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.

THEORY:

onCreate(Bundle savedInstanceState) Function in Android: When an Activity first call or launched then onCreate(Bundle savedInstanceState) method is responsible to create the activity.

TextView: A user interface element that displays text to the user.

setContentView(View): An activity is a single, focused thing that the user can do. Almost all activities interact with the user, so the Activity class takes care of creating a window for you in which you can place your UI with setContentView(View). While activities are often presented to the user as full-screen windows, they can also be used in other ways: as floating windows (via a theme with R.attr.windowIsFloating set), Multi-Window mode or embedded into other windows.

CODE:

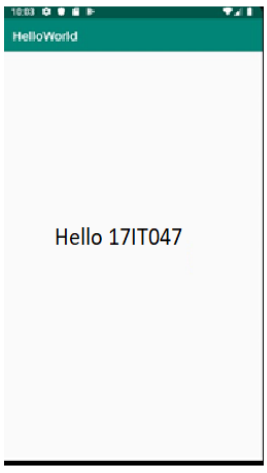
// MainActivity.java package
com.example.helloworld;
import androidx.appcompat.app.AppCompatActivity; import
android.os.Bundle;
public class MainActivity extends AppCompatActivity {
@Override
protected void onCreate(Bundle savedInstanceState) {
super.onCreate(savedInstanceState);
setContentView(R.layout.*activity_main*);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

// 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"
    android:id="@+id/constraint_layout"
    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="Hello 17IT047"
        android:textSize="20sp"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintLeft_toLeftOf="parent"
        app:layout_constraintRight_toRightOf="parent"
        app:layout_constraintTop_toTopOf="parent" />

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

OUTPUT:



LATEST APPLICATIONS:

All the application now days uses TextView and background color like whatsapp, facebook, Instagram, Google playstore, amazon etc.

LEARNING OUTCOME:

We can add different colors in the colors.xml files and use them in the application background and also text color.

PRACTICAL: 2

AIM:

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

THEORY:

About elements used:

EditText: A user interface element for entering and modifying text. When you define an edit text widget, you must specify the R.styleable.TextView_inputType attribute. For example, for plain text input set inputType to "text". Choosing the input type configures the keyboard type that is shown, acceptable characters, and appearance of the edit text.

Button: A user interface element the user can tap or click to perform an action. To specify an action when the button is pressed, set a click listener on the button object in the corresponding activity code.

About method used:

setOnClickListner(): setOnClickListener is a method in Android basically used with buttons, image buttons etc. While invoking this method a callback function will run. One can also create a class for more than one listener, so this can lead you to code reusability. After making the class you can implement android.view.View.OnClickListener{} method which gives you an override method inherited from super class called onClick(View v){} in which you can easily implement your code.

Toast.makeText(): This method takes three parameters: the application Context, the text message, and the duration for the toast. It returns a properly initialized Toast object. You can display the toast notification with show(). You may, however, want to position the toast differently or even use your own layout instead of a simple text message.

CODE:

//activity_main.xml

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

<androidx.constraintlayout.widget.ConstraintLayout
xmlns:android="http://schemas.android.com/apk/res/android"

    xmlns:app="http://schemas.android.com/apk/res-auto"

    xmlns:tools="http://schemas.android.com/tools"

    android:layout_width="match_parent"
```

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

<EditText
    android:id="@+id/Num1"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginTop="180dp"
    android:ems="10"
    android:hint="Number 1"
    android:inputType="number"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent" />

<EditText
    android:id="@+id/Num2"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginTop="72dp"
    android:ems="10"
    android:hint="Number 2"
    android:inputType="number"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
```

```
    app:layout_constraintTop_toBottomOf="@+id/Num1" />


<Button
    android:id="@+id/Sum"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginStart="64dp"
    android:layout_marginLeft="64dp"
    android:layout_marginTop="92dp"
    android:text="+"
    android:textSize="25dp"
    app:layout_constraintEnd_toStartOf="@+id/Min"
    app:layout_constraintHorizontal_bias="0.603"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/Num2" />


<Button
    android:id="@+id/Min"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginTop="92dp"
    android:layout_marginEnd="108dp"
    android:layout_marginRight="108dp"
    android:text="-"
    android:textSize="25dp"
```

```
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/Num2" />

<Button
    android:id="@+id/Mul"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginTop="36dp"
    android:text="*"
    android:textSize="25dp"
    app:layout_constraintEnd_toStartOf="@+id/Div"
    app:layout_constraintHorizontal_bias="0.755"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/Sum" />

<Button
    android:id="@+id/Div"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginTop="36dp"
    android:layout_marginEnd="108dp"
    android:layout_marginRight="108dp"
    android:text="/"
    android:textSize="25dp"
    app:layout_constraintEnd_toEndOf="parent"
```

```
        app:layout_constraintTop_toBottomOf="@+id/Min" />



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

//MainActivity.java

```
package com.example.practical2_17IT047;

import androidx.appcompat.app.AppCompatActivity;

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

public class MainActivity extends AppCompatActivity {
   EditText Num1,Num2;
   Button sum,min,mul,div;
   @Override
   protected void onCreate(Bundle savedInstanceState) {
      super.onCreate(savedInstanceState);
      setContentView(R.layout.activity_main);
         Num1 = (EditText)findViewById(R.id.Num1);
         Num2 = (EditText)findViewById(R.id.Num2);
         sum = (Button)findViewById(R.id.Sum);
         min = (Button)findViewById(R.id.Min);
         mul = (Button)findViewById(R.id.Mul);
         div = (Button)findViewById(R.id.Div);

         sum.setOnClickListener(new View.OnClickListener(){
            public void onClick(View v){
               int num1 = Integer.parseInt(Num1.getText().toString());
               int num2 = Integer.parseInt(Num2.getText().toString());
               Toast.makeText(getApplicationContext(),"Sum: "+
Integer.toString(num1+num2),Toast.LENGTH_LONG).show();
            }
         });
         min.setOnClickListener(new View.OnClickListener(){
         public void onClick(View v){
            int num1 = Integer.parseInt(Num1.getText().toString());
            int num2 = Integer.parseInt(Num2.getText().toString());
            Toast.makeText(getApplicationContext(),"Minus: "+ Integer.toString(num1-
num2),Toast.LENGTH_LONG).show();
         }
      });
         mul.setOnClickListener(new View.OnClickListener(){
```

```
        public void onClick(View v){
            int num1 = Integer.parseInt(Num1.getText().toString());
            int num2 = Integer.parseInt(Num2.getText().toString());
            Toast.makeText(getApplicationContext(),"Mul: "+
Integer.toString(num1*num2),Toast.LENGTH_LONG).show();
        }
    });
        div.setOnClickListener(new View.OnClickListener(){
        public void onClick(View v) {
            int num1 = Integer.parseInt(Num1.getText().toString());
            int num2 = Integer.parseInt(Num2.getText().toString());
            if (num2 == 0) {
                Toast.makeText(getApplicationContext(), "Divison is not possible ",
Toast.LENGTH_LONG).show();
            } else {
                Toast.makeText(getApplicationContext(), "Div: " + Integer.toString(num1 / num2),
Toast.LENGTH_LONG).show();
            }
        }
        });}}
```

OUTPUT:

LATEST APPLICATIONS:

Buttons and Toast messages are widely used in most of the applications now days as example google playstore, Facebook, Instagram etc.

LEARNING OUTCOME:

We can Toast a Message to provide user interaction when user press button in the application, we can provide the information about what backend process is running.

PRACTICAL: 3

AIM:

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

THEORY:

Toast: In Android, Toast is used to display information for a period of time. It contains a message to be displayed quickly and disappears after specified period of time. It does not block the user interaction. Toast is a subclass of Object class. In this we use two constants for setting the duration for the Toast. Toast notification in android always appears near the bottom of the screen. We can also create our custom toast by using custom layout(xml file).

Chronometer: In Android, Chronometer is a class that implements a simple timer. Chronometer is a subclass of TextView. This class helps us to add a timer in our app.

CODE:

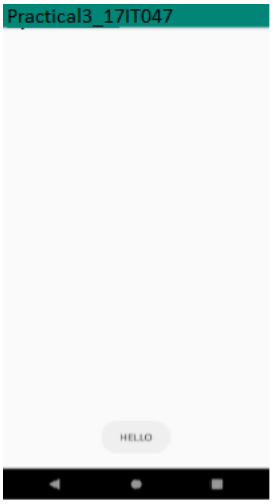
//MainActvity.java

```
package com.example.Practical3_17IT047;
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 {
   private Chronometer chronometer;
   private TextView textView;
   private int time;
   @Override
   protected void onCreate(Bundle savedInstanceState) {
      super.onCreate(savedInstanceState);
      setContentView(R.layout.activity_main);
      chronometer = (Chronometer)findViewById(R.id.chronometer);
      textView = (TextView)findViewById(R.id.mytext);
      chronometer.start();
      time = 0;
      chronometer.setOnChronometerTickListener(new Chronometer.
OnChronometerTickListener() {
         @Override
         public void onChronometerTick(Chronometer mychronometer) {
            time++;
            if (time%11==0){
               Toast.makeText(getApplicationContext(),"10 Second
Happened",Toast.LENGTH_SHORT).show();
            }
         }
      });
   }
```

```
}
//activity_mail.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"
   android:background="#03A9F4"
   >
   <Chronometer
      android:id="@+id/chronometer"
      android:layout_width="wrap_content"
      android:layout_height="wrap_content"
      android:padding="20dp"
      android:textAlignment="center"
      android:textColor="#555"
      android:textSize="50sp"
      app:layout_constraintBottom_toBottomOf="parent"
      app:layout_constraintEnd_toEndOf="parent"
      app:layout_constraintHorizontal_bias="0.497"
      app:layout_constraintStart_toStartOf="parent"
      app:layout_constraintTop_toTopOf="parent"
      app:layout_constraintVertical_bias="0.443" />
   <TextView
      android:layout_width="wrap_content"
      android:layout_height="wrap_content"
      android:text="@string/id"
      android:textSize="50sp"
      app:layout_constraintBottom_toBottomOf="parent"
      app:layout_constraintEnd_toEndOf="parent"
      app:layout_constraintHorizontal_bias="0.497"
      app:layout_constraintStart_toStartOf="parent"
      app:layout_constraintTop_toTopOf="parent"
      app:layout_constraintVertical_bias="0.179"
      android:textColor="#000"
      />
   <TextView
      android:id="@+id/mytext"
      android:layout_width="wrap_content"
      android:layout_height="wrap_content"
      android:text="@string/message"
      android:textColor="#000"
      android:textSize="20sp"
      app:layout_constraintBottom_toBottomOf="parent"
      app:layout_constraintEnd_toEndOf="parent"
      app:layout_constraintHorizontal_bias="0.497"
      app:layout_constraintStart_toStartOf="parent"
```

```
        app:layout_constraintTop_toTopOf="parent"
        app:layout_constraintVertical_bias="0.671" />
</androidx.constraintlayout.widget.ConstraintLayout>
```

OUTPUT:



LATEST APPLICATIONS:

Most of the applications uses this functionality for the user interaction like Goggle playstore, Clock, Gmail.

LEARNING OUTCOME:

Now days every application notifies user using toast message that what process is running in the background. So toast messages are useful for user interaction.

PRACTICAL: 4

AIM:

Create a temperature converter Application. (Fahrenheit-Celsius)

THEORY:

EditText: A user interface element for entering and modifying text. When you define an edit text widget, you must specify the R.styleable.TextView_inputType attribute. For example, for plain text input set inputType to "text". Choosing the input type configures the keyboard type that is shown, acceptable characters, and appearance of the edit text.

Button: A user interface element the user can tap or click to perform an action. To specify an action when the button is pressed, set a click listener on the button object in the corresponding activity code.

About method used:

setOnClickListner(): setOnClickListener is a method in Android basically used with buttons, image buttons etc. While invoking this method a callback function will run. One can also create a class for more than one listener, so this can lead you to code reusability. After making the class you can implement android.view.View.OnClickListener{} method which gives you an override method inherited from super class called onClick(View v){} in which you can easily implement your code.

CODE:

// MainActivity.java

```
package com.example.practical4_17IT047; 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 {
    EditText Value;
    Button cfk;
    Button fck;
    Button kcf;
    TextView text1;
    TextView text2;
```

```
   @Override
   protected void onCreate(Bundle savedInstanceState) {
super.onCreate(savedInstanceState);
setContentView(R.layout.activity_main);        cfk =
(Button)findViewById(R.id.cfk);        fck =
(Button)findViewById(R.id.fck);        kcf =
(Button)findViewById(R.id.kcf);
      Value = (EditText)findViewById(R.id.Value);
text1 = (TextView)findViewById(R.id.Text1);        text2
= (TextView)findViewById(R.id.Text2);
      cfk.setOnClickListener(new View.OnClickListener(){
      @Override
         public  void onClick(View v){
         double K,F;
         double intValue = Integer.parseInt(Value.getText().toString());
         K = intValue + 273.15;
F = (intValue*9/5) + 32;
         text1.setText(Double.toString(K) +" K" );
text2.setText(Double.toString(F) + " F");
      }
      });
      fck.setOnClickListener(new View.OnClickListener(){
         @Override
         public  void onClick(View v){
double C,K;
            double intValue = Integer.parseInt(Value.getText().toString());
C = (intValue-32)*5/9;
            K = C+273.15;
text1.setText(Double.toString(K) +" K" );
text2.setText(Double.toString(C) + " C");
         }
      });
      kcf.setOnClickListener(new View.OnClickListener(){
         @Override
         public  void onClick(View v){               double C,F;
double intValue = Integer.parseInt(Value.getText().toString());
C = intValue - 273.15;              F = (intValue*9/5) + 32;
            text1.setText(Double.toString(C) +" C" );
text2.setText(Double.toString(F) + " F");           }
});}}
```

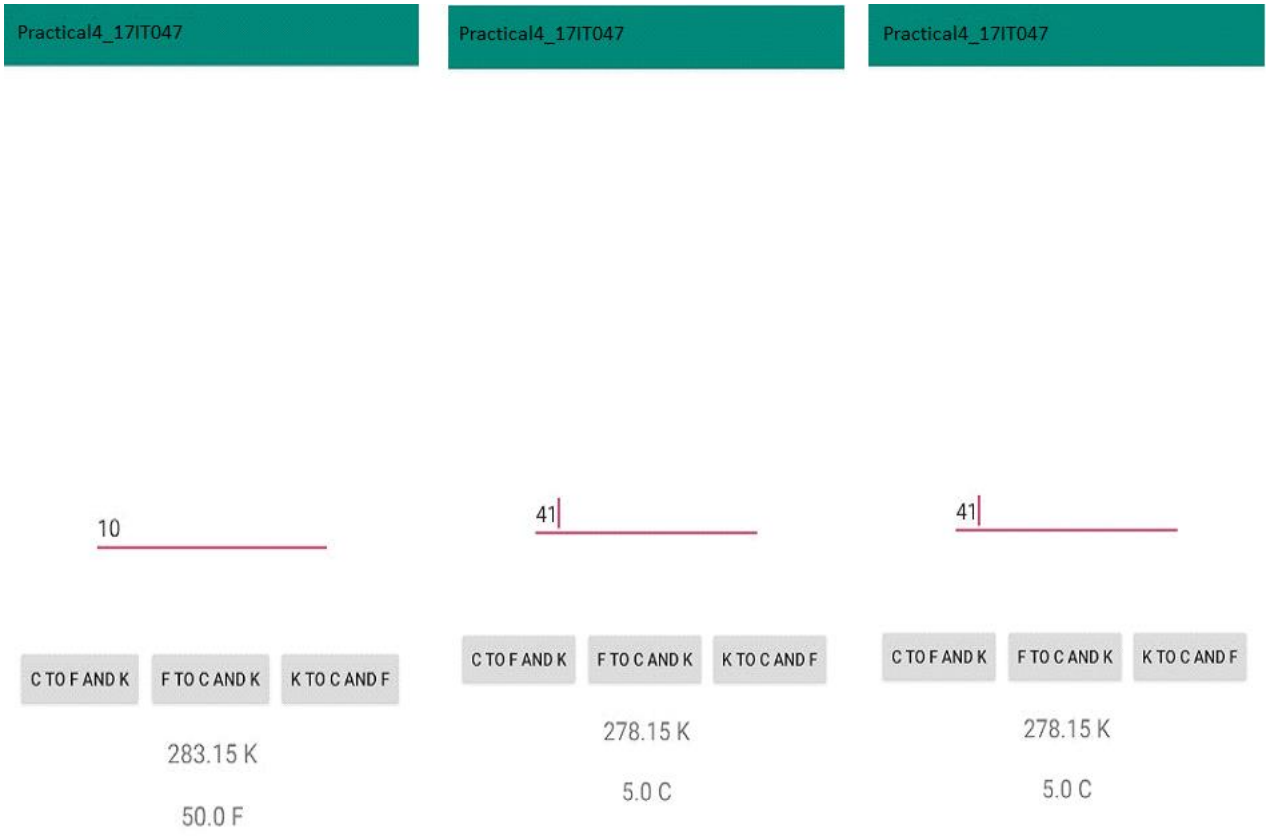
// activity_main.xml

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

```
<Button
    android:id="@+id/cfk"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginStart="16dp"
    android:layout_marginLeft="16dp"
    android:layout_marginBottom="196dp"
    android:text="C to F and K"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintStart_toStartOf="parent" />

<Button
    android:id="@+id/fck"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginBottom="196dp"
    android:text="F to C and K"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toStartOf="@+id/kcf"
    app:layout_constraintHorizontal_bias="0.585"
    app:layout_constraintStart_toEndOf="@+id/cfk" />

<Button
    android:id="@+id/kcf"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginEnd="28dp"
    android:layout_marginRight="28dp"
    android:layout_marginBottom="196dp"
    android:text="K to C and F"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent" />

<TextView
    android:id="@+id/Text1"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginBottom="20dp"
    android:textSize="20sp"
    app:layout_constraintBottom_toTopOf="@+id/Text2"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent" />

<TextView
    android:id="@+id/Text2"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginBottom="104dp"
    android:textSize="20sp"
```

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

    <EditText
        android:id="@+id/Value"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginTop="280dp"
        android:ems="10"
        android:hint="Temperature"
        android:inputType="number"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent" />

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

OUTPUT:



LATEST APPLICATIONS:

Application which has login page, that need button and EditText mostly, like facebook login page, instagram login page, Netflix etc.

LEARNING OUTCOME:

Using EditText we can take input from the user and using button we can do some process on the input given by the user as user want.

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.

THEORY:

EditText: A user interface element for entering and modifying text. When you define an edit text widget, you must specify the R.styleable.TextView_inputType attribute. For example, for plain text input set inputType to "text". Choosing the input type configures the keyboard type that is shown, acceptable characters, and appearance of the edit text.

Button: A user interface element the user can tap or click to perform an action. To specify an action when the button is pressed, set a click listener on the button object in the corresponding activity code.

About method used:

setOnClickListner(): setOnClickListener is a method in Android basically used with buttons, image buttons etc. While invoking this method a callback function will run. One can also create a class for more than one listener, so this can lead you to code reusability. After making the class you can implement android.view.View.OnClickListener{} method which gives you an override method inherited from super class called onClick(View v){} in which you can easily implement your code.

CODE:

```
// MainActivity.java

package com.example.practical5_17IT047;

import androidx.appcompat.app.AppCompatActivity;

import android.content.Intent;
import android.graphics.Color;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.LinearLayout;
import android.widget.TextView;
import android.widget.Toast;

public class MainActivity extends AppCompatActivity {

   Button Login;
   Button Cancel;
   EditText UserName;
   EditText Password;
   TextView OutPut;
   int Count =3;
   @Override
   protected void onCreate(Bundle savedInstanceState) {
      super.onCreate(savedInstanceState);
      setContentView(R.layout.activity_main);

   Login = (Button)findViewById(R.id.Login);
   Cancel = (Button)findViewById(R.id.Cancel);
   UserName = (EditText)findViewById(R.id.UserName);
   Password = (EditText)findViewById(R.id.Password);
   OutPut = (TextView)findViewById(R.id.OutPut);



   Login.setOnClickListener(new View.OnClickListener() {
      @Override
      public void onClick(View v) {

         if (UserName.getText().toString().equals("Admin") &&
Password.getText().toString().equals("admin")) {
            OutPut.setText("Login Successful");
            OutPut.setBackgroundColor(Color.GREEN);
         } else {
            if (Count > 0) {
```

```
                OutPut.setText("Login Failed");
                OutPut.setBackgroundColor(Color.RED);
                Count--;
                Toast.makeText(getApplicationContext(), Integer.toString(Count) + " Attempts
Remaining", Toast.LENGTH_LONG).show();
            } else {
                Login.setEnabled(false);
            }
          }
        }
    });
    Cancel.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            finish();
            System.exit(0);
        }
    });}}
```

//activity_main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:id="@+id/Layout1"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".MainActivity">

    <EditText
        android:id="@+id/UserName"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginTop="224dp"
        android:ems="10"
        android:hint="User Name"
        android:inputType="textPersonName"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.515"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent" />

    <EditText
        android:id="@+id/Password"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginTop="44dp"
```

```
        android:ems="10"
        android:hint="Password"
        android:inputType="textPassword"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/UserName" />

    <Button
        android:id="@+id/Login"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginBottom="264dp"
        android:text="Login"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toStartOf="@+id/Cancel"
        app:layout_constraintHorizontal_bias="0.712"
        app:layout_constraintStart_toStartOf="parent" />

    <Button
        android:id="@+id/Cancel"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginEnd="96dp"
        android:layout_marginRight="96dp"
        android:layout_marginBottom="264dp"
        android:text="Cancle"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent" />

    <TextView
        android:id="@+id/OutPut"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginBottom="140dp"
        android:textSize="20sp"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent" />
</androidx.constraintlayout.widget.ConstraintLayout>
```

OUTPUT:

LATEST APPLICATIONS:

Every application which contains data of the user must have the login page In the application. As example Gmail login, facebook login, Instagram login, CHARUSAT login etc.

LEARNING OUTCOME:

Login page is necessary in most of the application now days for security reasons. So by using login page in our application we can make our application more secure and user friendly.

PRACTICAL: 6

AIM:

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

THEORY:

Camera Manager: This is used to get all the cameras available in the device like front camera back camera each having the camera id.

CameraDevice: You can get it from Camera Manager class by its id.

CaptureRequest: You can create a capture request from camera device to capture images.

CameraCaptureSession: To get capture request's from Camera Device create a CameraCaptureSession.

CameraCaptureSession.CaptureCallback: This is going to provide the Capture session results.

CODE:

// **MainActivity.java**

```
package com.example.practical6_17IT047;

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

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.os.Build;
import android.os.Bundle;
import android.widget.CompoundButton;
import android.widget.ToggleButton;

@RequiresApi(api = Build.VERSION_CODES.LOLLIPOP)
public class MainActivity extends AppCompatActivity {

   private CameraManager mCameraManager;
   private String mCameraId;

   private ToggleButton toggleButton;

   @RequiresApi(api = Build.VERSION_CODES.LOLLIPOP)
   @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();
```

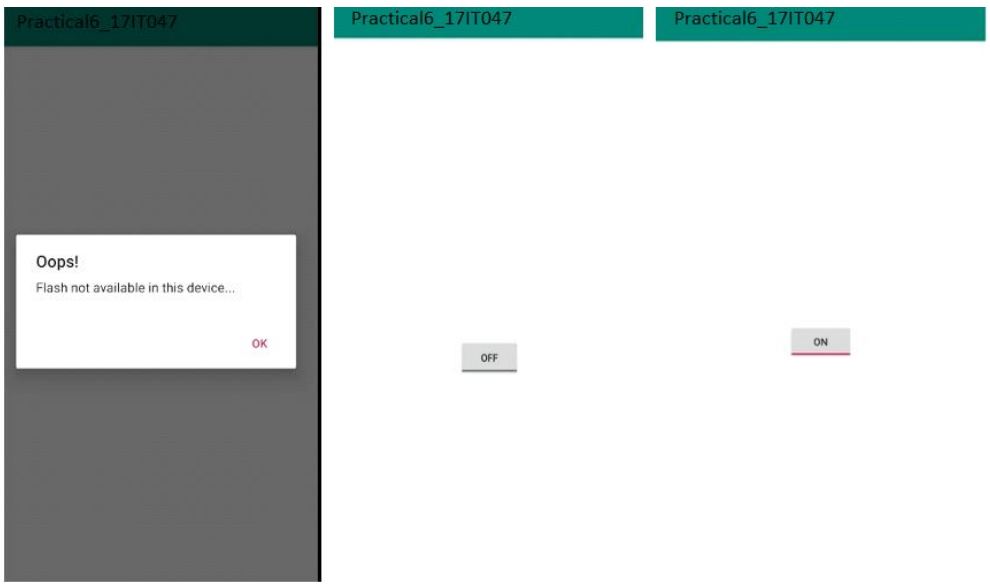
```
        }

        toggleButton = findViewById(R.id.toggleButton);

        toggleButton.setOnCheckedChangeListener(new
CompoundButton.OnCheckedChangeListener() {
            @RequiresApi(api = Build.VERSION_CODES.M)
            @Override
            public void onCheckedChanged(CompoundButton buttonView, boolean
isChecked) {
                switchFlashLight(isChecked);
            }
        });
    }

    public void showNoFlashError() {
        AlertDialog alert = new AlertDialog.Builder(this)
                .create();
        alert.setTitle("Oops!");
        alert.setMessage("Flash not available in this device...");
        alert.setButton(DialogInterface.BUTTON_POSITIVE, "OK", new
DialogInterface.OnClickListener() {
            public void onClick(DialogInterface dialog, int which) {
                finish();
            }
        });
        alert.show();
    }

    @RequiresApi(api = Build.VERSION_CODES.M)
    public void switchFlashLight(boolean status) {
        try {
            mCameraManager.setTorchMode(mCameraId, status);
        } catch (CameraAccessException e) {
            e.printStackTrace();
        }
    }
}
```

//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">
    <ToggleButton
        android:id="@+id/toggleButton"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="ToggleButton"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent" />
</androidx.constraintlayout.widget.ConstraintLayout>
```

OUTPUT:



LATEST APPLICATIONS:

Torch/FlashLight can be used to notify the person for the message or the call. TrueCaller is using this to send flash messages.

LEARNING OUTCOME:

1. Working with android manifest file.
2. Working with device local resources.

PRACTICAL: 7

AIM:

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

THEORY:

Android Menu: - Defines a Menu, which is a container for menu items. A <menu> element must be the root node for the file and can hold one or more <item> and <group> elements.

Menu items: - Creates a MenuItem, which represents a single item in a menu. This element may contain a nested <menu> element in order to create a submenu.

Menu Group: - An optional, invisible container for <item> elements. It allows you to categorize menu items so they share properties such as active state and visibility.

CODE:

// MainActivity.java

```
package com.example.practical7_17it047;

import androidx.annotation.NonNull;
import androidx.appcompat.app.AppCompatActivity;
import androidx.constraintlayout.widget.ConstraintLayout;

import android.graphics.Color;
import android.os.Bundle;
import android.view.Menu;
import android.view.MenuInflater;
import android.view.MenuItem;
import android.view.View;

public class MainActivity extends AppCompatActivity {
   private ConstraintLayout constraintLayout;
   @Override
   protected void onCreate(Bundle savedInstanceState) {
      super.onCreate(savedInstanceState);
      setContentView(R.layout.activity_main);
      constraintLayout = findViewById(R.id.constraint_layout);
   }
   @Override
   public boolean onCreateOptionsMenu(Menu menu) {
      MenuInflater inflater = getMenuInflater();
      inflater.inflate(R.menu.color_menu,menu);
      return true;
   }
```

```
        @Override
        public boolean onOptionsItemSelected(@NonNull MenuItem item) {

            switch (item.getItemId()){
                case R.id.RED:
                    this.constraintLayout.setBackgroundColor(Color.RED);
                    break;
                case R.id.GREEN:
                    this.constraintLayout.setBackgroundColor(Color.GREEN);
                    break;
                case R.id.BLUE:
                    this.constraintLayout.setBackgroundColor(Color.BLUE);
                    break;
                default:
                    return super.onOptionsItemSelected(item);
            }
            return  super.onOptionsItemSelected(item);
        }
    }
```

//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"
    android:id="@+id/constraint_layout"
    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="Hello 17IT002"
        android:textSize="20sp"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintLeft_toLeftOf="parent"
        app:layout_constraintRight_toRightOf="parent"
        app:layout_constraintTop_toTopOf="parent" />

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

OUTPUT:

LATEST APPLICATIONS:

Most of the application use menu to navigate to different tabs, like Playstore, gmail, facebook etc.

LEARNING OUTCOME:

1. Working with android different kind of layouts and resource files.
2. Working with android menu layout.

PRACTICAL: 8

AIM:

Create an application with the help of fragment.

THEORY:

Fragment: - In Android, Fragment is a part of an activity which enable more modular activity design. It will not be wrong if we say a fragment is a kind of sub-activity. It represents a behavior or a portion of user interface in an Activity. We can combine multiple Fragments in Single Activity to build a multi panel UI and reuse a Fragment in multiple Activities. We always need to embed Fragment in an activity and the fragment lifecycle is directly affected by the host activity's lifecycle.

Need of Fragments in Android: - Before the introduction of Fragment's we can only show a single Activity on the screen at one given point of time so we were not able to divide the screen and control different parts separately. With the help of Fragment's, we can divide the screens in different parts and controls different parts separately.

By using Fragments, we can comprise multiple Fragments in a single Activity. Fragments have their own events, layouts and complete life cycle. It provides flexibility and also removed the limitation of single Activity on the screen at a time.

CODE:

// MainActivity.java

```
package com.example.pr8;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.Toast;
import androidx.appcompat.app.AppCompatActivity;
public class MainActivity extends AppCompatActivity {
   private Button button;
   @Override
   protected void onCreate(Bundle savedInstanceState) {
      super.onCreate(savedInstanceState);
      setContentView(R.layout.activity_main);
      button = findViewById(R.id.button);
      button.setOnClickListener(new View.OnClickListener() {
         @Override
         public void onClick(View view) {
            Toast.makeText(getApplicationContext(),"Main Activity
Toast",Toast.LENGTH_SHORT).show();
         }
```

```
            });
        }
    }
```

// **MyFragment.java**

```
    package com.example.pr8;
    import android.os.Bundle;
    import android.view.LayoutInflater;
    import android.view.View;
    import android.view.ViewGroup;
    import android.widget.Button;
    import android.widget.Toast;

    import androidx.annotation.NonNull;
    import androidx.annotation.Nullable;
    import androidx.fragment.app.Fragment;
    public class MyFragment extends Fragment {
        private View view;
        private Button button;
        @Override
        public void onCreate(@Nullable Bundle savedInstanceState) {
            super.onCreate(savedInstanceState);

        }
        @Nullable
        @Override
        public View onCreateView(@NonNull LayoutInflater inflater, @Nullable ViewGroup
    container, @Nullable Bundle savedInstanceState) {
            view = inflater.inflate(R.layout.fragment_my,container,false);
            button = view.findViewById(R.id.button);
            button.setOnClickListener(new View.OnClickListener() {
                @Override
                public void onClick(View view) {
                    Toast.makeText(getActivity(),"Fragment
    Activity",Toast.LENGTH_SHORT).show();
                }
            });
            return view;
        }
    }
```

// **activity_mail.xml**

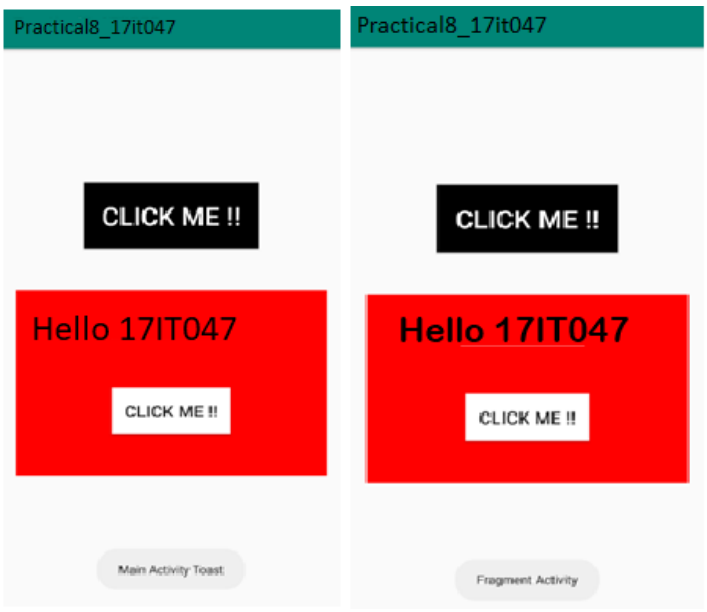
```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
   xmlns:app="http://schemas.android.com/apk/res-auto"
   xmlns:tools="http://schemas.android.com/tools"
   android:layout_width="match_parent"
   android:layout_height="match_parent"
   tools:context=".MainActivity"
   android:gravity="center"
   android:orientation="vertical">
      <Button
      android:id="@+id/button"
      android:layout_width="wrap_content"
      android:layout_height="wrap_content"
      android:text="Click me !!"
      android:textSize="30sp"
      android:background="#000"
      android:textColor="#FFF"
      android:padding="20dp"/>
   <fragment
      android:id="@+id/fragment"
      android:name="com.example.pr8.MyFragment"
      android:layout_width="match_parent"
      android:layout_height="wrap_content"
      android:layout_marginTop="50dp"
      android:layout_marginHorizontal="20dp"/>
</LinearLayout>
```

// fragment_my.xml

```
<?xml version="1.0" encoding="utf-8"?>
<FrameLayout 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"
   tools:context="com.example.pr8.MyFragment">
   <LinearLayout
      android:layout_width="match_parent"
      android:layout_height="match_parent"
      android:orientation="vertical"
      android:gravity="center"
      android:background="#FF0000"       >
      <TextView
         android:layout_width="match_parent"
         android:layout_height="wrap_content"
```

```
            android:text="17IT047"
            android:textAlignment="center"
            android:textSize="50sp"
            android:textColor="#000"/>
        <Button
            android:id="@+id/button"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="Click Me !!"
            android:textSize="20sp"
            android:padding="15dp"
            android:background="#FFF"
            android:layout_margin="50dp" />
    </LinearLayout>
</FrameLayout>
```

OUTPUT:



LATEST APPLICATIONS:

Application like Google play store, Movies, Music and many more which shows content in tab layout uses fragment for display in different content for different tabs.

LEARNING OUTCOME:

1. Working with android fragment.
2. How android fragment activity is different from MainActivity.
3. Use of fragment in more than 1 activity.

PRACTICAL: 9

AIM:

Create an application with the help of web view.

THEORY:

WebView: - WebView is a view used to display the web pages in application. This class is the basis upon which you can roll your own web browser or simply use it to display some online content within your Activity. We can also specify HTML string and can show it inside our application using a WebView. Basically, WebView turns application into a web application.

CODE:

// MainActivity.java

```
package com.example.pr9;
import android.os.Bundle;
import android.webkit.WebView;
import androidx.appcompat.app.AppCompatActivity;
public class MainActivity extends AppCompatActivity {
   private WebView webView;
   @Override
   protected void onCreate(Bundle savedInstanceState) {
      super.onCreate(savedInstanceState);
      setContentView(R.layout.activity_main);
      webView = findViewById(R.id.mldag);
      webView.loadUrl("https://sites.google.com/view/miracle-of-science/home");
   }
}
```

// activity_mail.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"
   tools:context=".MainActivity"
   android:gravity="center"
   android:orientation="vertical">

   <WebView
      android:id="@+id/mldag"
      android:layout_width="match_parent"
      android:layout_height="match_parent" />
```

```
        </LinearLayout>

// AndroidManifest.xml

        <?xml version="1.0" encoding="utf-8"?>
        <manifest xmlns:android="http://schemas.android.com/apk/res/android"
            package="com.example.pr9">
            <uses-permission android:name="android.permission.INTERNET" />
            <application
                android:allowBackup="true"
                android:icon="@mipmap/ic_launcher"
                android:label="@string/app_name"
                android:roundIcon="@mipmap/ic_launcher_round"
                android:supportsRtl="true"
                android:theme="@style/AppTheme">
                <activity android:name="com.example.pr9.MainActivity">
                    <intent-filter>
                        <action android:name="android.intent.action.MAIN" />
                        <category android:name="android.intent.category.LAUNCHER" />
                    </intent-filter>
                </activity>
            </application>
</manifest>
```

OUTPUT:

LATEST APPLICATIONS:

Application like WebView, Instagram, Gmail, and many more which shows some of the app content in webview.

LEARNING OUTCOME:

1. Handling internet permission for WebView in android manifest.
2. Working with WebView.

PRACTICAL: 10

AIM:

Create an application with the help of database.

THEORY:

Room: - Room is a persistence library, part of the Android Architecture Components. It makes it easier to work with SQLiteDatabase objects in your app, decreasing the amount of boilerplate code and verifying SQL queries at compile time.

SQLite: - SQLite is a Structure query base database, open source, light weight, no network access and standalone database. It support embedded relational database features.

onCreate(SQLiteDatabase sqLiteDatabase) method is called only once throughout the application lifecycle. It will be called whenever there is a first call to getReadableDatabase() or getWritableDatabase() function available in super SQLiteOpenHelper class. So SQLiteOpenHelper class call the onCreate() method after creating database and instantiate SQLiteDatabase object. Database name is passed in constructor call.

onUpgrade(SQLiteDatabase db,int oldVersion, int newVersion) is only called whenever there is a updation in existing version. So to update a version we have to increment the value of version variable passed in the superclass constructor.

CODE:

```
//MainActivity.java
package com.example.practical10_17it047;

import androidx.appcompat.app.AppCompatActivity;

import android.app.Activity;
import android.app.AlertDialog;
import android.content.Context;
import android.database.Cursor;
import android.database.sqlite.SQLiteDatabase;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;

public class MainActivity extends Activity implements
android.view.View.OnClickListener {

   EditText Rollno,Name,Marks;
   Button Insert,Delete,Update,View,ViewAll;
   SQLiteDatabase db;
   /** Called when the activity is first created. */
   @Override
   public void onCreate(Bundle savedInstanceState)
   {
      super.onCreate(savedInstanceState);
      setContentView(R.layout.activity_main);

      Rollno=(EditText)findViewById(R.id.Rollno);
      Name=(EditText)findViewById(R.id.Name);
      Marks=(EditText)findViewById(R.id.Marks);
      Insert=(Button)findViewById(R.id.Insert);
      Delete=(Button)findViewById(R.id.Delete);
      Update=(Button)findViewById(R.id.Update);
      View=(Button)findViewById(R.id.View);
      ViewAll=(Button)findViewById(R.id.ViewAll);

      Insert.setOnClickListener(this);
      Delete.setOnClickListener(this);
      Update.setOnClickListener(this);
      View.setOnClickListener(this);
      ViewAll.setOnClickListener(this);

      // Creating database and table
      db=openOrCreateDatabase("StudentDB",
Context.MODE_PRIVATE, null);
      db.execSQL("CREATE TABLE IF NOT EXISTS student(rollno
VARCHAR,name VARCHAR,marks VARCHAR);");
   }
```

```
   public void onClick(android.view.View view)
   {
      // Inserting a record to the Student table
      if(view==Insert)
      {
         // Checking for empty fields
         if(Rollno.getText().toString().trim().length()==0||
               Name.getText().toString().trim().length()==0||
               Marks.getText().toString().trim().length()==0)
         {
            showMessage("Error", "Please enter all values");
            return;
         }
         db.execSQL("INSERT INTO student
VALUES('"+Rollno.getText()+"','"+Name.getText()+
               "','"+Marks.getText()+"');");
         showMessage("Success", "Record added");
         clearText();
      }
      // Deleting a record from the Student table
      if(view==Delete)
      {
         // Checking for empty roll number
         if(Rollno.getText().toString().trim().length()==0)
         {
            showMessage("Error", "Please enter Rollno");
            return;
         }
         Cursor c=db.rawQuery("SELECT * FROM student WHERE
rollno='"+Rollno.getText()+"'", null);
         if(c.moveToFirst())
         {
            db.execSQL("DELETE FROM student WHERE
rollno='"+Rollno.getText()+"'");
            showMessage("Success", "Record Deleted");
         }
         else
         {
            showMessage("Error", "Invalid Rollno");
         }
         clearText();
      }
      // Updating a record in the Student table
      if(view==Update)
      {
         // Checking for empty roll number
         if(Rollno.getText().toString().trim().length()==0)
         {
            showMessage("Error", "Please enter Rollno");
            return;
         }
```

```
        Cursor c=db.rawQuery("SELECT * FROM student WHERE
rollno='"+Rollno.getText()+"'", null);
        if(c.moveToFirst()) {
           db.execSQL("UPDATE student SET name='" +
Name.getText() + "',marks='" + Marks.getText() +
                  "' WHERE rollno='"+Rollno.getText()+"'");
           showMessage("Success", "Record Modified");
        }
        else {
           showMessage("Error", "Invalid Rollno");
        }
        clearText();
     }
     // Display a record from the Student table
     if(view==View)
     {
        // Checking for empty roll number
        if(Rollno.getText().toString().trim().length()==0)
        {
           showMessage("Error", "Please enter Rollno");
           return;
        }
        Cursor c=db.rawQuery("SELECT * FROM student WHERE
rollno='"+Rollno.getText()+"'", null);
        if(c.moveToFirst())
        {
           Name.setText(c.getString(1));
           Marks.setText(c.getString(2));
        }
        else
        {
           showMessage("Error", "Invalid Rollno");
           clearText();
        }
     }
     // Displaying all the records
     if(view==ViewAll)
     {
        Cursor c=db.rawQuery("SELECT * FROM student", null);
        if(c.getCount()==0)
        {
           showMessage("Error", "No records found");
           return;
        }
        StringBuffer buffer=new StringBuffer();
        while(c.moveToNext())
        {
           buffer.append("Rollno: "+c.getString(0)+"\n");
           buffer.append("Name: "+c.getString(1)+"\n");
           buffer.append("Marks: "+c.getString(2)+"\n\n");
        }
```

```
            showMessage("Student Details", buffer.toString());
        }
    }
    public void showMessage(String title,String message)
    {
        AlertDialog.Builder builder=new AlertDialog.Builder(this);
        builder.setCancelable(true);
        builder.setTitle(title);
        builder.setMessage(message);
        builder.show();
    }
    public void clearText()
    {
        Rollno.setText("");
        Name.setText("");
        Marks.setText("");
        Rollno.requestFocus();
    }
}
```

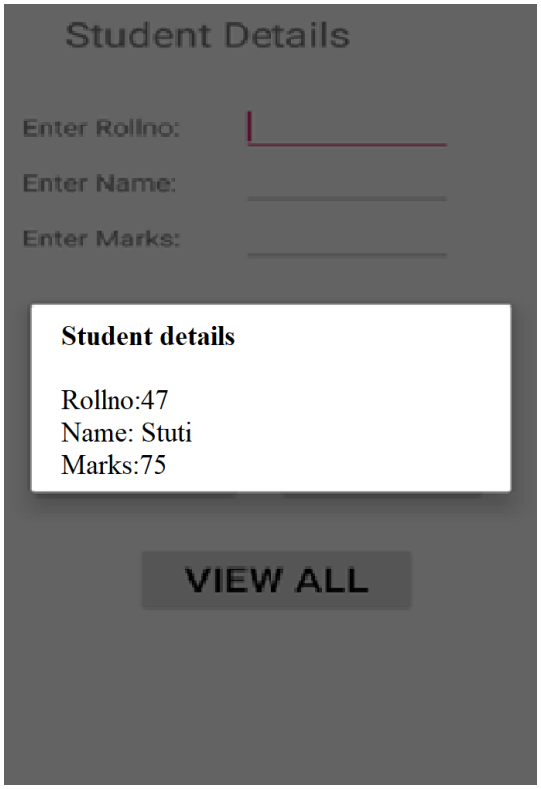
//activity_main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<AbsoluteLayout
xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent">
    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_x="50dp"
        android:layout_y="20dp"
        android:text="Student Details"
        android:textSize="30sp" />

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_x="20dp"
        android:layout_y="110dp"
        android:text="Enter Rollno:"
        android:textSize="20sp" />

    <EditText
        android:id="@+id/Rollno"
        android:layout_width="150dp"
        android:layout_height="wrap_content"
        android:layout_x="175dp"
        android:layout_y="100dp"
        android:inputType="number"
        android:textSize="20sp" />
```

```
<TextView
   android:layout_width="wrap_content"
   android:layout_height="wrap_content"
   android:layout_x="20dp"
   android:layout_y="160dp"
   android:text="Enter Name:"
   android:textSize="20sp" />

<EditText
   android:id="@+id/Name"
   android:layout_width="150dp"
   android:layout_height="wrap_content"
   android:layout_x="175dp"
   android:layout_y="150dp"
   android:inputType="text"
   android:textSize="20sp" />

<TextView
   android:layout_width="wrap_content"
   android:layout_height="wrap_content"
   android:layout_x="20dp"
   android:layout_y="210dp"
   android:text="Enter Marks:"
   android:textSize="20sp" />

<EditText
   android:id="@+id/Marks"
   android:layout_width="150dp"
   android:layout_height="wrap_content"
   android:layout_x="175dp"
   android:layout_y="200dp"
   android:inputType="number"
   android:textSize="20sp" />

<Button
   android:id="@+id/Insert"
   android:layout_width="150dp"
   android:layout_height="wrap_content"
   android:layout_x="25dp"
   android:layout_y="300dp"
   android:text="Insert"
   android:textSize="30dp" />

<Button
   android:id="@+id/Delete"
   android:layout_width="150dp"
   android:layout_height="wrap_content"
   android:layout_x="200dp"
   android:layout_y="300dp"
   android:text="Delete"
   android:textSize="30dp" />
```

```
    <Button
        android:id="@+id/Update"
        android:layout_width="150dp"
        android:layout_height="wrap_content"
        android:layout_x="25dp"
        android:layout_y="400dp"
        android:text="Update"
        android:textSize="30dp" />

    <Button
        android:id="@+id/View"
        android:layout_width="150dp"
        android:layout_height="wrap_content"
        android:layout_x="200dp"
        android:layout_y="400dp"
        android:text="View"
        android:textSize="30dp" />

    <Button
        android:id="@+id/ViewAll"
        android:layout_width="200dp"
        android:layout_height="wrap_content"
        android:layout_x="100dp"
        android:layout_y="500dp"
        android:text="View All"
        android:textSize="30dp" />

</AbsoluteLayout>
```

OUTPUT:



LATEST APPLICATIONS:

Gaming application like Subway surfer, poker and temple run, etc. uses database for storing local data.

LEARNING OUTCOME:

1. Introduction to databases in android.
2. How to use SQLite in android studio.
3. Use of Room library in android studio.

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 hint.

THEORY:

Single sign-on: Single sign-on is a property of access control of multiple related, yet independent, software systems. With this property, a user logs in with a single ID and password to gain access to any of several related systems.

GoogleSignInAccount: Class that holds the basic account information of the signed in Google user.

GoogleSignInClient: A client for interacting with the Google Sign In API.

CODE:

// MainActivity.java

```
package com.example.pr11;

import androidx.annotation.Nullable;

import androidx.appcompat.app.AppCompatActivity;

import android.content.Intent;

import android.os.Bundle;

import android.util.Log;

import android.view.View;

import android.widget.Toast;

import com.google.android.gms.auth.api.signin.GoogleSignIn;

import com.google.android.gms.auth.api.signin.GoogleSignInAccount;

import com.google.android.gms.auth.api.signin.GoogleSignInClient;

import com.google.android.gms.auth.api.signin.GoogleSignInOptions;

import com.google.android.gms.common.SignInButton;

import com.google.android.gms.common.api.ApiException;

import com.google.android.gms.tasks.Task;

public class MainActivity extends AppCompatActivity {
```

```
    private GoogleSignInClient mGoogleSignInClient;
    private GoogleSignInOptions gso;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        gso = new
GoogleSignInOptions.Builder(GoogleSignInOptions.DEFAULT_SIGN_IN)
                .requestEmail()
                .build();
        mGoogleSignInClient = GoogleSignIn.getClient(this, gso);
        SignInButton signInButton = findViewById(R.id.sign_in_button);
        signInButton.setSize(SignInButton.SIZE_STANDARD);
        signInButton.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                switch (v.getId()) {
                    case R.id.sign_in_button:
                        Intent signInIntent = mGoogleSignInClient.getSignInIntent();
                        startActivityForResult(signInIntent, 9411);
                        break;
                }
            }
        });
    }
    @Override
    protected void onStart() {
        super.onStart();
```

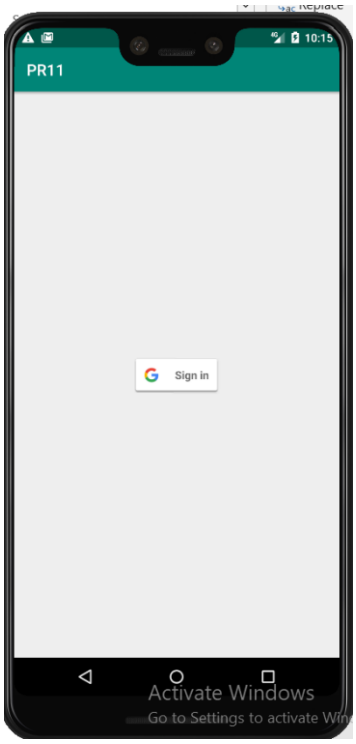
```
        GoogleSignInAccount account = GoogleSignIn.getLastSignedInAccount(this);
        if ( account == null) return;
        updateUI(account);
    }
    private void handleSignInResult(Task<GoogleSignInAccount> completedTask) {
        try {
            GoogleSignInAccount account = completedTask.getResult(ApiException.class);
            updateUI(account);
        } catch (ApiException e) {
            Log.e("SSO","signInResult:failed code=" + e.getStatusCode());
        }
    }
    private void updateUI(GoogleSignInAccount account) {
        Toast.makeText(this,"Welcome " +
account.getDisplayName(),Toast.LENGTH_LONG).show();
    }
    @Override
    protected void onActivityResult(int requestCode, int resultCode, @Nullable Intent data)
{
        super.onActivityResult(requestCode, resultCode, data);
        if (requestCode == 9411) {
            Task<GoogleSignInAccount> task =
GoogleSignIn.getSignedInAccountFromIntent(data);
            handleSignInResult(task);
        }
    }
}
```

activity_mail.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"
    tools:context=".MainActivity">
    <com.google.android.gms.common.SignInButton
        android:id="@+id/sign_in_button"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content" />
</LinearLayout>
```

OUTPUT:



LATEST APPLICATIONS:

Most of the application uses SSO in the application so users don't have to login every time. Like Instagram, gmail, facebook.

LEARNING OUTCOME:

1. Working with android different kind of layouts and resource files.
2. Working with android menu layout.

PRACTICAL: 12

AIM:

Create an application to handle support voice interaction.

THEORY:

Intent: An Intent is a simple message object that is used to communicate between android components such as activities, content providers, broadcast receivers and services. Intents are also used to transfer data between activities.

CODE:

\\ **MainActivity.java**

```
package com.example.pr12;
import androidx.annotation.Nullable;
import
androidx.appcompat.app.AppCompatActivity; import
android.content.ActivityNotFoundException;
import android.content.Intent; import
android.os.Bundle;
import android.speech.RecognizerIntent;
import android.view.View;
import android.widget.Button;
import com.google.android.material.snackbar.Snackbar;
import java.util.ArrayList;
public class MainActivity extends AppCompatActivity {
private Button mic;
private View myview;
@Override
protected void onCreate(Bundle savedInstanceState) {
super.onCreate(savedInstanceState);
setContentView(R.layout.activity_main);
mic = findViewById(R.id.mic);
      mic.setOnClickListener(new View.OnClickListener() {
         @Override
         public void onClick(View v) {
            Intent intent = new Intent(RecognizerIntent.ACTION_RECOGNIZE_SPEECH);
myview = v;              try {
            //Start the Activity and wait for the response//
startActivityForResult(intent,100);              } catch
(ActivityNotFoundException a) { }
         }
      });
   }
   @Override
   protected void onActivityResult(int requestCode, int resultCode, @Nullable Intent data)
{
      super.onActivityResult(requestCode,  resultCode,  data);
if (requestCode == 100) {
```

```
                if (resultCode == RESULT_OK && null != data) {
        ArrayList<String> result =
        data.getStringArrayListExtra(RecognizerIntent.EXTRA_RESULTS);
                    Snackbar.make(myview, result.get(0), Snackbar.LENGTH_LONG)
                        .setAction("Action", null).show();
                }
              }
            }
        }
```

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:gravity="center"
        tools:context=".MainActivity">
           <Button
              android:id="@+id/mic"
              android:layout_width="wrap_content"
        android:layout_height="wrap_content"
              android:text="Listen"/>
        </LinearLayout>
```

OUTPUT:

LATEST APPLICATIONS:

Applications for music like spotify, wynk, or like youtube use this feature of voice search for the content.

LEARNING OUTCOME:

1. Use of Intent.
2. Handing intents in onActivityResult.
3. How intent works and how to create an intent.

PRACTICAL: 13

AIM:

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

THEORY:

PIP is a special type of multi-window mode mainly used for activities that need to be active on screen but should not take up the whole screen space like watching videos, video calls, navigation, etc. It lets the user watch a video in a small window pinned to a corner of the screen (by default bottom right) while navigating between apps or browsing content on the main screen. Android 8.0 (API level 26) and above allows activities to launch in PIP mode.

The PIP window appears in the top-most layer of the screen. You can drag the PIP window to another location using some special toggles. When you tap on the window two special controls appear:

- a full-screen toggle (in the center of the window) and
- a close button (an "X" in the upper right corner).

CODE:

// MainActivity.java

```
package com.example.pr13;
import android.app.PictureInPictureParams;
import android.graphics.Point; import
android.os.Bundle; import
android.util.Rational; import
android.view.Display;
import androidx.appcompat.app.AppCompatActivity; import
com.pierfrancescosoffritti.androidyoutubeplayer.core.player.views.YouTubePlayerView;
public class MainActivity extends AppCompatActivity {
   @Override
   protected void onCreate(Bundle savedInstanceState) {
super.onCreate(savedInstanceState);
setContentView(R.layout.activity_main);        YouTubePlayerView
youTubePlayerView = findViewById(R.id.youtube_player_view);
      getLifecycle().addObserver(youTubePlayerView);
   }
   @Override
   public void onUserLeaveHint () {
Display d = getWindowManager()
            .getDefaultDisplay();
Point p = new Point();
d.getSize(p);        int width =
p.x;        int height = p.y;
Rational ratio
```

```
                = new Rational(height, width);
        PictureInPictureParams.Builder
                pip_Builder
                = new PictureInPictureParams
                .Builder();
            pip_Builder.setAspectRatio(ratio).build();
        enterPictureInPictureMode(pip_Builder.build());
          }
        }
```

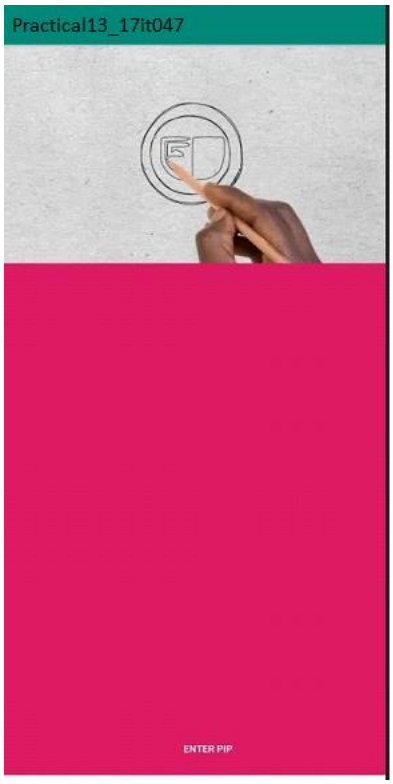
// activity_mail.xml

```
        <?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="wrap_content"
           xmlns:app="http://schemas.android.com/apk/res-auto"
        tools:context=".MainActivity">

        <com.pierfrancescosoffritti.androidyoutubeplayer.core.player.views.YouTubePlayerView
        android:id="@+id/youtube_player_view"        android:layout_width="match_parent"
        android:layout_height="wrap_content"

            app:videoId="S0Q4gqBUs7c"
            app:autoPlay="true"
            app:showFullScreenButton="false" />
        </LinearLayout>
```

//AndroidManifest.xml

```
        <?xml version="1.0" encoding="utf-8"?>
        <manifest xmlns:android="http://schemas.android.com/apk/res/android"
        package="com.example.pr13">
           <application
              android:allowBackup="true"
        android:icon="@mipmap/ic_launcher"
        android:label="@string/app_name"
              android:roundIcon="@mipmap/ic_launcher_round"
        android:supportsRtl="true"
              android:theme="@style/AppTheme.NoActionBar">
              <activity android:name=".MainActivity"
        android:supportsPictureInPicture="true"
                 android:configChanges="screenSize|smallestScreenSize|screenLayout|orientation">
                 <intent-filter>
                    <action android:name="android.intent.action.MAIN" />
                    <category android:name="android.intent.category.LAUNCHER" />
                 </intent-filter>
              </activity>
           </application>
        </manifest>
```

OUTPUT:



LATEST APPLICATIONS:

Most of the video streaming application use this feature like YouTube, Netflix, Hotstar, Twitch etc.

LEARNING OUTCOME:

1. Working with android YouTube API.
2. How to use PIP in android.
3. Managing PIP ratio.

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.

THEORY:

TensorFlow:

TensorFlow is a free and open-source software library for dataflow and differentiable programming across a range of tasks. It is a symbolic math library, and is also used for machine learning applications such as neural networks.

TensorFlow is an end-to-end open source platform for machine learning. It has a comprehensive, flexible ecosystem of tools, libraries and community resources that lets researchers push the state-of-the-art in ML and developers easily build and deploy ML powered applications.

What we learn:

- How to train a handwritten digit classifier model using TensorFlow.
- How to convert a TensorFlow model to TensorFlow Lite.
- How to deploy a TensorFlow Lite model to an Android app.

CODE:

//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(30.0f)
    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"
  }
}
```

//DigitalClassifier.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 org.tensorflow.lite.Interpreter
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
```

```
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.
    val assetManager = context.assets
    val model = loadModelFile(assetManager, "mnist.tflite")
    // Initialize TF Lite Interpreter with NNAPI enabled.
    val options = Interpreter.Options()
    options.setUseNNAPI(true)
```

```
    val interpreter = Interpreter(model, options)

    // TODO: Read the model 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

    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.

// Preprocessing: 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
```

```
  }
}
```

//AndroidManifest.xml

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
        xmlns:tools="http://schemas.android.com/tools"
        package="org.tensorflow.lite.codelabs.digitclassifier">
  <uses-sdk />
  <application
        android:allowBackup="true"
        android:icon="@mipmap/ic_launcher"
        android:label="WCMC_PR14_17IT047"
        android:roundIcon="@mipmap/ic_launcher_round"
        android:supportsRtl="true"
        android:theme="@style/AppTheme"
        tools:ignore="GoogleAppIndexingWarning">
    <activity android:name=".MainActivity"
            android:exported="true">
      <intent-filter>
        <action android:name="android.intent.action.MAIN"/>
        <category android:name="android.intent.category.LAUNCHER"/>
      </intent-filter>
    </activity>
  </application>
</manifest>
```

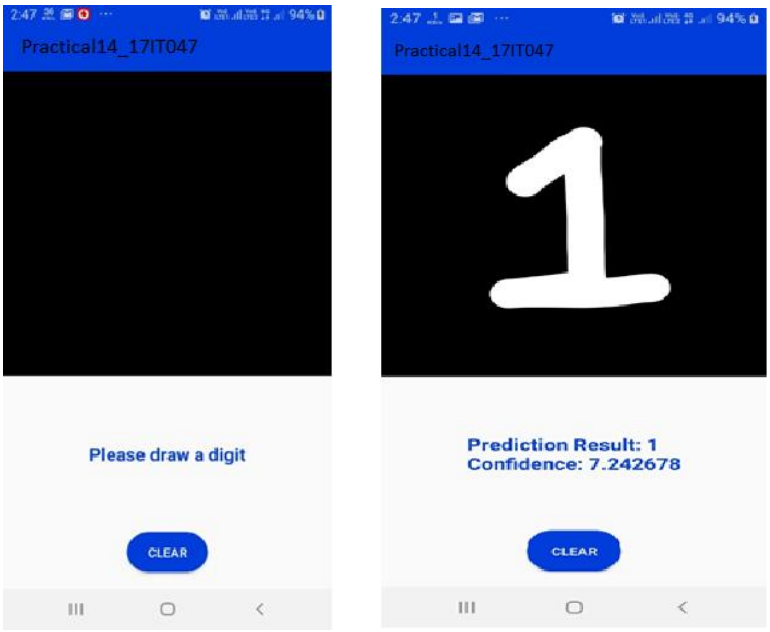
//activity_main.xml

```
<?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"
      android:textColor="@color/colorPrimary"/>
  <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"
    android:layout_marginBottom="20dp"
    android:textColor="#fff"
    android:background="@drawable/btn_shap"/>
</androidx.constraintlayout.widget.ConstraintLayout>
```

OUTPUT:



LATEST APPLICATIONS:

Amazon, flipkart, bevkoof.com, mynthra, linedin, zometo, swiggy, twitter, google search engine etc.

LEARNING OUTCOME:

In above application we can create a functionality in which we need to write the number between 1-9 and our application give you that how much accurately he try understand that whatever you write is which number. For that we develop one ML-model in TensorFlow Lite which can understand human hand writing and deploy it in one application. For that we learn that how train the ML-model in TensorFlow Lite and how to deploy it in android application.

Practical-17

CryptoCalc Link:

https://play.google.com/store/apps/details?id=com.scryptosoft.cryptocalc