"

   android:visibility="visible" />
 </RelativeLayout>
```

Mainactivity.java

```
package com.example.administrator.prac9_webview;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.view.View;
import android.webkit.WebView;
import android.webkit.WebViewClient;
import android.widget.Button;
import android.widget.EditText;
public class MainActivity extends AppCompatActivity
{
   Button b1;
   EditText ed1;
   private WebView wv1;
   protected void onCreate(Bundle savedInstanceState)

   {
```

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



    b1=(Button)findViewById(R.id.button);
    ed1=(EditText)findViewById(R.id.editText);



    wv1=(WebView)findViewById(R.id.webView);

    wv1.setWebViewClient(new MyBrowser());



    b1.setOnClickListener(new View.OnClickListener() {

        @Override

        public void onClick(View v) {

            String url = ed1.getText().toString();



            wv1.getSettings().setLoadsImagesAutomatically(true);

            wv1.getSettings().setJavaScriptEnabled(true);

            wv1.setScrollBarStyle(View.SCROLLBARS_INSIDE_OVERLAY);

            wv1.loadUrl(url);

        }

    });

}




private class MyBrowser extends WebViewClient {

    @Override

    public boolean shouldOverrideUrlLoading(WebView view, String url) {

        view.loadUrl(url);
```

```
            return true;

        }

    }

}
```

Output :

Conclusion:

We have successfully completed an application with the help of webview.

Practical-10

Aim: Create an application with the help of database.

Source Code:
Java File/s:
P10.java

```
package com.example.a17it086_wcmc.Fragment.p10;

import android.app.AlertDialog;
import android.database.Cursor;
import android.os.Bundle;
import androidx.fragment.app.Fragment;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.Button;
import android.widget.EditText;
import com.example.a17it086_wcmc.Core.Feature;
import com.example.a17it086_wcmc.R;

public class p10 extends Fragment {
    DatabaseHelper mydb;
    EditText name,surname,marks,id;
    Button addData,viewData,update,delete;
    boolean isInserted,isUpdate;
    Integer deletedRows;

    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
    }

    @Override
    public View onCreateView(LayoutInflater inflater, ViewGroup container, Bundle savedInstanceState) {
        View header =  inflater.inflate(R.layout.fragment_p10, container, false);

        mydb = new DatabaseHelper(getContext());

        name = header.findViewById(R.id.p10_name);
        surname = header.findViewById(R.id.p10_surname);
        marks = header.findViewById(R.id.p10_marks);
        id = header.findViewById(R.id.p10_id);
        addData = header.findViewById(R.id.p10_add_data);
        viewData = header.findViewById(R.id.p10_view_data);
        update = header.findViewById(R.id.p10_update);
        delete = header.findViewById(R.id.p10_delete);

        addData.setOnClickListener(new View.OnClickListener() {
            @Override
```

```
        public void onClick(View v) {
            isInserted =
mydb.insertData(name.getText().toString(),surname.getText().toString(),marks.getText().toString());
            if (isInserted == true){
                Feature.Toast(getContext(),"Data Inserted");
            }else {
                Feature.Toast(getContext(),"Data not Inserted");
            }
        }
    });

    viewData.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            Cursor res = mydb.getAllData();
            if (res.getCount() == 0){
                showMessage("Error","Nothing Found");
                return;
            }

            StringBuffer buffer = new StringBuffer();
            while (res.moveToNext()){
                buffer.append("ID :"+res.getString(0)+"\n");
                buffer.append("NAME :"+res.getString(1)+"\n");
                buffer.append("SURNAME :"+res.getString(2)+"\n");
                buffer.append("MARKS :"+res.getString(3)+"\n\n");
            }
            showMessage("Data",buffer.toString());
        }
    });

    update.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            isUpdate =
mydb.updateData(id.getText().toString(),name.getText().toString(),surname.getText().toString(),marks.getText()
.toString());
            if (isUpdate == true){
                Feature.Toast(getContext(),"Data Updated");
            }else {
                Feature.Toast(getContext(),"Data not Updated");
            }
        }
    });
    delete.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            deletedRows = mydb.deleteData(id.getText().toString());
            if (deletedRows > 0){
                Feature.Toast(getContext(),"Data Deleted");
            }else {
                Feature.Toast(getContext(), "Data not Deleted");
```

```
            }
          }
        });
        return header;
    }
    public void showMessage(String title,String Message){
        AlertDialog.Builder builder = new AlertDialog.Builder(getContext());
        builder.setCancelable(true);
        builder.setTitle(title);
        builder.setMessage(Message);
        builder.show();
    }
}
```

DatabaseHelper.java

```
package com.example.a17it086_wcmc.Fragment.p10;

import android.content.ContentValues;
import android.content.Context;
import android.database.Cursor;
import android.database.sqlite.SQLiteDatabase;
import android.database.sqlite.SQLiteOpenHelper;
import androidx.annotation.Nullable;
import com.example.a17it086_wcmc.Core.ConstantString;

public class DatabaseHelper extends SQLiteOpenHelper {
    long result;

    public DatabaseHelper(@Nullable Context context) {
        super(context, ConstantString.DATABASE_NAME, null, 1);
    }

    @Override
    public void onCreate(SQLiteDatabase db) {
        db.execSQL("create table " + ConstantString.TABLE_NAME + " (ID INTEGER PRIMARY KEY
AUTOINCREMENT, NAME TEXT, SURNAME TEXT, MARKS INTEGER)");
    }

    @Override
    public void onUpgrade(SQLiteDatabase db, int oldVersion, int newVersion) {
        db.execSQL("DROP TABLE IF EXISTS " + ConstantString.TABLE_NAME);
        onCreate(db);
    }

    public boolean insertData(String name,String surname,String marks){
        SQLiteDatabase DB = this.getWritableDatabase();
        ContentValues contentValues = new ContentValues();
        contentValues.put(ConstantString.COL_2,name);
        contentValues.put(ConstantString.COL_3,surname);
        contentValues.put(ConstantString.COL_4,marks);
        result = DB.insert(ConstantString.TABLE_NAME,null,contentValues);
        if (result == -1){
```

```
            return false;
        }else {
            return true;
        }
    }
    public Cursor getAllData(){
        SQLiteDatabase DB = this.getWritableDatabase();
        Cursor res = DB.rawQuery("select * from " +ConstantString.TABLE_NAME,null);
        return res;
    }
    public boolean updateData(String id,String name,String surname,String marks){
        SQLiteDatabase DB = this.getWritableDatabase();
        ContentValues contentValues = new ContentValues();
        contentValues.put(ConstantString.COL_1,id);
        contentValues.put(ConstantString.COL_2,name);
        contentValues.put(ConstantString.COL_3,surname);
        contentValues.put(ConstantString.COL_4,marks);
        DB.update(ConstantString.TABLE_NAME,contentValues,"ID = ?",new String[] { id });
        return true;
    }
    public Integer deleteData(String id){
        SQLiteDatabase DB = this.getWritableDatabase();
        return DB.delete(ConstantString.TABLE_NAME,"ID = ?",new String[] { id });
    }
}
```

Layout File/s:

fragment_p10.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=".Fragment.p10.p10">

    <TextView
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="Student Data"
        android:gravity="center"
        android:textSize="40sp"
        android:padding="50dp"
        android:textColor="@color/black"/>

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

```
    android:orientation="horizontal"
    android:layout_marginLeft="20dp"
    android:layout_marginRight="20dp">

    <TextView
        android:layout_width="110dp"
        android:layout_height="wrap_content"
        android:text="NAME        : "
        android:textColor="@color/black"
        android:textSize="20sp" />

    <EditText
        android:id="@+id/p10_name"
        android:layout_width="260dp"
        android:layout_height="wrap_content"
        android:textColor="@color/black"
        android:textSize="20sp"
        android:imeOptions="actionNext"/>
</LinearLayout>

<LinearLayout
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:orientation="horizontal"
    android:layout_marginLeft="20dp"
    android:layout_marginRight="20dp">

    <TextView
        android:layout_width="110dp"
        android:layout_height="wrap_content"
        android:text="SURNAME : "
        android:textColor="@color/black"
        android:textSize="20sp" />

    <EditText
        android:id="@+id/p10_surname"
        android:layout_width="260dp"
        android:layout_height="wrap_content"
        android:textColor="@color/black"
        android:textSize="20sp"
        android:imeOptions="actionNext"/>
</LinearLayout>

<LinearLayout
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:orientation="horizontal"
    android:layout_marginLeft="20dp"
    android:layout_marginRight="20dp">

    <TextView
        android:layout_width="110dp"
```

```
        android:layout_height="wrap_content"
        android:text="MARKS      : "
        android:textColor="@color/black"
        android:textSize="20sp" />

    <EditText
        android:id="@+id/p10_marks"
        android:layout_width="260dp"
        android:layout_height="wrap_content"
        android:textColor="@color/black"
        android:textSize="20sp"
        android:inputType="numberDecimal"
        android:imeOptions="actionNext"/>
</LinearLayout>

<LinearLayout
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:orientation="horizontal"
    android:layout_marginLeft="20dp"
    android:layout_marginRight="20dp">

    <TextView
        android:layout_width="110dp"
        android:layout_height="wrap_content"
        android:text="ID              : "
        android:textColor="@color/black"
        android:textSize="20sp" />

    <EditText
        android:id="@+id/p10_id"
        android:layout_width="260dp"
        android:layout_height="wrap_content"
        android:textColor="@color/black"
        android:textSize="20sp"
        android:inputType="numberDecimal"
        android:imeOptions="actionDone"/>
</LinearLayout>

<LinearLayout
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:orientation="horizontal"
    android:layout_marginLeft="30dp"
    android:layout_marginRight="20dp"
    android:layout_marginTop="50dp">
    <Button
        android:id="@+id/p10_add_data"
        android:layout_width="140dp"
        android:layout_height="60dp"
        android:text="Add Data "
        android:textColor="@color/black"
```

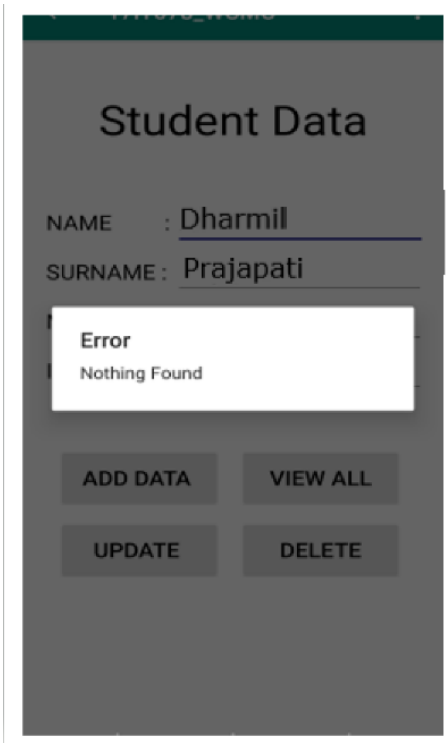
```
        android:textSize="20sp" />
    <Button
        android:id="@+id/p10_view_data"
        android:layout_width="140dp"
        android:layout_height="60dp"
        android:text="View All"
        android:textColor="@color/black"
        android:textSize="20sp"
        android:layout_marginLeft="15dp"/>
  </LinearLayout>

  <LinearLayout
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:orientation="horizontal"
    android:layout_marginLeft="30dp"
    android:layout_marginRight="20dp"
    android:layout_marginTop="10dp">

    <Button
        android:id="@+id/p10_update"
        android:layout_width="140dp"
        android:layout_height="60dp"
        android:text="Update "
        android:textColor="@color/black"
        android:textSize="20sp" />

    <Button
        android:id="@+id/p10_delete"
        android:layout_width="140dp"
        android:layout_height="60dp"
        android:text="Delete"
        android:textColor="@color/black"
        android:textSize="20sp"
        android:layout_marginLeft="15dp"/>
  </LinearLayout>
</LinearLayout>
```

Output:


Student Data
NAME : Dharmil
SURNAME : Prajapati
MARKS : 8.11
ID : 17IT086
ADD DATA
VIEW ALL
UPDATE
DELETE
Data Inserted


Student Data
NAME : Dharmil
SURNAME : Prajapati
MARKS : 8.11
ID : 17IT086
ADD DATA
VIEW ALL
UPDATE
DELETE
Data Updated


Student Data
NAME : Dharmil
SURNAME : Prajapati
MARKS : 8.11
ID : 17IT086
ADD DATA
VIEW ALL
UPDATE
DELETE
Data Deleted


Student Data
NAME : Dharmil
SURNAME : Prajapati
Error
Nothing Found
ADD DATA
VIEW ALL
UPDATE
DELETE

Practical-11

Aim: Creating an application that provides Single Sign-on (SSO) with Chrome Custom Tabs via the AppAuth library, and optionally push managed configuration to provide a user login.

MainActivity.java

```
package com.example.sso;
import android.app.PendingIntent;
import android.content.BroadcastReceiver;
import android.content.Context;
import android.content.Intent;
import android.content.IntentFilter;
import android.content.RestrictionsManager;
import android.net.Uri;
import android.os.AsyncTask;
import android.os.Bundle;
import com.google.android.material.snackbar.Snackbar;
import com.squareup.picasso.Picasso;
import androidx.annotation.NonNull;
import androidx.annotation.Nullable;
import androidx.appcompat.app.AppCompatActivity;
import android.os.UserManager;
import android.text.TextUtils;
import android.util.Log;
import android.view.View;
import android.widget.Button;
import android.widget.ImageView;
import android.widget.TextView;
import android.widget.Toast;
import net.openid.appauth.AuthState;
import net.openid.appauth.AuthorizationException;
import net.openid.appauth.AuthorizationRequest;
import net.openid.appauth.AuthorizationResponse;
import net.openid.appauth.AuthorizationService;
import net.openid.appauth.AuthorizationServiceConfiguration;
import net.openid.appauth.TokenResponse;
import org.json.JSONException;
import org.json.JSONObject;
import java.util.HashMap;
import java.util.Map;
import okhttp3.OkHttpClient;
import okhttp3.Request;
import okhttp3.Response;
import static com.example.sso.MainApplication.LOG_TAG;

public class MainActivity extends AppCompatActivity {
    private static final String SHARED_PREFERENCES_NAME = "AuthStatePreference";
    private static final String AUTH_STATE = "AUTH_STATE";
    private static final String USED_INTENT = "USED_INTENT";
    private static final String LOGIN_HINT = "login_hint";

    MainApplication mMainApplication;
```

```
AuthState mAuthState;
Button mAuthorize, mMakeApiCall, mSignOut;
TextView mGivenName, mFamilyName, mFullName;
ImageView mProfileView;

protected String mLoginHint;

BroadcastReceiver mRestrictionsReceiver;

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

   mMainApplication = (MainApplication) getApplication();
   mAuthorize = findViewById(R.id.authorize);
   mMakeApiCall = findViewById(R.id.makeApiCall);
   mSignOut = findViewById(R.id.signOut);
   mGivenName = findViewById(R.id.givenName);
   mFamilyName =findViewById(R.id.familyName);
   mFullName = findViewById(R.id.fullName);
   mProfileView = findViewById(R.id.profileImage);

   enablePostAuthorizationFlows();
   mAuthorize.setOnClickListener(new AuthorizeListener(this));
   getAppRestrictions();
}

@Override
protected void onResume(){
   super.onResume();

   // Retrieve app restrictions and take appropriate action
   getAppRestrictions();

   // Register a receiver for app restrictions changed broadcast
   registerRestrictionsReceiver();
}
@Override
protected void onStop(){
   super.onStop();
   // Unregister receiver for app restrictions changed broadcast
   unregisterReceiver(mRestrictionsReceiver);
}

@Override
protected void onNewIntent(Intent intent) {
   checkIntent(intent);
}
private void checkIntent(@Nullable Intent intent) {
   if (intent != null) {
      String action = intent.getAction();
```

```
        switch (action) {
            case "com.google.codelabs.appauth.HANDLE_AUTHORIZATION_RESPONSE":
                if (!intent.hasExtra(USED_INTENT)) {
                    handleAuthorizationResponse(intent);
                    intent.putExtra(USED_INTENT, true);
                }
                break;
            default:
                // do nothing
        }
    }
}

@Override
protected void onStart() {
    super.onStart();
    checkIntent(getIntent());
    // Register a receiver for app restrictions changed broadcast
    registerRestrictionsReceiver();
}

private void enablePostAuthorizationFlows() {
    mAuthState = restoreAuthState();
    if (mAuthState != null && mAuthState.isAuthorized()) {
        if (mMakeApiCall.getVisibility() == View.GONE) {
            mMakeApiCall.setVisibility(View.VISIBLE);
            mMakeApiCall.setOnClickListener(new MakeApiCallListener(this, mAuthState, new
AuthorizationService(this)));
        }
        if (mSignOut.getVisibility() == View.GONE) {
            mSignOut.setVisibility(View.VISIBLE);
            mSignOut.setOnClickListener(new SignOutListener(this));
        }
    } else {
        mMakeApiCall.setVisibility(View.GONE);
        mSignOut.setVisibility(View.GONE);
    }
}

private void handleAuthorizationResponse(@NonNull Intent intent) {
    AuthorizationResponse response = AuthorizationResponse.fromIntent(intent);
    AuthorizationException error = AuthorizationException.fromIntent(intent);
    final AuthState authState = new AuthState(response, error);
    if (response != null) {
        Log.i(LOG_TAG, String.format("Handled Authorization Response %s ", authState.toJsonString()));
        AuthorizationService service = new AuthorizationService(this);
        service.performTokenRequest(response.createTokenExchangeRequest(), new
AuthorizationService.TokenResponseCallback() {
            @Override
            public void onTokenRequestCompleted(@Nullable TokenResponse tokenResponse, @Nullable
AuthorizationException exception) {
                if (exception != null) {
```

```
                Log.w(LOG_TAG, "Token Exchange failed", exception);
            } else {
                if (tokenResponse != null) {
                    authState.update(tokenResponse, exception);
                    persistAuthState(authState);
                    Log.i(LOG_TAG, String.format("Token Response [ Access Token: %s, ID Token: %s ]",
tokenResponse.accessToken, tokenResponse.idToken));
                }
            }
        }
    });
  }
}

private void persistAuthState(@NonNull AuthState authState) {
  getSharedPreferences(SHARED_PREFERENCES_NAME, Context.MODE_PRIVATE).edit()
      .putString(AUTH_STATE, authState.toJsonString())
      .commit();
  enablePostAuthorizationFlows();
}

private void clearAuthState() {
  getSharedPreferences(SHARED_PREFERENCES_NAME, Context.MODE_PRIVATE)
      .edit()
      .remove(AUTH_STATE)
      .apply();
}

@Nullable
private AuthState restoreAuthState() {
  String jsonString = getSharedPreferences(SHARED_PREFERENCES_NAME,
Context.MODE_PRIVATE)
      .getString(AUTH_STATE, null);
  if (!TextUtils.isEmpty(jsonString)) {
    try {
      return AuthState.fromJson(jsonString);
    } catch (JSONException jsonException) {
      // should never happen
    }
  }
  return null;
}

public static class AuthorizeListener implements Button.OnClickListener {
  private final MainActivity mMainActivity;

  public AuthorizeListener(@NonNull MainActivity mainActivity) {
    mMainActivity = mainActivity;
  }

  @Override
  public void onClick(View view) {
```

```
    AuthorizationServiceConfiguration serviceConfiguration = new AuthorizationServiceConfiguration(
            Uri.parse("https://accounts.google.com/o/oauth2/v2/auth") /* auth endpoint */,
            Uri.parse("https://www.googleapis.com/oauth2/v4/token") /* token endpoint */
        );
        AuthorizationService authorizationService = new AuthorizationService(view.getContext());
        String clientId = "511828570984-fuprh0cm7665emlne3rnf9pk34kkn86s.apps.googleusercontent.com";
        Uri redirectUri = Uri.parse("com.google.codelabs.appauth:/oauth2callback");
        AuthorizationRequest.Builder builder = new AuthorizationRequest.Builder(
            serviceConfiguration,
            clientId,
            AuthorizationRequest.RESPONSE_TYPE_CODE,
            redirectUri
        );
        builder.setScopes("profile");

        if(mMainActivity.getLoginHint() != null){
          Map loginHintMap = new HashMap<String, String>();
          loginHintMap.put(LOGIN_HINT,mMainActivity.getLoginHint());
          builder.setAdditionalParameters(loginHintMap);

          Log.i(LOG_TAG, String.format("login_hint: %s", mMainActivity.getLoginHint()));
        }

        AuthorizationRequest request = builder.build();
        String action = "com.google.codelabs.appauth.HANDLE_AUTHORIZATION_RESPONSE";
        Intent postAuthorizationIntent = new Intent(action);
        PendingIntent pendingIntent = PendingIntent.getActivity(view.getContext(), request.hashCode(),
postAuthorizationIntent, 0);
        authorizationService.performAuthorizationRequest(request, pendingIntent);
      }
    }

    public static class SignOutListener implements Button.OnClickListener {
      private final MainActivity mMainActivity;

      public SignOutListener(@NonNull MainActivity mainActivity) {
        mMainActivity = mainActivity;
      }
      @Override
      public void onClick(View view) {
        mMainActivity.mAuthState = null;
        mMainActivity.clearAuthState();
        mMainActivity.enablePostAuthorizationFlows();
      }
    }
    public static class MakeApiCallListener implements Button.OnClickListener {
      private final MainActivity mMainActivity;
      private AuthState mAuthState;
      private AuthorizationService mAuthorizationService;

      public MakeApiCallListener(@NonNull MainActivity mainActivity, @NonNull AuthState authState,
@NonNull AuthorizationService authorizationService) {
```

```
        mMainActivity = mainActivity;
        mAuthState = authState;
        mAuthorizationService = authorizationService;
    }

    @Override
    public void onClick(View view) {
        mAuthState.performActionWithFreshTokens(mAuthorizationService, new AuthState.AuthStateAction()
{
            @Override
            public void execute(@Nullable String accessToken, @Nullable String idToken, @Nullable
AuthorizationException exception) {
                new AsyncTask<String, Void, JSONObject>() {
                    @Override
                    protected JSONObject doInBackground(String... tokens) {
                        OkHttpClient client = new OkHttpClient();
                        Request request = new Request.Builder()
                                .url("https://www.googleapis.com/oauth2/v3/userinfo")
                                .addHeader("Authorization", String.format("Bearer %s", tokens[0]))
                                .build();
                        try {
                            Response response = client.newCall(request).execute();
                            String jsonBody = response.body().string();
                            Log.i(LOG_TAG, String.format("User Info Response %s", jsonBody));
                            return new JSONObject(jsonBody);
                        } catch (Exception exception) {
                            Log.w(LOG_TAG, exception);
                        }
                        return null;
                    }

                    @Override
                    protected void onPostExecute(JSONObject userInfo) {
                        if (userInfo != null) {
                            String fullName = userInfo.optString("name", null);
                            String givenName = userInfo.optString("given_name", null);
                            String familyName = userInfo.optString("family_name", null);
                            String imageUrl = userInfo.optString("picture", null);
                            if (!TextUtils.isEmpty(imageUrl)) {
                                Picasso.with(mMainActivity)
                                        .load(imageUrl)
                                        .placeholder(R.drawable.ic_account_circle_black_24dp)
                                        .into(mMainActivity.mProfileView);
                            }
                            if (!TextUtils.isEmpty(fullName)) {
                                mMainActivity.mFullName.setText(fullName);
                            }
                            if (!TextUtils.isEmpty(givenName)) {
                                mMainActivity.mGivenName.setText(givenName);
                            }
                            if (!TextUtils.isEmpty(familyName)) {
                                mMainActivity.mFamilyName.setText(familyName);
```

```
                    }

                    String message;
                    if (userInfo.has("error")) {
                        message = String.format("%s [%s]", mMainActivity.getString(R.string.request_failed),
userInfo.optString("error_description", "No description"));
                    } else {
                        message = mMainActivity.getString(R.string.request_complete);
                    }
                    Snackbar.make(mMainActivity.mProfileView, message, Snackbar.LENGTH_SHORT)
                            .show();
                }
            }
        }.execute(accessToken);
    }
  });
 }
}

private void getAppRestrictions(){
  RestrictionsManager restrictionsManager =
      (RestrictionsManager) this
          .getSystemService(Context.RESTRICTIONS_SERVICE);

  Bundle appRestrictions = restrictionsManager.getApplicationRestrictions();

  // Block user if KEY_RESTRICTIONS_PENDING is true, and save login hint if available
  if(!appRestrictions.isEmpty()){
    if(appRestrictions.getBoolean(UserManager.
        KEY_RESTRICTIONS_PENDING)!=true){
      mLoginHint = appRestrictions.getString(LOGIN_HINT);
    }
    else {
      Toast.makeText(this,R.string.restrictions_pending_block_user,
          Toast.LENGTH_LONG).show();
      finish();
    }
  }
}

private void registerRestrictionsReceiver(){
  IntentFilter restrictionsFilter =
      new IntentFilter(Intent.ACTION_APPLICATION_RESTRICTIONS_CHANGED);

  mRestrictionsReceiver = new BroadcastReceiver() {
    @Override
    public void onReceive(Context context, Intent intent) {
      getAppRestrictions();
    }
  };
  registerReceiver(mRestrictionsReceiver, restrictionsFilter);
}
```

```
    public String getLoginHint(){
        return mLoginHint;
    }
}
```

MainApplication.java

```
import android.app.Application;

public class MainApplication extends Application {
    public static final String LOG_TAG = "AppAuthSample";
    @Override
    public void onCreate() {
        super.onCreate();
    }
}
```

Layout File/s:

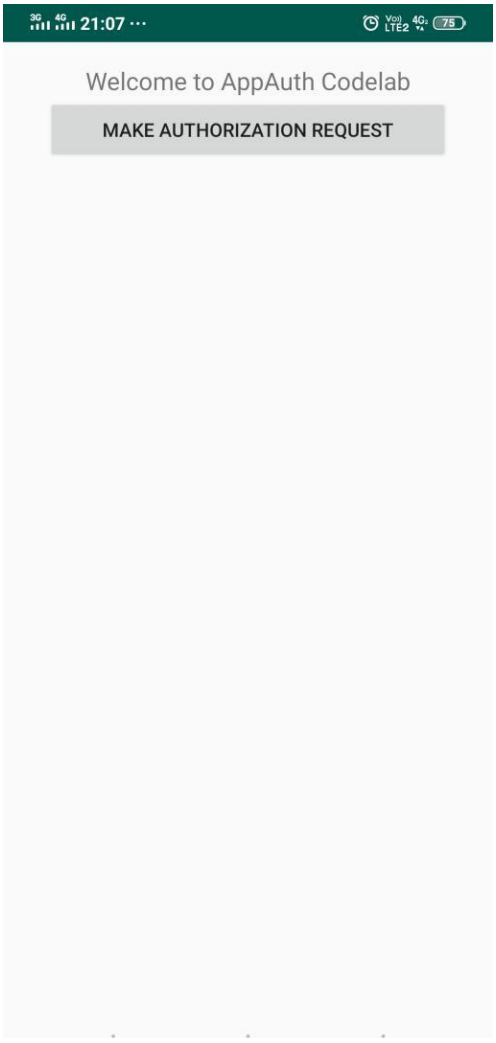
activity_main.xml

```
<?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"
    android:paddingBottom="@dimen/activity_vertical_margin"
    android:paddingLeft="@dimen/activity_horizontal_margin"
    android:paddingRight="@dimen/activity_horizontal_margin"
    android:paddingTop="@dimen/activity_vertical_margin">

    <TextView
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:gravity="center_horizontal"
        android:text="@string/welcome_to_appauth"
        android:textAppearance="?android:textAppearanceMedium"/>

    <Button
        android:id="@+id/authorize"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_marginLeft="@dimen/action_margin"
        android:layout_marginRight="@dimen/action_margin"
        android:text="@string/make_authorization_request"/>

    <Button
        android:id="@+id/makeApiCall"
        android:layout_width="match_parent"
```

```
    android:layout_height="wrap_content"
    android:layout_marginLeft="@dimen/action_margin"
    android:layout_marginRight="@dimen/action_margin"
    android:text="@string/make_api_call"
    android:visibility="gone"/>

  <Button
    android:id="@+id/signOut"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_marginLeft="@dimen/action_margin"
    android:layout_marginRight="@dimen/action_margin"
    android:text="@string/sign_out"
    android:visibility="gone"/>
  <LinearLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_marginLeft="@dimen/action_margin"
    android:layout_marginRight="@dimen/action_margin"
    android:orientation="horizontal">
    <ImageView
      android:id="@+id/profileImage"
      android:layout_width="0dp"
      android:layout_height="wrap_content"
      android:layout_marginLeft="@dimen/action_margin"
      android:layout_marginRight="@dimen/action_margin"
      android:layout_weight="4"
      tools:src="@drawable/ic_account_circle_black_24dp"/>

    <LinearLayout
      android:layout_width="0dp"
      android:layout_height="wrap_content"
      android:layout_gravity="center_vertical"
      android:layout_weight="6"
      android:orientation="vertical">

      <TextView
        android:id="@+id/givenName"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_marginTop="@dimen/action_margin"
        android:textAppearance="?android:textAppearanceSmall"
        tools:text="@string/tools_given_name"/>

      <TextView
        android:id="@+id/familyName"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_marginTop="@dimen/action_margin"
        android:textAppearance="?android:textAppearanceSmall"
        tools:text="@string/tools_family_name"/>
```

```
        <TextView
            android:id="@+id/fullName"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:layout_marginTop="@dimen/action_margin"
            android:textAppearance="?android:textAppearanceSmall"
            tools:text="@string/tools_full_name"/>

    </LinearLayout>
  </LinearLayout>
</LinearLayout>
```

Output:



Practical 12

Aim:Create an application to handle support voice interaction.

Source Code:

Java File/s:

MainActivity.java

```
package com.example.a17it086_p12;


import androidx.appcompat.app.AppCompatActivity;

import android.os.Bundle;
import android.content.ActivityNotFoundException;
import android.content.Intent;
import android.speech.RecognitionService;
import android.speech.RecognizerIntent;
import android.view.View;
import android.widget.TextView;
import android.widget.Toast;
import java.util.ArrayList;
import java.util.Locale;

public class MainActivity extends AppCompatActivity {
   private TextView resultText;
   @Override
   protected void onCreate(Bundle savedInstanceState) {
      super.onCreate(savedInstanceState);
      setContentView(R.layout.activity_main);
      resultText = (TextView) findViewById(R.id.textView);
   }
   public void onButtonClick(View v)
   { if(v.getId() == R.id.imageButton5
   )
   {
      promptSpeechInput();
   }
   }
   private void promptSpeechInput()
   {
         Intent i = new Intent(RecognizerIntent.ACTION_RECOGNIZE_SPEECH);
         i.putExtra(RecognizerIntent.EXTRA_LANGUAGE_MODEL
,RecognizerIntent.LANGUAGE_MODEL_FREE_FORM);
         i.putExtra(RecognizerIntent.EXTRA_LANGUAGE , Locale.getDefault());
         i.putExtra(RecognizerIntent.EXTRA_PROMPT , "Say Something!!");
      try {
         startActivityForResult(i,100);
      }
```

```
        catch (ActivityNotFoundException e)
        {
            Toast.makeText(MainActivity.this , "Sorry! Your device doesn't speech
language!",Toast.LENGTH_LONG).show();
        }
    }
    public void onActivityResult(int request_code,int result_code,Intent i)
    {
        super.onActivityResult(request_code,request_code,i);
        switch(request_code)
            {
                case 100:
                    if(result_code == RESULT_OK && i != null)
                    {
                        ArrayList<String> result = i.getStringArrayListExtra(RecognizerIntent.EXTRA_RESULTS);
                         resultText.setText(result.get(0));
                    }break;
            }
    }
}
```

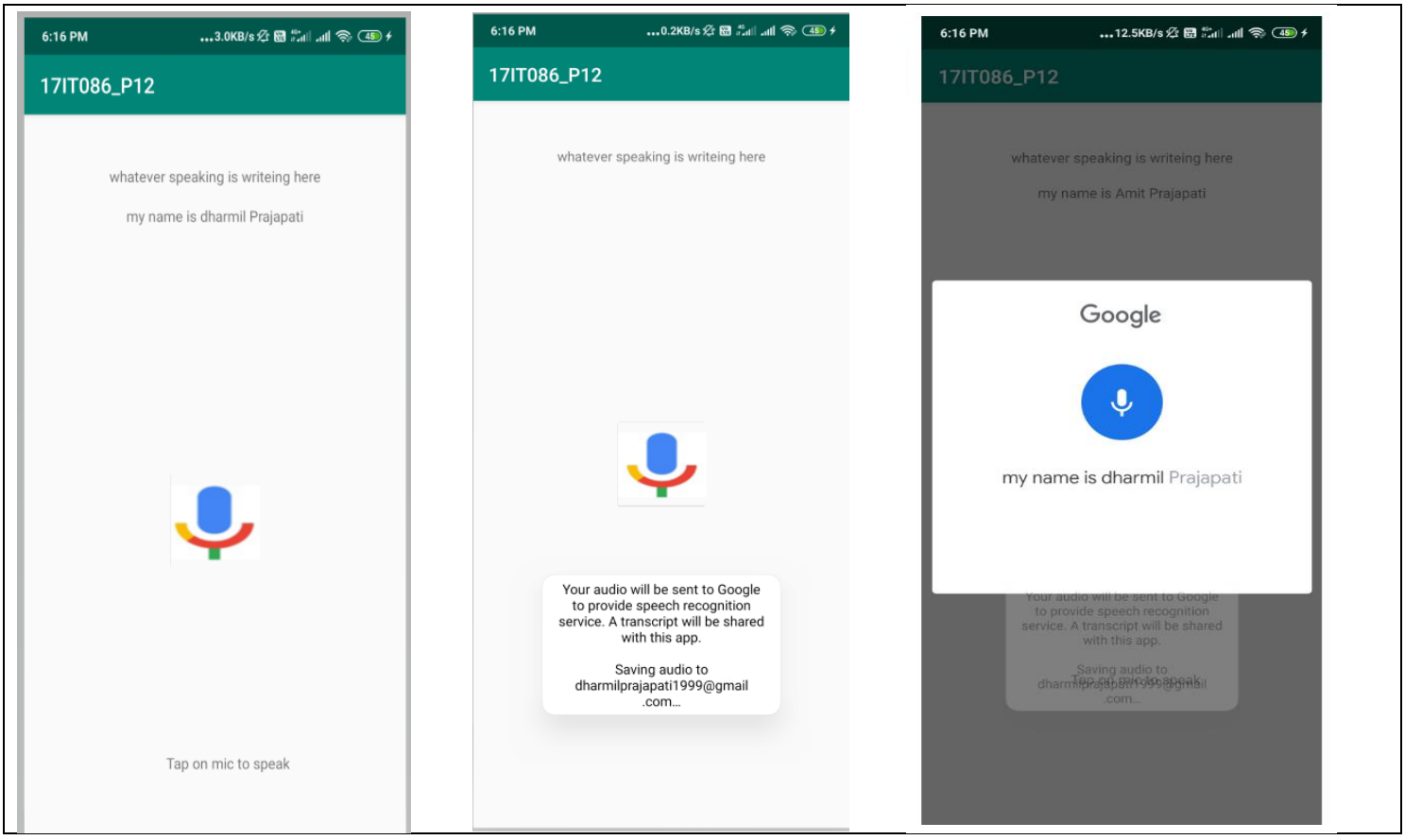
Layout File/s:

activity_main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
xmlns:app="http://schemas.android.com/apk/res-auto" xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".MainActivity">
    <TextView android:id="@+id/textView"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignParentTop="true"
        android:layout_centerHorizontal="true"
        android:layout_marginTop="85dp"
        tools:layout_editor_absoluteX="154dp"
        tools:layout_editor_absoluteY="84dp" />
    <ImageButton android:id="@+id/imageButton5"
        android:layout_width="100sp"
        android:layout_height="100sp"
        android:layout_centerInParent="true"
        android:onClick="onButtonClick"
        app:srcCompat="@drawable/voice"
        tools:layout_editor_absoluteX="132dp"
        tools:layout_editor_absoluteY="205dp"
        tools:ignore="OnClick" />
    <TextView android:id="@+id/textView2"
        android:layout_width="wrap_content"
        android:layout_height="19dp"
```

```
      android:layout_alignStart="@+id/imageButton5"
      android:layout_alignParentBottom="true"
      android:layout_marginBottom="144dp"
      android:text="Tap on mic to speak"
      tools:layout_editor_absoluteX="132dp"
      tools:layout_editor_absoluteY="382dp"
      android:layout_alignLeft="@+id/imageButton5" />
   <TextView
    android:id="@+id/textView3"
      android:layout_width="wrap_content"
      android:layout_height="wrap_content"
      android:layout_alignParentTop="true"
      android:layout_centerHorizontal="true"
      android:layout_marginTop="48dp"
      android:text="whatever speaking is writeing here" />
</RelativeLayout>
```

Output:



PRACTICAL:13

AIM:Create an application to play video using the YouTube API in PIP mode.

Source Code:

Program: activity_main.xml

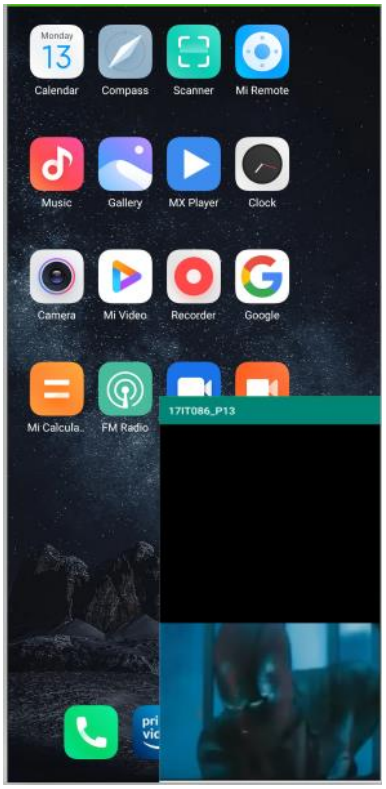
```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout
   xmlns:android="http://schemas.android.com/apk/res/android"
   xmlns:app="http://schemas.android.com/apk/res-auto"
   xmlns:tools="http://schemas.android.com/tools"
   android:layout_width="match_parent"
   android:layout_height="match_parent"
   tools:context=".MainActivity">
<VideoView
      android:layout_width="match_parent"
      android:layout_height="match_parent"
      android:id="@+id/video"
      android:layout_above="@id/pipbtn"/>
<Button
      android:layout_width="match_parent"
      android:layout_height="wrap_content"
      android:text="Enter PIP mode"
      android:layout_alignParentBottom="true"
      android:id="@+id/pipbtn"/>
</RelativeLayout>
```

MainActivity.java

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

import android.app.ActionBar;
import android.app.Notification;
import android.app.PictureInPictureParams;
import android.drm.DrmStore;
import android.graphics.Point;
import android.net.Uri;
import android.os.Bundle;
import android.util.Rational;
import android.view.Display;
import android.view.View;
import android.widget.Button;
import android.widget.MediaController;
import android.widget.VideoView;

public class MainActivity extends AppCompatActivity {

   Button pipbtn;
   String path = "/storage/DCIM/Camera/movie.mp4";
   ActionBar actionBar;
   VideoView video;
```

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

        video = (VideoView)findViewById(R.id.video);
        actionBar = getActionBar();
        MediaController mediaController= new MediaController(this);
        mediaController.setAnchorView(video);
        video.setMediaController(mediaController);
        video.setVideoURI(Uri.parse(path));
        video.requestFocus();
        video.start();

        pipbtn = (Button)findViewById(R.id.pipbtn);

        pipbtn.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                Display display = getWindowManager().getDefaultDisplay();
                Point point = new Point();
                PictureInPictureParams.Builder pip_builder = new PictureInPictureParams.Builder();
                pip_builder.setAspectRatio(ratio).build();
                pipbtn.setVisibility(View.INVISIBLE);
                enterPictureInPictureMode(pip_builder.build());
            }
        });
}
}
```

Output:



Practical 14

AIM: Create an application that uses end-to-end process of training a machine learning model that can recognize handwritten digit images with TensorFlow and deploy it to an Android app.

Source Code:

Kotlin File/s:

MainActivity.kt

```
package org.tensorflow.lite.codelabs.digitclassifier
import android.annotation.SuppressLint
import android.graphics.Color
import android.os.Bundle
import android.util.Log
import android.view.MotionEvent
import android.widget.Button
import android.widget.TextView
import androidx.appcompat.app.AppCompatActivity
import com.divyanshu.draw.widget.DrawView

class MainActivity : AppCompatActivity() {
  private var drawView: DrawView? = null
  private var clearButton: Button? = null
  private var predictedTextView: TextView? = null
  private var digitClassifier = DigitClassifier(this)

  @SuppressLint("ClickableViewAccessibility")
  override fun onCreate(savedInstanceState: Bundle?) {
    super.onCreate(savedInstanceState)
    setContentView(R.layout.activity_main)

    // Setup view instances.
    drawView = findViewById(R.id.draw_view)
    drawView?.setStrokeWidth(70.0f)
    drawView?.setColor(Color.WHITE)
    drawView?.setBackgroundColor(Color.BLACK)
    clearButton = findViewById(R.id.clear_button)
    predictedTextView = findViewById(R.id.predicted_text)

    // Setup clear drawing button.
    clearButton?.setOnClickListener {
      drawView?.clearCanvas()
      predictedTextView?.text = getString(R.string.prediction_text_placeholder)
    }

    // Setup classification trigger so that it classify after every stroke drew.
    drawView?.setOnTouchListener { _, event ->
      // As we have interrupted DrawView's touch event,
      // we first need to pass touch events through to the instance for the drawing to show up.
      drawView?.onTouchEvent(event)
      // Then if user finished a touch event, run classification
```

```
    if (event.action == MotionEvent.ACTION_UP) {
      classifyDrawing()
    }
    true
  }
  // Setup digit classifier.
  digitClassifier
    .initialize()
    .addOnFailureListener { e -> Log.e(TAG, "Error to setting up digit classifier.", e) }
 }
 override fun onDestroy() {
  // Sync DigitClassifier instance lifecycle with MainActivity lifecycle,
  // and free up resources (e.g. TF Lite instance) once the activity is destroyed.
  digitClassifier.close()
  super.onDestroy()
 }

 private fun classifyDrawing() {
  val bitmap = drawView?.getBitmap()
  if ((bitmap != null) && (digitClassifier.isInitialized)) {
    digitClassifier
      .classifyAsync(bitmap)
      .addOnSuccessListener { resultText -> predictedTextView?.text = resultText }
      .addOnFailureListener { e ->
        predictedTextView?.text = getString(
          R.string.classification_error_message,
          e.localizedMessage
        )
        Log.e(TAG, "Error classifying drawing.", e)
      }
  }
 }
 companion object {
  private const val TAG = "MainActivity"
 }
}
```

DigitClassifier.kt

```
package org.tensorflow.lite.codelabs.digitclassifier

import android.content.Context
import android.content.res.AssetManager
import android.graphics.Bitmap
import android.util.Log
import com.google.android.gms.tasks.Task
import com.google.android.gms.tasks.Tasks.call
import java.io.FileInputStream
import java.io.IOException
import java.nio.ByteBuffer
import java.nio.ByteOrder
```

```
import java.nio.channels.FileChannel
import java.util.concurrent.Callable
import java.util.concurrent.ExecutorService
import java.util.concurrent.Executors
import org.tensorflow.lite.Interpreter

class DigitClassifier(private val context: Context) {
  // TODO: Add a TF Lite interpreter as a field.
  private var interpreter: Interpreter? = null
  var isInitialized = false
    private set

  /** Executor to run inference task in the background. */
  private val executorService: ExecutorService = Executors.newCachedThreadPool()

  private var inputImageWidth: Int = 0 // will be inferred from TF Lite model.
  private var inputImageHeight: Int = 0 // will be inferred from TF Lite model.
  private var modelInputSize: Int = 0 // will be inferred from TF Lite model.

  fun initialize(): Task<Void> {
    return call(
      executorService,
      Callable<Void> {
        initializeInterpreter()
        null
      }
    )}
  @Throws(IOException::class)
  private fun initializeInterpreter() {
    // TODO: Load the TF Lite model from file and initialize an interpreter.

    // Load the TF Lite model from asset folder and initialize TF Lite Interpreter with NNAPI enabled.
    val assetManager = context.assets
    val model = loadModelFile(assetManager, "mnist_17it085.tflite")
    val options = Interpreter.Options()
    options.setUseNNAPI(true)
    val interpreter = Interpreter(model, options)

    // Read input shape from model file.
    val inputShape = interpreter.getInputTensor(0).shape()
    inputImageWidth = inputShape[1]
    inputImageHeight = inputShape[2]
    modelInputSize = FLOAT_TYPE_SIZE * inputImageWidth *
            inputImageHeight * PIXEL_SIZE
    // Finish interpreter initialization.
    this.interpreter = interpreter
    isInitialized = true
    Log.d(TAG, "Initialized TFLite interpreter.")
  }

  @Throws(IOException::class)
  private fun loadModelFile(assetManager: AssetManager, filename: String): ByteBuffer {
```

```
  val fileDescriptor = assetManager.openFd(filename)
  val inputStream = FileInputStream(fileDescriptor.fileDescriptor)
  val fileChannel = inputStream.channel
  val startOffset = fileDescriptor.startOffset
  val declaredLength = fileDescriptor.declaredLength
  return fileChannel.map(FileChannel.MapMode.READ_ONLY, startOffset, declaredLength)
 }

 private fun classify(bitmap: Bitmap): String {
  check(isInitialized) { "TF Lite Interpreter is not initialized yet." }
  // TODO: Add code to run inference with TF Lite.
  // Pre-processing: resize the input image to match the model input shape.
  val resizedImage = Bitmap.createScaledBitmap(
    bitmap,
    inputImageWidth,
    inputImageHeight,
    true
  )
  val byteBuffer = convertBitmapToByteBuffer(resizedImage)
  // Define an array to store the model output.
  val output = Array(1) { FloatArray(OUTPUT_CLASSES_COUNT) }

  // Run inference with the input data.
  interpreter?.run(byteBuffer, output)
  // Post-processing: find the digit that has the highest probability
  // and return it a human-readable string.
  val result = output[0]
  val maxIndex = result.indices.maxBy { result[it] } ?: -1
  val resultString =
    "Prediction Result: %d\nConfidence: %2f"
      .format(maxIndex, result[maxIndex])
  return resultString
 }
 fun classifyAsync(bitmap: Bitmap): Task<String> {
  return call(executorService, Callable<String> { classify(bitmap) })
 }
 fun close() {
  call(
    executorService,
    Callable<String> {
      // TODO: close the TF Lite interpreter here
      interpreter?.close()
      Log.d(TAG, "Closed TFLite interpreter.")
      null
    }
  )
 }
 private fun convertBitmapToByteBuffer(bitmap: Bitmap): ByteBuffer {
  val byteBuffer = ByteBuffer.allocateDirect(modelInputSize)
  byteBuffer.order(ByteOrder.nativeOrder())
  val pixels = IntArray(inputImageWidth * inputImageHeight)
  bitmap.getPixels(pixels, 0, bitmap.width, 0, 0, bitmap.width, bitmap.height)
```

```
    for (pixelValue in pixels) {
      val r = (pixelValue shr 16 and 0xFF)
      val g = (pixelValue shr 8 and 0xFF)
      val b = (pixelValue and 0xFF)

      // Convert RGB to grayscale and normalize pixel value to [0..1].
      val normalizedPixelValue = (r + g + b) / 3.0f / 255.0f
      byteBuffer.putFloat(normalizedPixelValue)
    }
    return byteBuffer
  }
  companion object {
    private const val TAG = "DigitClassifier"
    private const val FLOAT_TYPE_SIZE = 4
    private const val PIXEL_SIZE = 1
    private const val OUTPUT_CLASSES_COUNT = 10
  }
}
```

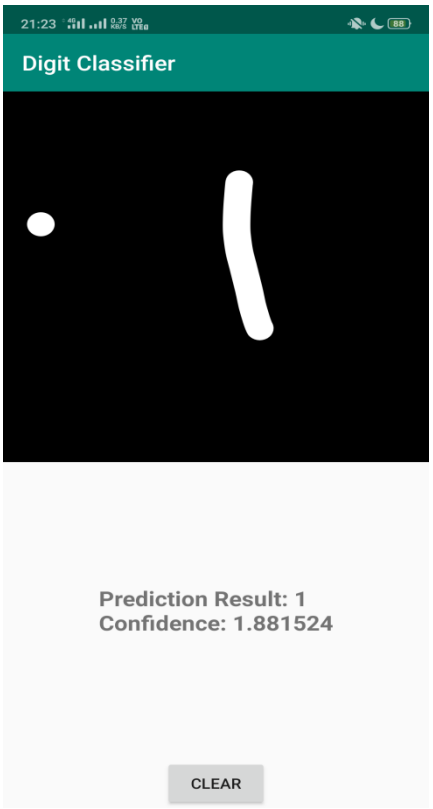
Layout File/s:

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">
  <com.divyanshu.draw.widget.DrawView
      android:id="@+id/draw_view"
      android:layout_width="match_parent"
      android:layout_height="0dp"
      app:layout_constraintDimensionRatio="1:1"
      app:layout_constraintTop_toTopOf="parent"/>
  <TextView
      android:id="@+id/predicted_text"
      android:textStyle="bold"
      android:layout_width="wrap_content"
      android:layout_height="wrap_content"
      android:text="@string/prediction_text_placeholder"
      android:textSize="20sp"
      app:layout_constraintBottom_toTopOf="@id/clear_button"
      app:layout_constraintLeft_toLeftOf="parent"
      app:layout_constraintRight_toRightOf="parent"
      app:layout_constraintTop_toBottomOf="@id/draw_view"/>
  <Button
```

```
    android:id="@+id/clear_button"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="@string/clear_button_text"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintLeft_toLeftOf="parent"
    app:layout_constraintRight_toRightOf="parent"/>
</androidx.constraintlayout.widget.ConstraintLayout>
```

Output:



Practical 15

Aim: Develop project in android.

Source Code:

Java File/s:

MainActivity.java

```
package com.google.firebase.codelab.mlkit;
import android.app.Activity;
import android.content.Context;
import android.content.Intent;
import android.content.res.AssetManager;
import android.graphics.Bitmap;
import android.graphics.BitmapFactory;
import android.net.Uri;
import android.os.Bundle;
import android.provider.MediaStore;
import androidx.annotation.NonNull;
import androidx.appcompat.app.AppCompatActivity;
import android.util.Log;
import android.util.Pair;
import android.view.View;
import android.widget.AdapterView;
import android.widget.ArrayAdapter;
import android.widget.Button;
import android.widget.ImageView;
import android.widget.Spinner;
import android.widget.Toast;

import androidx.annotation.Nullable;

import com.google.android.gms.tasks.Continuation;
import com.google.android.gms.tasks.OnFailureListener;
import com.google.android.gms.tasks.Task;
import com.google.firebase.ml.common.FirebaseMLException;
import com.google.firebase.ml.common.modeldownload.FirebaseLocalModel;
import com.google.firebase.ml.common.modeldownload.FirebaseModelDownloadConditions;
import com.google.firebase.ml.common.modeldownload.FirebaseModelManager;
import com.google.firebase.ml.common.modeldownload.FirebaseRemoteModel;
import com.google.firebase.ml.custom.FirebaseModelDataType;
import com.google.firebase.ml.custom.FirebaseModelInputOutputOptions;
import com.google.firebase.ml.custom.FirebaseModelInputs;
import com.google.firebase.ml.custom.FirebaseModelInterpreter;
import com.google.firebase.ml.custom.FirebaseModelOptions;
import com.google.firebase.ml.custom.FirebaseModelOutputs;

import java.io.BufferedReader;
import java.io.IOException;
import java.io.InputStream;
```

```
import java.io.InputStreamReader;
import java.nio.ByteBuffer;
import java.nio.ByteOrder;
import java.util.AbstractMap;
import java.util.ArrayList;
import java.util.Comparator;
import java.util.List;
import java.util.Map;
import java.util.PriorityQueue;

public class MainActivity extends AppCompatActivity implements AdapterView.OnItemSelectedListener {
   private static final String TAG = "MainActivity";
   private ImageView mImageView;
   private Button mRunCustomModelButton;
   private Bitmap mSelectedImage;
   private GraphicOverlay mGraphicOverlay;

   Button button;
   private static final int PICK_IMAGE = 100;
   Uri imgUri1;
   Spinner sp;

      private Integer mImageMaxWidth;
   // Max height (portrait mode)
   private Integer mImageMaxHeight;

   private static final String HOSTED_MODEL_NAME = "cloud_model_1";
   private static final String LOCAL_MODEL_ASSET = "mobilenet_v1_1.0_224_quant.tflite";

   private static final String LABEL_PATH = "labels.txt";

   private static final int RESULTS_TO_SHOW = 3;
      private static final int DIM_BATCH_SIZE = 1;
   private static final int DIM_PIXEL_SIZE = 3;
   private static final int DIM_IMG_SIZE_X = 224;
   private static final int DIM_IMG_SIZE_Y = 224;

   private List<String> mLabelList;
   private final PriorityQueue<Map.Entry<String, Float>> sortedLabels =
         new PriorityQueue<>(
               RESULTS_TO_SHOW,
               new Comparator<Map.Entry<String, Float>>() {
                  @Override
                  public int compare(Map.Entry<String, Float> o1, Map.Entry<String, Float>
                        o2) {
                     return (o1.getValue()).compareTo(o2.getValue());
                  }
               });
     private final int[] intValues = new int[DIM_IMG_SIZE_X * DIM_IMG_SIZE_Y];

   private FirebaseModelInterpreter mInterpreter;
```

```
    private FirebaseModelInputOutputOptions mDataOptions;

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

        mImageView = findViewById(R.id.image_view);
        button = (Button)findViewById(R.id.btnSelectImage);
        sp = (Spinner)findViewById(R.id.spinner);

        button.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
              openGallary();
            }
        });

        mRunCustomModelButton = findViewById(R.id.button_run_custom_model);

        mGraphicOverlay = findViewById(R.id.graphic_overlay);

        mRunCustomModelButton.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                runModelInference();
            }
        });
        Spinner dropdown = findViewById(R.id.spinner);
        String[] items = new String[]{"Test Image 1 (Text)", "Test Image 2 (Text)", "Test Image 3" +
                " (Face)", "Test Image 4 (Object)", "Test Image 5 (Object)","Test Image 6 (object)"};
        ArrayAdapter<String> adapter = new ArrayAdapter<>(this, android.R.layout
                .simple_spinner_dropdown_item, items);
        dropdown.setAdapter(adapter);
        dropdown.setOnItemSelectedListener(this);
        initCustomModel();
    }

    private void openGallary(){
        Intent gallary = new Intent(Intent.ACTION_PICK,
MediaStore.Images.Media.INTERNAL_CONTENT_URI);
        startActivityForResult(gallary,PICK_IMAGE);
    }

    @Override
    protected void onActivityResult(int requestCode, int resultCode, @Nullable Intent data) {
        super.onActivityResult(requestCode, resultCode, data);
        if (resultCode == RESULT_OK && requestCode ==PICK_IMAGE){
            imgUri1 = data.getData();
            mImageView.setImageURI(imgUri1);
            try {
                Bitmap bitmap = MediaStore.Images.Media.getBitmap(this.getContentResolver(), imgUri1);
```

```
            } catch (IOException e) {
                e.printStackTrace();
            }
        }
    }

    private void initCustomModel() {
        mLabelList = loadLabelList(this);

        int[] inputDims = {DIM_BATCH_SIZE, DIM_IMG_SIZE_X, DIM_IMG_SIZE_Y, DIM_PIXEL_SIZE};
        int[] outputDims = {DIM_BATCH_SIZE, mLabelList.size()};
        try {
            mDataOptions =
                    new FirebaseModelInputOutputOptions.Builder() //Configurations for data types and dimensions of
input and output data.
                            .setInputFormat(0, FirebaseModelDataType.BYTE, inputDims)
                            .setOutputFormat(0, FirebaseModelDataType.BYTE, outputDims)
                            .build();
            FirebaseModelDownloadConditions conditions = new FirebaseModelDownloadConditions
//Configurations for data types and dimensions of input and output data.
                    .Builder()
                    .requireWifi()
                    .build();
            FirebaseRemoteModel remoteModel = new FirebaseRemoteModel.Builder
                    (HOSTED_MODEL_NAME) //Describes a remote model to be downloaded to the device.
                    .enableModelUpdates(true) //Enables the download of model updates.
                    .setInitialDownloadConditions(conditions) //Sets the conditions for initial model download
                    .setUpdatesDownloadConditions(conditions)  // Sets the conditions for downloading the model
updates.

                    .build();
            FirebaseLocalModel localModel =
                    new FirebaseLocalModel.Builder("asset")
                            .setAssetFilePath(LOCAL_MODEL_ASSET).build(); //Manages the registration of remote
and local models.
            FirebaseModelManager manager = FirebaseModelManager.getInstance();
            manager.registerRemoteModel(remoteModel);
            manager.registerLocalModel(localModel);
            FirebaseModelOptions modelOptions =
                    new FirebaseModelOptions.Builder()
                            .setRemoteModelName(HOSTED_MODEL_NAME)
                            .setLocalModelName("asset")
                            .build();
            mInterpreter = FirebaseModelInterpreter.getInstance(modelOptions);
        } catch (FirebaseMLException e) {
            showToast("Error while setting up the model");
            e.printStackTrace();
        }
    }

    private void runModelInference() {
        if (mInterpreter == null) {
```

```
        Log.e(TAG, "Image classifier has not been initialized; Skipped.");
        return;
    }
    // Create input data.
    ByteBuffer imgData = convertBitmapToByteBuffer(mSelectedImage, mSelectedImage.getWidth(),
            mSelectedImage.getHeight());

    try {
        FirebaseModelInputs inputs = new FirebaseModelInputs.Builder().add(imgData).build();
        // Here's where the magic happens!!
        mInterpreter
                .run(inputs, mDataOptions)
                .addOnFailureListener(new OnFailureListener() {
                    @Override
                    public void onFailure(@NonNull Exception e) {
                        e.printStackTrace();
                        showToast("Error running model inference");
                    }
                })
                .continueWith(
                        new Continuation<FirebaseModelOutputs, List<String>>() {
                            @Override
                            public List<String> then(Task<FirebaseModelOutputs> task) {
                                byte[][] labelProbArray = task.getResult()
                                        .<byte[][]>getOutput(0);
                                List<String> topLabels = getTopLabels(labelProbArray);
                                mGraphicOverlay.clear();
                                GraphicOverlay.Graphic labelGraphic = new LabelGraphic
                                        (mGraphicOverlay, topLabels);
                                mGraphicOverlay.add(labelGraphic);
                                return topLabels;
                            }
                        });
    } catch (FirebaseMLException e) {
        e.printStackTrace();
        showToast("Error running model inference");
    }

}

 private synchronized List<String> getTopLabels(byte[][] labelProbArray) {
    for (int i = 0; i < mLabelList.size(); ++i) {
        sortedLabels.add(
                new AbstractMap.SimpleEntry<>(mLabelList.get(i), (labelProbArray[0][i] &
                        0xff) / 255.0f));
        if (sortedLabels.size() > RESULTS_TO_SHOW) {
            sortedLabels.poll();
        }
    }
    List<String> result = new ArrayList<>();
    final int size = sortedLabels.size();
    for (int i = 0; i < size; ++i) {
```

```
        Map.Entry<String, Float> label = sortedLabels.poll();
        result.add(label.getKey() + ":" + label.getValue());
      }
      Log.d(TAG, "labels: " + result.toString());
      return result;
    }


    private List<String> loadLabelList(Activity activity) {
      List<String> labelList = new ArrayList<>();
      try (BufferedReader reader =
               new BufferedReader(new InputStreamReader(activity.getAssets().open
                   (LABEL_PATH)))) {
        String line;
        while ((line = reader.readLine()) != null) {
          labelList.add(line);
        }
      } catch (IOException e) {
        Log.e(TAG, "Failed to read label list.", e);
      }
      return labelList;
    }


    private synchronized ByteBuffer convertBitmapToByteBuffer(
        Bitmap bitmap, int width, int height) {
      ByteBuffer imgData =
          ByteBuffer.allocateDirect(
              DIM_BATCH_SIZE * DIM_IMG_SIZE_X * DIM_IMG_SIZE_Y * DIM_PIXEL_SIZE);
      imgData.order(ByteOrder.nativeOrder());
      Bitmap scaledBitmap = Bitmap.createScaledBitmap(bitmap, DIM_IMG_SIZE_X, DIM_IMG_SIZE_Y,
          true);
      imgData.rewind(); //makes a buffer ready for re-reading the data that it already contains: It leaves the limit
unchanged and sets the position to zero.
      scaledBitmap.getPixels(intValues, 0, scaledBitmap.getWidth(), 0, 0,
          scaledBitmap.getWidth(), scaledBitmap.getHeight());
      // Convert the image to int points.
      int pixel = 0;
      for (int i = 0; i < DIM_IMG_SIZE_X; ++i) {
        for (int j = 0; j < DIM_IMG_SIZE_Y; ++j) {
          final int val = intValues[pixel++];
          imgData.put((byte) ((val >> 16) & 0xFF));
          imgData.put((byte) ((val >> 8) & 0xFF));
          imgData.put((byte) (val & 0xFF));
        }
      }
      return imgData;
    }

    private void showToast(String message) {
      Toast.makeText(getApplicationContext(), message, Toast.LENGTH_SHORT).show();
    }
```

```
// Functions for loading images from app assets.

// Returns max image width, always for portrait mode. Caller needs to swap width / height for
// landscape mode.
private Integer getImageMaxWidth() {
   if (mImageMaxWidth == null) {
      // Calculate the max width in portrait mode. This is done lazily since we need to
      // wait for
      // a UI layout pass to get the right values. So delay it to first time image
      // rendering time.
      mImageMaxWidth = mImageView.getWidth();
   }

   return mImageMaxWidth;
}

// Returns max image height, always for portrait mode. Caller needs to swap width / height for
// landscape mode.
private Integer getImageMaxHeight() {
   if (mImageMaxHeight == null) {
      // Calculate the max width in portrait mode. This is done lazily since we need to
      // wait for

      mImageMaxHeight =
            mImageView.getHeight();
   }

   return mImageMaxHeight;
}

// Gets the targeted width / height.
private Pair<Integer, Integer> getTargetedWidthHeight() {
   int targetWidth;
   int targetHeight;
   int maxWidthForPortraitMode = getImageMaxWidth();
   int maxHeightForPortraitMode = getImageMaxHeight();
   targetWidth = maxWidthForPortraitMode;
   targetHeight = maxHeightForPortraitMode;
   return new Pair<>(targetWidth, targetHeight);
}

public void onItemSelected(AdapterView<?> parent, View v, int position, long id) {
   mGraphicOverlay.clear();
   switch (position) {
      case 0:
         mSelectedImage = getBitmapFromAsset(this, "Please_walk_on_the_grass.jpg");
         break;
      case 1:
         // Whatever you want to happen when the thrid item gets selected
         mSelectedImage = getBitmapFromAsset(this, "nl2.jpg");
         break;
```

```
        case 2:
            // Whatever you want to happen when the thrid item gets selected
            mSelectedImage = getBitmapFromAsset(this, "grace_hopper.jpg");
            break;
        case 3:
            // Whatever you want to happen when the thrid item gets selected
            mSelectedImage = getBitmapFromAsset(this, "tennis.jpg");
            break;
        case 4:
            // Whatever you want to happen when the thrid item gets selected
            mSelectedImage = getBitmapFromAsset(this, "mountain.jpg");
            break;
        case 5:
            // Whatever you want to happen when the thrid item gets selected
            mSelectedImage = getBitmapFromAsset(this, "website.jpg");
            break;
        case 6:
            try {
                mSelectedImage = MediaStore.Images.Media.getBitmap(this.getContentResolver(), imgUri1);
            } catch (IOException e) {
                e.printStackTrace();
            }
            ;
    }
    if (mSelectedImage != null) {
        // Get the dimensions of the View
        Pair<Integer, Integer> targetedSize = getTargetedWidthHeight();

        int targetWidth = targetedSize.first;
        int maxHeight = targetedSize.second;

        // Determine how much to scale down the image
        float scaleFactor =
                Math.max(
                        (float) mSelectedImage.getWidth() / (float) targetWidth,
                        (float) mSelectedImage.getHeight() / (float) maxHeight);

        Bitmap resizedBitmap =
                Bitmap.createScaledBitmap(
                        mSelectedImage,
                        (int) (mSelectedImage.getWidth() / scaleFactor),
                        (int) (mSelectedImage.getHeight() / scaleFactor),
                        true);

        mImageView.setImageBitmap(resizedBitmap);
        mSelectedImage = resizedBitmap;
    }
}

@Override
public void onNothingSelected(AdapterView<?> parent) {
    // Do nothing
```

```
    }

    public static Bitmap getBitmapFromAsset(Context context, String filePath) {
        AssetManager assetManager = context.getAssets();

        InputStream is;
        Bitmap bitmap = null;
        try {
            is = assetManager.open(filePath);
            bitmap = BitmapFactory.decodeStream(is);
        } catch (IOException e) {
            e.printStackTrace();
        }

        return bitmap;
    }
}
```

CloudTextGraphic.java

```
package com.google.firebase.codelab.mlkit;

import android.graphics.Canvas;
import android.graphics.Color;
import android.graphics.Paint;
import android.graphics.Rect;

import com.google.firebase.codelab.mlkit.GraphicOverlay.Graphic;
import com.google.firebase.ml.vision.document.FirebaseVisionDocumentText;

import java.util.List;

/**
 * Graphic instance for rendering TextBlock position, size, and ID within an associated graphic
 * overlay view.
 */
public class CloudTextGraphic extends Graphic {
    private static final int TEXT_COLOR = Color.GREEN;
    private static final float TEXT_SIZE = 60.0f;
    private static final float STROKE_WIDTH = 5.0f;

    private final Paint rectPaint;
    private final Paint textPaint;
    private final FirebaseVisionDocumentText.Word word;
    private final GraphicOverlay overlay;

    CloudTextGraphic(GraphicOverlay overlay, FirebaseVisionDocumentText.Word word) {
        super(overlay);

        this.word = word;
        this.overlay = overlay;

        rectPaint = new Paint();
```

```
        rectPaint.setColor(TEXT_COLOR);
        rectPaint.setStyle(Paint.Style.STROKE);
        rectPaint.setStrokeWidth(STROKE_WIDTH);

        textPaint = new Paint();
        textPaint.setColor(TEXT_COLOR);
        textPaint.setTextSize(TEXT_SIZE);
        // Redraw the overlay, as this graphic has been added.
        postInvalidate();
    }

    /**
     * Draws the text block annotations for position, size, and raw value on the supplied canvas.
     */
    @Override
    public void draw(Canvas canvas) {
        if (word == null) {
            throw new IllegalStateException("Attempting to draw a null text.");
        }

        float x = overlay.getWidth() / 4.0f;
        float y = overlay.getHeight() / 4.0f;

        StringBuilder wordStr = new StringBuilder();
        Rect wordRect = word.getBoundingBox();
        canvas.drawRect(wordRect, rectPaint);
        List<FirebaseVisionDocumentText.Symbol> symbols = word.getSymbols();
        for (int m = 0; m < symbols.size(); m++) {
            wordStr.append(symbols.get(m).getText());
        }
        canvas.drawText(wordStr.toString(), wordRect.left, wordRect.bottom, textPaint);
    }
}
```

LabelGraphic.java

```
package com.google.firebase.codelab.mlkit;

import android.graphics.Canvas;
import android.graphics.Color;
import android.graphics.Paint;
import android.graphics.Rect;
import android.text.TextPaint;

import java.util.List;

/**
 * Graphic instance for rendering image labels.
 */
public class LabelGraphic extends GraphicOverlay.Graphic {

    private final Paint textPaint;
```

```
    private final Paint bgPaint;
    private final GraphicOverlay overlay;

    private List<String> labels;

    LabelGraphic(GraphicOverlay overlay, List<String> labels) {
        super(overlay);
        this.overlay = overlay;
        this.labels = labels;
        textPaint = new Paint();
        textPaint.setColor(Color.WHITE);
        textPaint.setTextSize(60.0f);
        bgPaint = new Paint();
        bgPaint.setColor(Color.BLACK);
        bgPaint.setAlpha(50);
    }

    @Override
    public synchronized void draw(Canvas canvas) {
        float x = overlay.getWidth() / 4.0f;
        float y = overlay.getHeight() / 4.0f;

        for (String label : labels) {
            drawTextWithBackground(label, x, y, new TextPaint(textPaint), bgPaint, canvas);
            y = y - 62.0f;
        }
    }

    private void drawTextWithBackground(String text, float x, float y, TextPaint paint,
                                        Paint bgPaint, Canvas canvas) {
        Paint.FontMetrics fontMetrics = paint.getFontMetrics();
        canvas.drawRect(new Rect((int) (x), (int) (y + fontMetrics.top),
                (int) (x + paint.measureText(text)), (int) (y + fontMetrics.bottom)), bgPaint);
        canvas.drawText(text, x, y, textPaint);
    }
}
```

TextGraphic.java

```
package com.google.firebase.codelab.mlkit;

import android.graphics.Canvas;
import android.graphics.Color;
import android.graphics.Paint;
import android.graphics.RectF;
import android.util.Log;

import com.google.firebase.ml.vision.text.FirebaseVisionText;
import com.google.firebase.codelab.mlkit.GraphicOverlay.Graphic;

/**
 * Graphic instance for rendering TextBlock position, size, and ID within an associated graphic
```

```
 * overlay view.
 */
public class TextGraphic extends Graphic {

    private static final String TAG = "TextGraphic";
    private static final int TEXT_COLOR = Color.RED;
    private static final float TEXT_SIZE = 54.0f;
    private static final float STROKE_WIDTH = 4.0f;

    private final Paint rectPaint;
    private final Paint textPaint;
    private final FirebaseVisionText.Element element;

    TextGraphic(GraphicOverlay overlay, FirebaseVisionText.Element element) {
        super(overlay);

        this.element = element;

        rectPaint = new Paint();
        rectPaint.setColor(TEXT_COLOR);
        rectPaint.setStyle(Paint.Style.STROKE);
        rectPaint.setStrokeWidth(STROKE_WIDTH);

        textPaint = new Paint();
        textPaint.setColor(TEXT_COLOR);
        textPaint.setTextSize(TEXT_SIZE);
        // Redraw the overlay, as this graphic has been added.
        postInvalidate();
    }

    /**
     * Draws the text block annotations for position, size, and raw value on the supplied canvas.
     */
    @Override
    public void draw(Canvas canvas) {
        Log.d(TAG, "on draw text graphic");
        if (element == null) {
            throw new IllegalStateException("Attempting to draw a null text.");
        }

        // Draws the bounding box around the TextBlock.
        RectF rect = new RectF(element.getBoundingBox());
        canvas.drawRect(rect, rectPaint);

        // Renders the text at the bottom of the box.
        canvas.drawText(element.getText(), rect.left, rect.bottom, textPaint);
    }
}
```

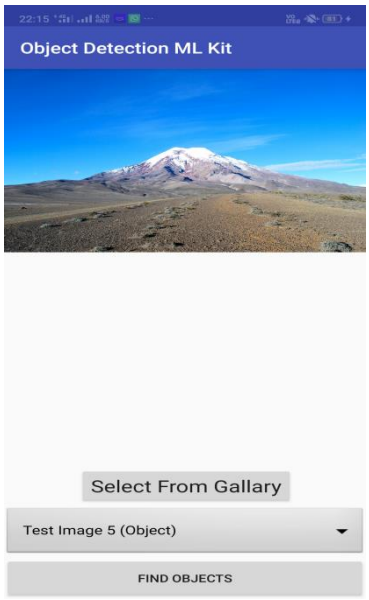
FaceContourGraphic.java

```
package com.google.firebase.codelab.mlkit;

import android.graphics.Canvas;
import android.graphics.Color;
import android.graphics.Paint;

import com.google.firebase.ml.vision.face.FirebaseVisionFace;
import com.google.firebase.ml.vision.face.FirebaseVisionFaceContour;
import com.google.firebase.ml.vision.face.FirebaseVisionFaceLandmark;

/** Graphic instance for rendering face contours graphic overlay view. */
public class FaceContourGraphic extends GraphicOverlay.Graphic {

  private static final float FACE_POSITION_RADIUS = 10.0f;
  private static final float ID_TEXT_SIZE = 70.0f;
  private static final float ID_Y_OFFSET = 80.0f;
  private static final float ID_X_OFFSET = -70.0f;
  private static final float BOX_STROKE_WIDTH = 5.0f;

  private static final int[] COLOR_CHOICES = {
    Color.BLUE, Color.CYAN, Color.GREEN, Color.MAGENTA, Color.RED, Color.WHITE, Color.YELLOW
  };
  private static int currentColorIndex = 0;

  private final Paint facePositionPaint;
  private final Paint idPaint;
  private final Paint boxPaint;

  private volatile FirebaseVisionFace firebaseVisionFace;

  public FaceContourGraphic(GraphicOverlay overlay) {
    super(overlay);

    currentColorIndex = (currentColorIndex + 1) % COLOR_CHOICES.length;
    final int selectedColor = COLOR_CHOICES[currentColorIndex];

    facePositionPaint = new Paint();
    facePositionPaint.setColor(selectedColor);

    idPaint = new Paint();
    idPaint.setColor(selectedColor);
    idPaint.setTextSize(ID_TEXT_SIZE);

    boxPaint = new Paint();
    boxPaint.setColor(selectedColor);
    boxPaint.setStyle(Paint.Style.STROKE);
    boxPaint.setStrokeWidth(BOX_STROKE_WIDTH);
  }

  public void updateFace(FirebaseVisionFace face) {
    firebaseVisionFace = face;
```

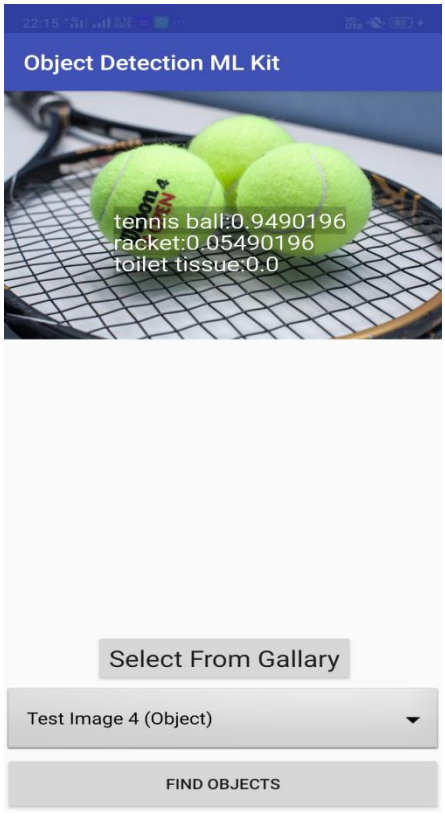
```
  postInvalidate();
 }
  @Override
 public void draw(Canvas canvas) {
  FirebaseVisionFace face = firebaseVisionFace;
  if (face == null) {
   return;
  }

  // Draws a circle at the position of the detected face, with the face's track id below.
  float x = translateX(face.getBoundingBox().centerX());
  float y = translateY(face.getBoundingBox().centerY());
  canvas.drawCircle(x, y, FACE_POSITION_RADIUS, facePositionPaint);
  canvas.drawText("id: " + face.getTrackingId(), x + ID_X_OFFSET, y + ID_Y_OFFSET, idPaint);

  // Draws a bounding box around the face.
  float xOffset = scaleX(face.getBoundingBox().width() / 2.0f);
  float yOffset = scaleY(face.getBoundingBox().height() / 2.0f);
  float left = x - xOffset;
  float top = y - yOffset;
  float right = x + xOffset;
  float bottom = y + yOffset;
  canvas.drawRect(left, top, right, bottom, boxPaint);

  FirebaseVisionFaceContour contour = face.getContour(FirebaseVisionFaceContour.ALL_POINTS);
  for (com.google.firebase.ml.vision.common.FirebaseVisionPoint point : contour.getPoints()) {
   float px = translateX(point.getX());
   float py = translateY(point.getY());
   canvas.drawCircle(px, py, FACE_POSITION_RADIUS, facePositionPaint);
  }

  if (face.getSmilingProbability() >= 0) {
   canvas.drawText(
       "happiness: " + String.format("%.2f", face.getSmilingProbability()),
       x + ID_X_OFFSET * 3,
       y - ID_Y_OFFSET,
       idPaint);
  }

  if (face.getRightEyeOpenProbability() >= 0) {
   canvas.drawText(
       "right eye: " + String.format("%.2f", face.getRightEyeOpenProbability()),
       x - ID_X_OFFSET,
       y,
       idPaint);
  }
  if (face.getLeftEyeOpenProbability() >= 0) {
   canvas.drawText(
       "left eye: " + String.format("%.2f", face.getLeftEyeOpenProbability()),
       x + ID_X_OFFSET * 6,
       y,
       idPaint);
```

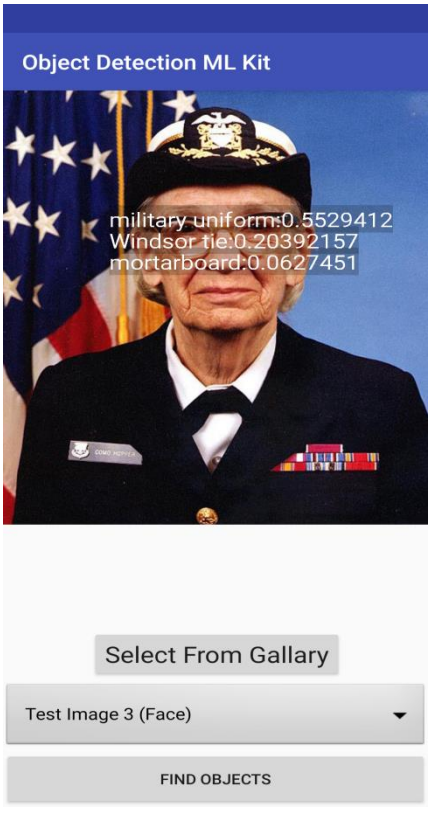
```
    }
    FirebaseVisionFaceLandmark leftEye = face.getLandmark(FirebaseVisionFaceLandmark.LEFT_EYE);
    if (leftEye != null && leftEye.getPosition() != null) {
      canvas.drawCircle(
          translateX(leftEye.getPosition().getX()),
          translateY(leftEye.getPosition().getY()),
          FACE_POSITION_RADIUS,
          facePositionPaint);
    }
    FirebaseVisionFaceLandmark rightEye = face.getLandmark(FirebaseVisionFaceLandmark.RIGHT_EYE);
    if (rightEye != null && rightEye.getPosition() != null) {
      canvas.drawCircle(
          translateX(rightEye.getPosition().getX()),
          translateY(rightEye.getPosition().getY()),
          FACE_POSITION_RADIUS,
          facePositionPaint);
    }

    FirebaseVisionFaceLandmark leftCheek = face.getLandmark(FirebaseVisionFaceLandmark.LEFT_CHEEK);
    if (leftCheek != null && leftCheek.getPosition() != null) {
      canvas.drawCircle(
          translateX(leftCheek.getPosition().getX()),
          translateY(leftCheek.getPosition().getY()),
          FACE_POSITION_RADIUS,
          facePositionPaint);
    }
    FirebaseVisionFaceLandmark rightCheek =
        face.getLandmark(FirebaseVisionFaceLandmark.RIGHT_CHEEK);
    if (rightCheek != null && rightCheek.getPosition() != null) {
      canvas.drawCircle(
          translateX(rightCheek.getPosition().getX()),
          translateY(rightCheek.getPosition().getY()),
          FACE_POSITION_RADIUS,
          facePositionPaint);
    }
  }
}
```

Output:


Object Detection ML Kit
Select From Gallary
Test Image 5 (Object)
FIND OBJECTS


Object Detection ML Kit
volcano:0.9254902
valley:0.03529412
alp:0.023529412
Select From Gallary
Test Image 5 (Object)
FIND OBJECTS


Object Detection ML Kit
tennis ball:0.9490196
racket:0.05490196
toilet tissue:0.0
Select From Gallary
Test Image 4 (Object)
FIND OBJECTS


Object Detection ML Kit
military uniform:0.5529412
Windsor tie:0.20392157
mortarboard:0.0627451
Select From Gallary
Test Image 3 (Face)
FIND OBJECTS