TITLE-APPS TAUGHT IN THE COURSE

SUBMITTED TO: -

Mr. MAYUR DEV SEWAK (General Manager, Operations Eisystems)

Mr. CHANDAN VERMA (Head of Training, CS Domain Eisystems)

SUBMITTED BY: -

NARENDRA PAL

ABSTRACT

Simple call Dialler helps the users to call anybody without knowing the phone number if the social site provides the correct info about the user.

A **checkbox** (**check box, tick box**) is a GUI widget that permits the user to make a binary choice, i.e. a choice between one of two or more possible mutually exclusive options.

An Intent is a messaging object you can use to request an action from another app component. Although intents facilitate communication between components in several ways, there are three fundamental use cases starting an activity, starting a service, Delivering a broadcast.
Android provides many ways of storing data of an application. One of this way is called Shared Preferences. Shared Preferences allow you to save and retrieve data in the form of key, value pair.

In order to use shared preferences, you have to call a method getSharedPreferences() that returns a Shared Preference instance pointing to the file that contains the values of preferences.

WIFI Turn on/off is simple interaction with the hardware of device we installed our app.

Objective

The aim of the project is to understand the construction of app in android studio by learning to create UI in activity.xml, creating java codes and running the various project to get the desired result in an emulator.

This project will give us a simple idea of performing various steps to get a fully working app that we can install in our phone to get the desired result.

CONTENT

Figure Table

SIMPLE CALL DIALLER

The projects to this point have emphasized that Android devices are highly-connected computers and data processors. But smartphones are also phones, so in this project we show how we can use an application to implement a phone call by loading the Android phone dialler with a number.

Of course, we could do the same thing manually by opening the dialler screen and typing in the desired phone number, or invoking the number from a stored contacts list.

The primary advantage of having the computer do it is that it can deal automatically with a dynamically-updated list of phone numbers, which need not be in the contacts list for the particular phone.

The disadvantage that they require network data access. But even that issue can be dealt with by having the system cache its most recently acquired data, so that if the data network is down it is still likely that correct data will be used for the phone call.

Creating the Project in Android Studio

STEP-1: CREATING UI

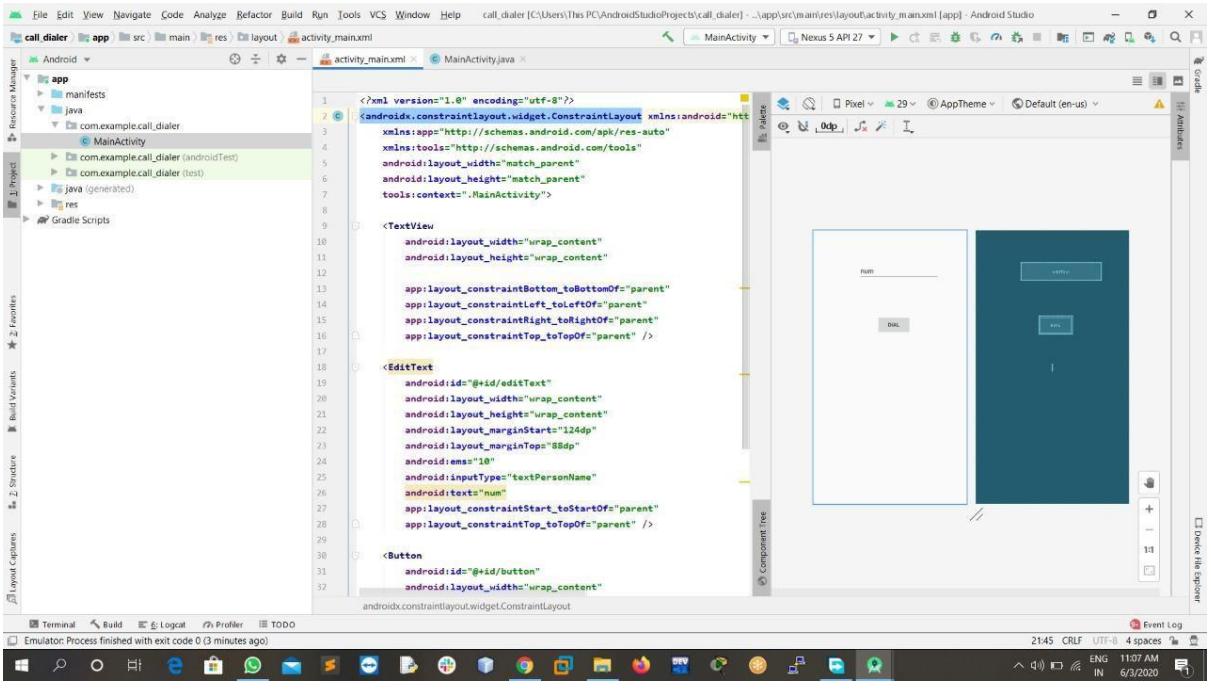

FIGURE-1.1

STEP–2: CREATING JAVA CODE

```
package com.example.calldialer;
```

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

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

public class MainActivity extends AppCompatActivity implements View.OnClickListener {
   Button dial;
   EditText num;

   @Override
   protected void onCreate(Bundle savedInstanceState) {
      super.onCreate(savedInstanceState);
      setContentView(R.layout.activity_main);
      dial=(Button)findViewById(R.id.button);
      num=(EditText)findViewById(R.id.editText);
      dial.setOnClickListener(this);
   }

   @Override
   public void onClick(View v) {
      if(v==dial)
      {
       if(num.getText().toString().trim().length()==0)
       {
          Toast.makeText(MainActivity.this,"Please Enter Mobile
Number",Toast.LENGTH_LONG).show();
       }
       if(!(num.getText().toString().length()==10))
       {
          Toast.makeText(MainActivity.this,"Number Must be 10
Digit",Toast.LENGTH_LONG).show();
       }
       else
       {
          String number="tel:"+num.getText().toString();
          Intent call=new Intent(Intent.ACTION_CALL);
          startActivity(call);
       }
      }
   }
}
```

STEP-3: ENTER THE NUMBER AND CALL

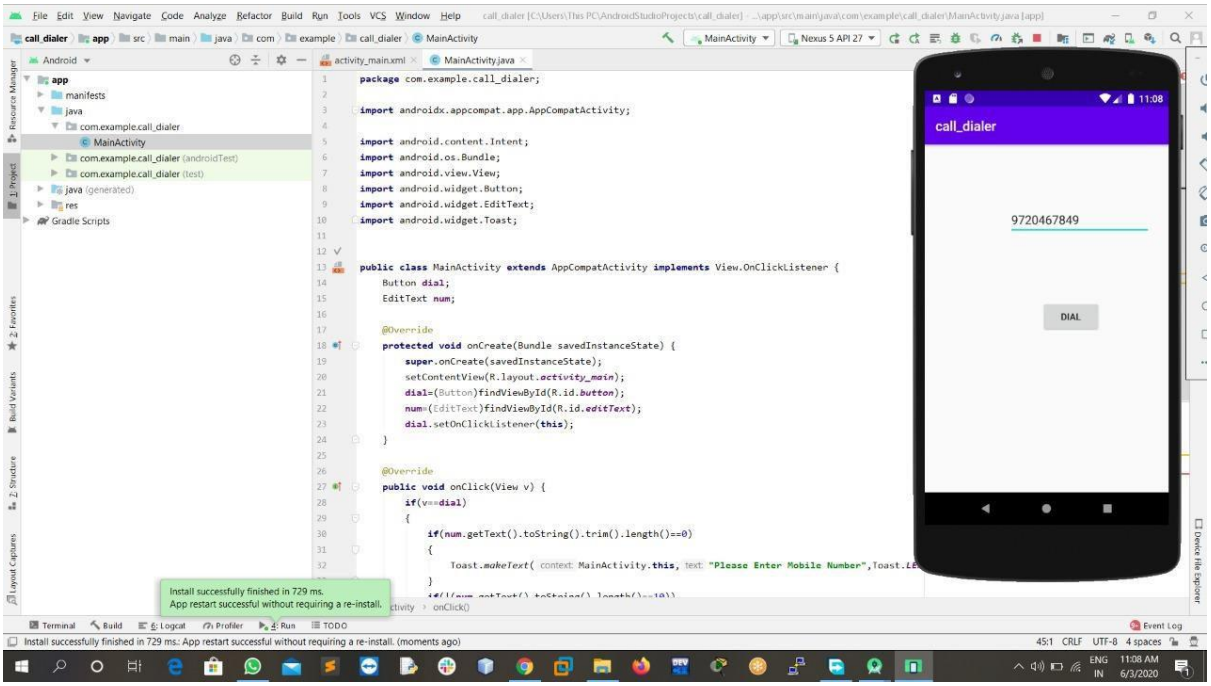

FIGURE-1.2

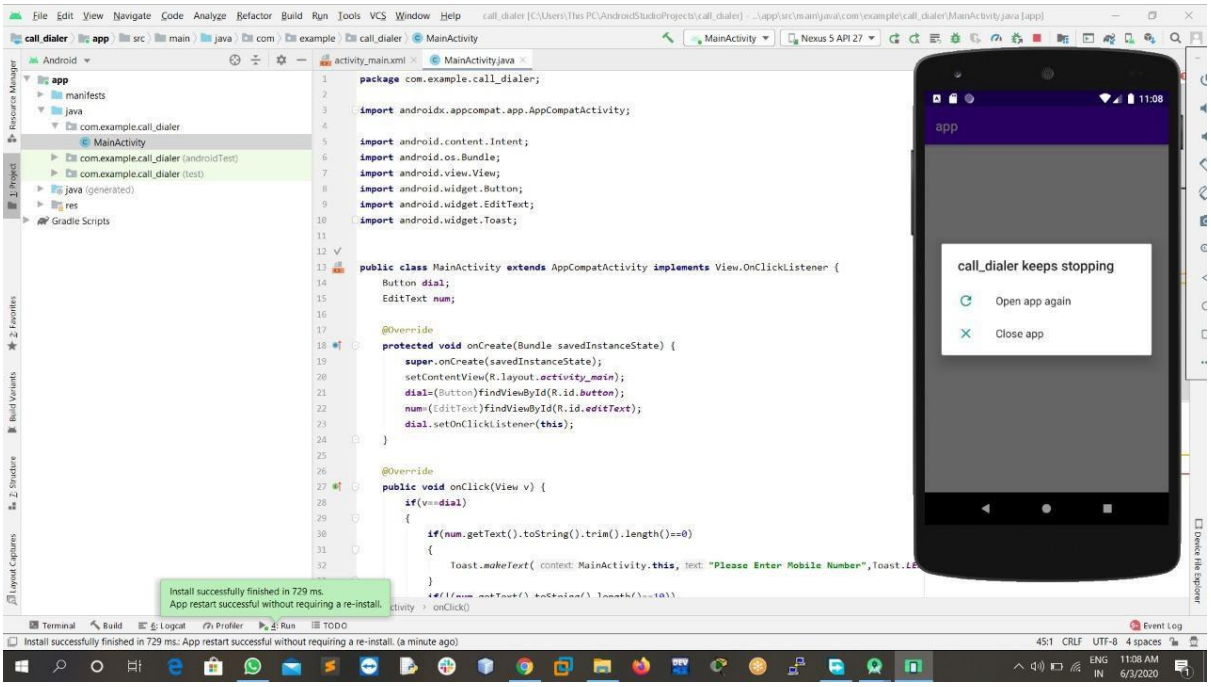

FIGURE-1.3

NOTE:- CALLING FACILITY IS NOT AVAILABLE IN THE EMULATOR OF ANDROID STUDIO WE HAVE TO PUT THIS PROJECT IN A CELL PHONE TO RUN IT COMPLETELY

CHECK BOX

In Android, CheckBox is a type of two state button either unchecked or checked in Android. Or you can say it is a type of on/off switch that can be toggled by the users. You should use checkbox when presenting a group of selectable options to users that are not mutually exclusive. CompoundButton is the parent class of CheckBox class.
In android there is a lot of usage of check box. For example, to take survey in Android app we can list few options and allow user to choose using CheckBox. The user will simply checked these checkboxes rather than type their own option in EditText. Another very common use of CheckBox is as remember me option in Login form.

Creating the Project in Android Studio

STEP-1: CREATING UI

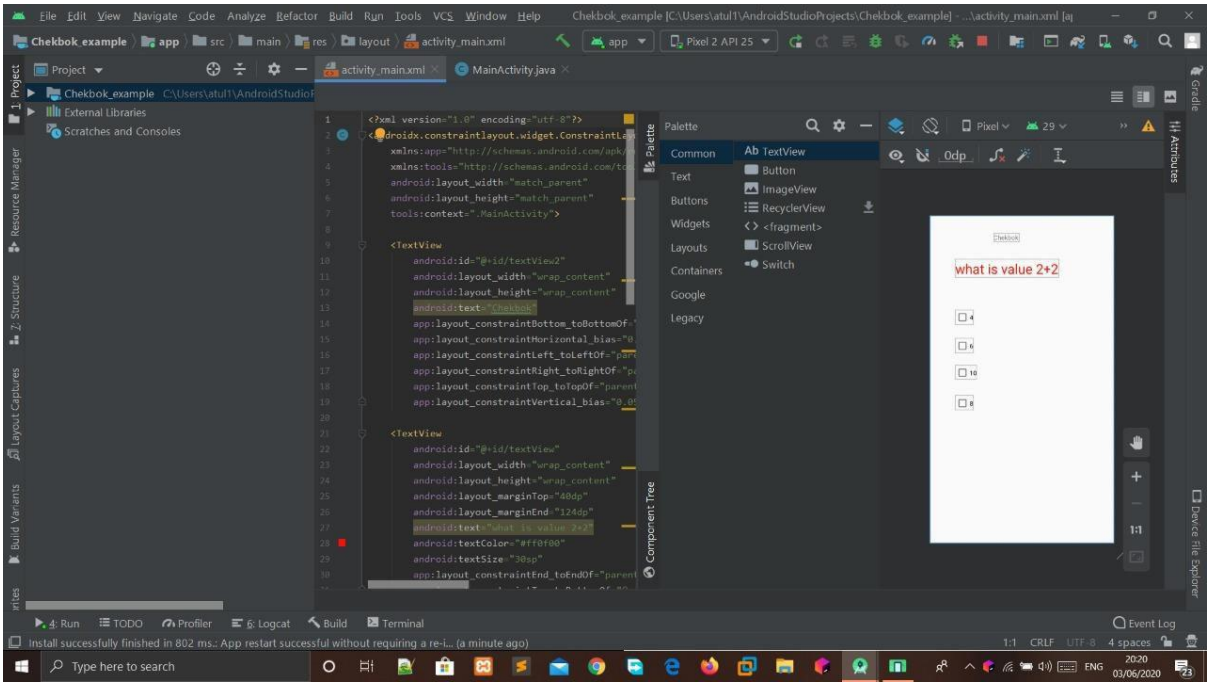

FIGURE-2.1

STEP-2: CREATING JAVA FILE

```
package com.example.chekbok_example;

import androidx.appcompat.app.AppCompatActivity;

import android.graphics.Color;
import android.os.Bundle;
import android.view.View;
import android.widget.CheckBox;
import android.widget.Toast;

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

```
CheckBox c1,c2,c3,c4;

@Override
protected void onCreate(Bundle savedInstanceState) {

    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);
    c1=(CheckBox)findViewById(R.id.checkBox);
    c2=(CheckBox)findViewById(R.id.checkBox2);
    c3=(CheckBox)findViewById(R.id.checkBox3);
    c4=(CheckBox)findViewById(R.id.checkBox4);
    c1.setOnClickListener(this);
    c2.setOnClickListener(this);
    c3.setOnClickListener(this);
    c4.setOnClickListener(this);
}

@Override
public void onClick(View v) {
    switch (v.getId())
    {
        case R.id.checkBox:
            c2.setChecked(false);
            c3.setChecked(false);
            c4.setChecked(false);
            c1.setTextColor(Color.GREEN);

            Toast.makeText(MainActivity.this,"Ans is Correct",Toast.LENGTH_LONG).show();
            break;
        case R.id.checkBox2:
            c1.setChecked(false);
            c3.setChecked(false);
            c4.setChecked(false);
            c2.setTextColor(Color.RED);
            Toast.makeText(MainActivity.this,"Ans is not
Correct",Toast.LENGTH_LONG).show();
            break;
        case R.id.checkBox3:
            c2.setChecked(false);
            c3.setTextColor(Color.RED);
            c1.setChecked(false);
            c4.setChecked(false);
            Toast.makeText(MainActivity.this,"Ans is not
Correct",Toast.LENGTH_LONG).show();
            break;
        case R.id.checkBox4:
            c2.setChecked(false);
```

```
            c3.setChecked(false);
            c1.setChecked(false);
            c4.setTextColor(Color.RED);
            Toast.makeText(MainActivity.this,"Ans is not
Correct",Toast.LENGTH_LONG).show();
            break;
        }
    }
}
```

STEP-3: RUN THE PROJECT

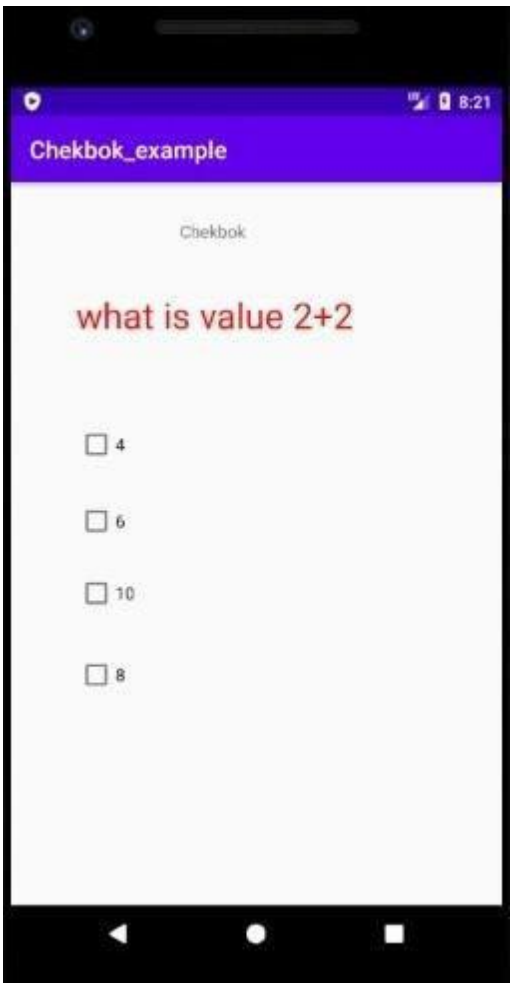

FIGURE-2.2

STEP-4: CLICK ON AN OPTION

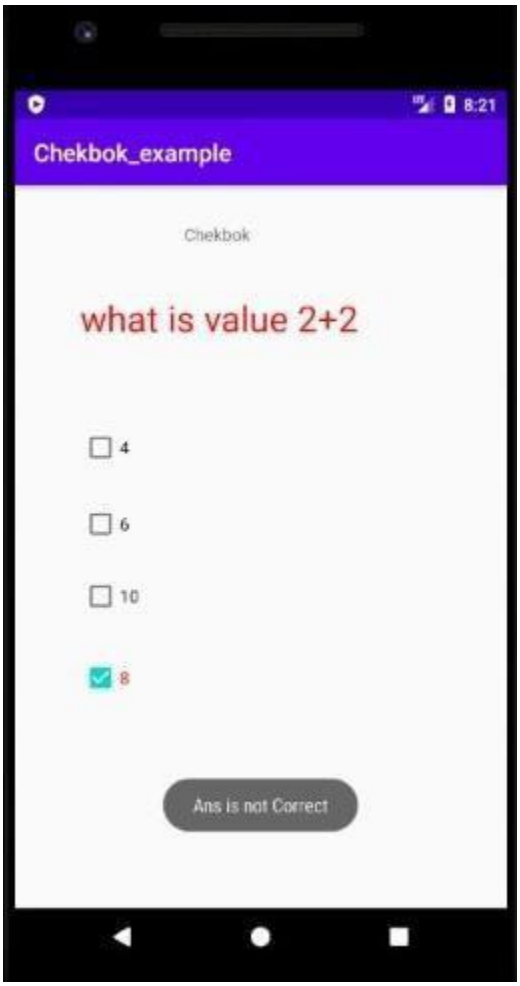

FIGURE-2.3

FIGURE-2.4

INTENT

Android application components can connect to other Android applications. This connection is based on a task description represented by an Intent object.

Intents are asynchronous messages which allow application components to request functionality from other Android components. Intents allow you to interact with components from the same applications as well as with components contributed by other applications. For example, an activity can start an external activity for taking a picture.

In Android the reuse of other application components is a concept known as task. An application can access other Android components to achieve a task. For example, from a component of your application you can trigger another component in the Android system, which manages photos, even if this component is not part of your application. In this component you select a photo and return to your application to use the selected photo.

PROJECT-1

Creating the Project in Android Studio
STEP-1:CREATING UI

MAIN ACTIVITY

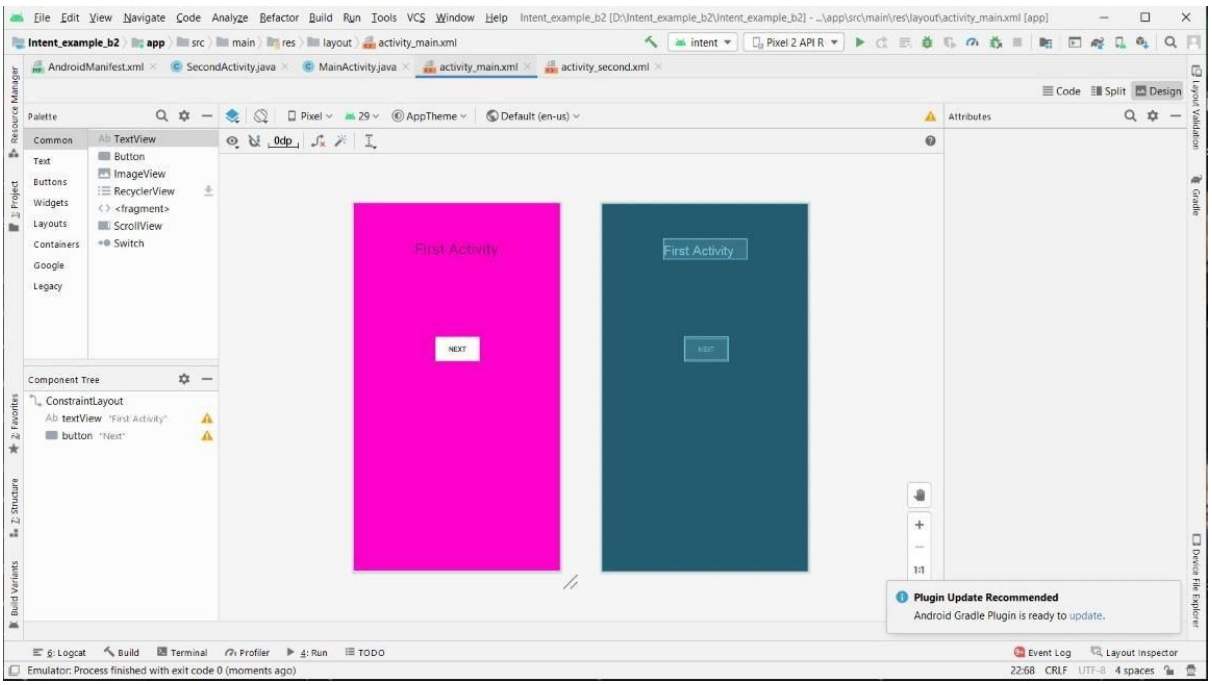

FIGURE-3.1

SECOND ACTIVITY

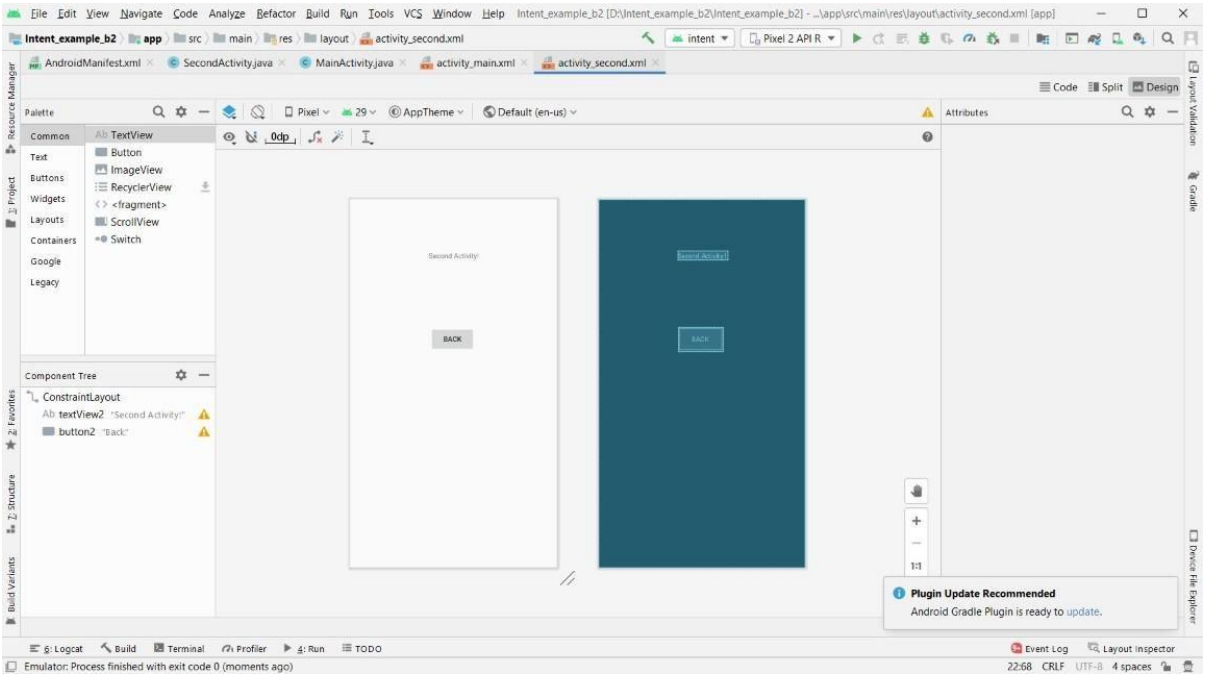

FIGURE-3.2

STEP-2: CREATING JAVA CODE

MAIN ACTIVITY

```
package com.example.intent_example_b2;
import androidx.appcompat.app.AppCompatActivity;
import android.content.Intent;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.Toast;
public class MainActivity extends AppCompatActivity {
   Button next;
   @Override
   protected void onCreate(Bundle savedInstanceState) {
      super.onCreate(savedInstanceState);
      setContentView(R.layout.activity_main);
      next=(Button)findViewById(R.id.button);
      next.setOnClickListener(new View.OnClickListener() {
         @Override
         public void onClick(View v) {
            Toast.makeText(MainActivity.this,"This is main
page",Toast.LENGTH_LONG).show();
            Intent obj=new Intent(MainActivity.this,SecondActivity.class);
            startActivity(obj);
         }
      });
   }
```

```
}
```

SECOND ACTIVITY

```
package com.example.intent_example_b2;
import android.content.Intent;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.Toast;
import androidx.appcompat.app.AppCompatActivity;
public class SecondActivity extends AppCompatActivity {
   Button back;
   @Override
   protected void onCreate(Bundle savedInstanceState) {
      super.onCreate(savedInstanceState);
      setContentView(R.layout.activity_second);
      back=(Button)findViewById(R.id.button2);
      back.setOnClickListener(new View.OnClickListener() {
         @Override
         public void onClick(View v) {
            Toast.makeText(SecondActivity.this,"This is Second
page",Toast.LENGTH_LONG).show();
            Intent obj=new Intent(SecondActivity.this,MainActivity.class);
            startActivity(obj);
         }
      });

   }
}
```

STEP-3: RUN THE PROJECT

FIGURE-3.3

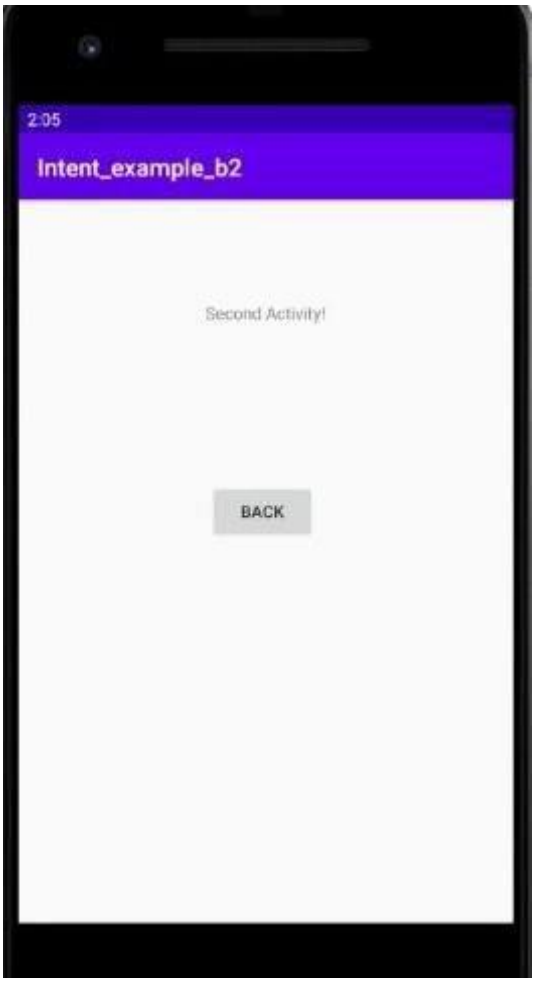

FIGURE-3.4

FIGURE-3.5

PROJECT-2

Creating the Project in Android Studio

STEP-1: CREATING UI

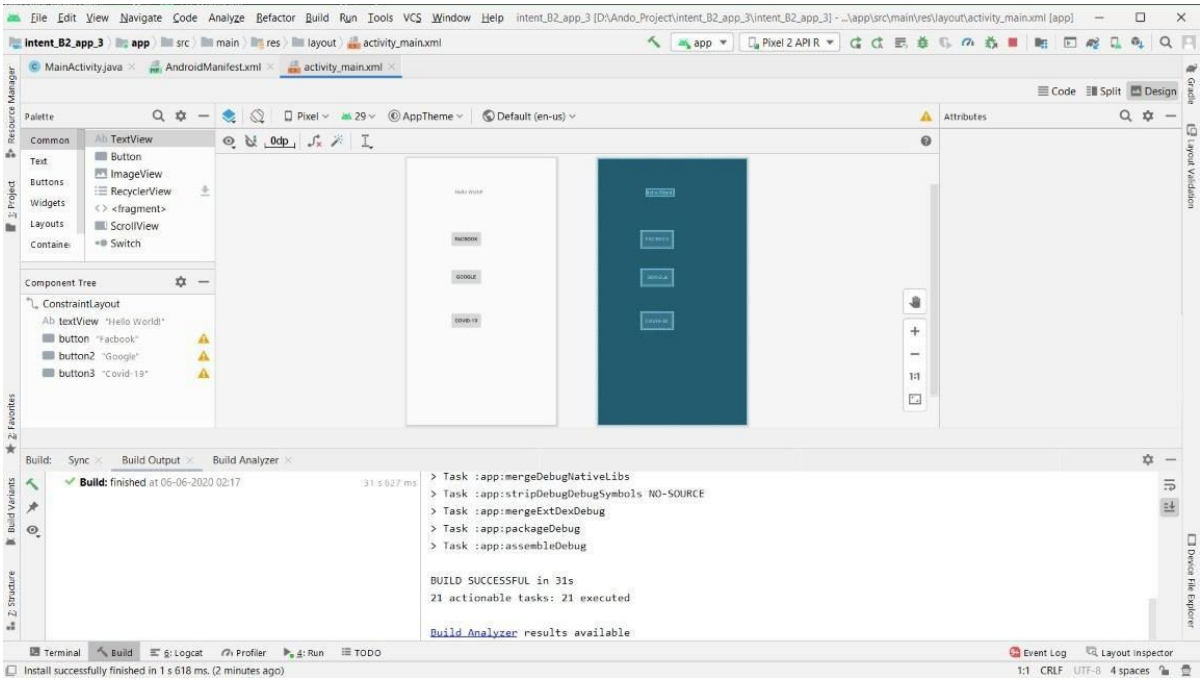

FIGURE-3.6

STEP-2: JAVA CODE

```
package com.example.intent_b2_app_3;

import androidx.appcompat.app.AppCompatActivity;

import android.content.Intent;

import android.net.Uri;

import android.os.Bundle;

import android.view.View;

import  android.widget.Button;


public class MainActivity extends AppCompatActivity {

    Button b1,b2,b3;


    @Override
```

```
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        b1=(Button)findViewById(R.id.button);
        b2=(Button)findViewById(R.id.button2);
        b3=(Button)findViewById(R.id.button3);
        b3.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                Intent c=new Intent(Intent.ACTION_VIEW,
Uri.parse("https://www.mohfw.gov.in/"));
                startActivity(c);
            }
        });
        b1.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                Intent c=new Intent(Intent.ACTION_VIEW,Uri.parse("https://www.google.com/"));
                startActivity(c);
            }
        });
    }
}
```

STEP-3: RUN PROJECT

FIGURE-3.7

FIGURE-3.8

SHARED PREFERENCE

One of the most Interesting Data Storage option **Android** provides its users is **Shared Preferences**. **Shared Preferences** is the way in which one can store and retrieve small amounts of primitive data as key/value pairs to a file on the device storage such as String, int, float, Boolean that make up your preferences in an XML file inside the app on the device storage. **Shared Preferences** can be thought of as a dictionary or a key/value pair. For example, you might have a key being “username” and for the value, you might store the user’s username. And then you could retrieve that by its key (here username). You can have simple shared preferences API that you can use to store preferences and pull them back as and when needed. Shared Preferences class provides APIs for reading, writing and managing this data.

Shared Preferences is suitable in different situations. For example, when the user’s settings need to be saved or to store data that can be used in different activities within the app. As you know, onPause() will always be called before your activity is placed in the background or destroyed, So for the data to be saved persistently we prefer saving it in onPause(), which could be restored in onCreate() of the activity. The data stored using shared preferences are kept private within the scope of the application. However, shared preferences are different from that activity’s instance state.

Creating the Project in Android Studio

STEP-1: CREATING UI

FIRST ACTIVITY

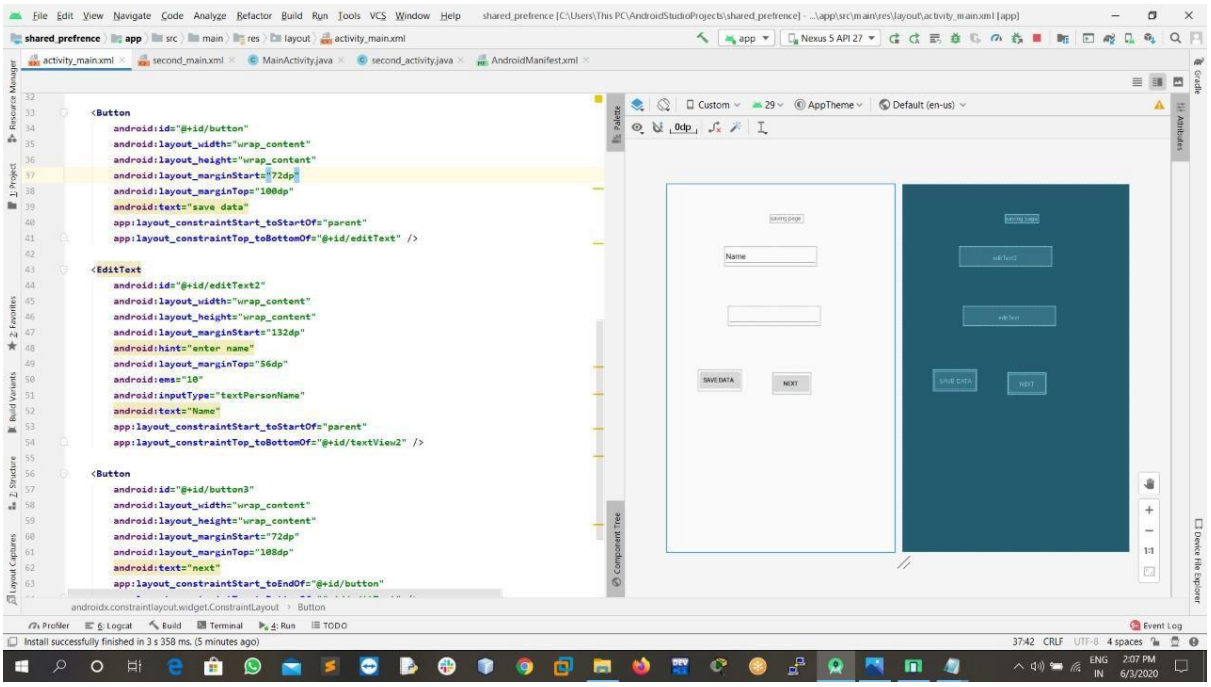

FIGURE-4.1

SECOND ACTVITY

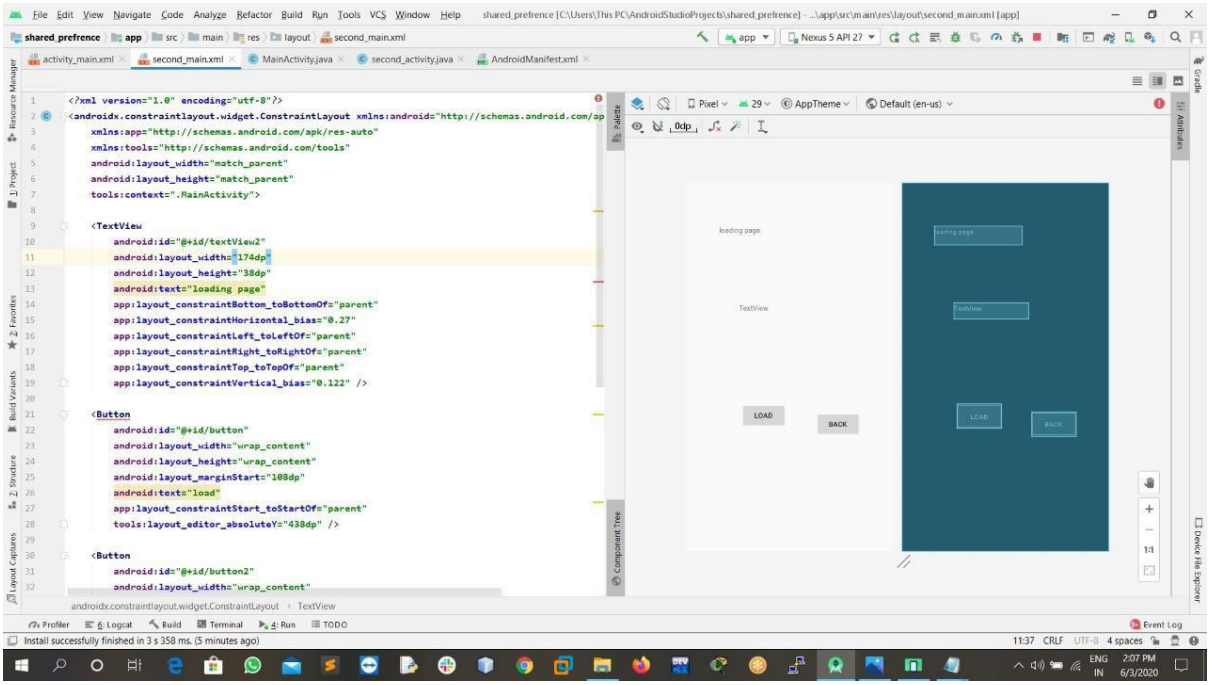

FIGURE-4.2

STEP-2: JAVA CODE

MAIN ACTVITY

```
package com.example.shared_prefrence;

import android.content.Context;

import android.content.Intent;

import android.content.SharedPreferences;

import android.os.Bundle;

import android.view.View;

import android.widget.Button;

import android.widget.TextView;

import android.widget.Toast;

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

```
public class second_activity extends AppCompatActivity{
    TextView s_name , s_pass;
    Button load , back;
    String myval = "abs";

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        s_name  =(TextView)findViewById(R.id.textView2);
        s_pass =(TextView)findViewById(R.id.textView);
        back = (Button)findViewById(R.id.button2);

        load = (Button)findViewById(R.id.button);
        back.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                Toast.makeText(second_activity.this," this main page",
Toast.LENGTH_LONG).show();
                Intent i= new Intent(second_activity.this , MainActivity.class);
                startActivity(i);
            }
        });
        load.setOnClickListener(new View.OnClickListener() {
```

```
        @Override
        public void onClick(View v) {
           SharedPreferences sp = getSharedPreferences("mydata", Context.MODE_PRIVATE);
           String na = sp.getString( "k_name", "myval") ;
           String pa = sp.getString("k_pass", "myval");

           if(na.equals("myval") || pa.equals("myval"))
           {

              Toast.makeText(second_activity.this ,"no data found",
Toast.LENGTH_LONG).show();
           }else{

              Toast.makeText(second_activity.this ,"no data found",
Toast.LENGTH_LONG).show();
              s_name.setText(na);
              s_pass.setText(pa);
           }
        }
      });
   }
}
```

SECOND ACTIVITY

```
package com.example.shared_prefrence;

import androidx.appcompat.app.AppCompatActivity;

import android.content.Context;

import android.content.Intent;

import android.content.SharedPreferences;

import android.os.Bundle;

import android.view.View;

import android.widget.Button;

import android.widget.EditText;

import android.widget.Toast;

import android.content.Intent;

import android.os.Bundle;

import android.view.View;

import android.widget.Button;

import android.widget.EditText;

import android.widget.Toast;


public class MainActivity extends AppCompatActivity {

    Button save ,next;

    EditText name , pass;


    @Override

    protected void onCreate(Bundle savedInstanceState) {

        super.onCreate(savedInstanceState);
```

```
        setContentView(R.layout.activity_main);
        save = (Button)findViewById(R.id.button);
        next =(Button)findViewById(R.id.button3);
        name = (EditText)findViewById(R.id.editText2);
        pass = (EditText)findViewById(R.id.editText);
        next.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                Toast.makeText(MainActivity.this , "this is second page"
,Toast.LENGTH_LONG).show();
                Intent i = new Intent(MainActivity.this , second_activity.class);
                startActivity(i);
            }
        });
            save.setOnClickListener(new View.OnClickListener() {
                @Override
                public void onClick(View v) {
                    SharedPreferences sp = getSharedPreferences("mydata",
Context.MODE_PRIVATE);
                    SharedPreferences.Editor ed = sp.edit();
                    ed.putString("k_name",name.getText().toString());
                    ed.putString("k_pass",pass.getText().toString());
                    ed.commit();
                    Toast.makeText(MainActivity.this,"data has been saved!
",Toast.LENGTH_LONG).show();
                }
```

```
        });
    }
}
```

STEP-2: RUN PROJECT

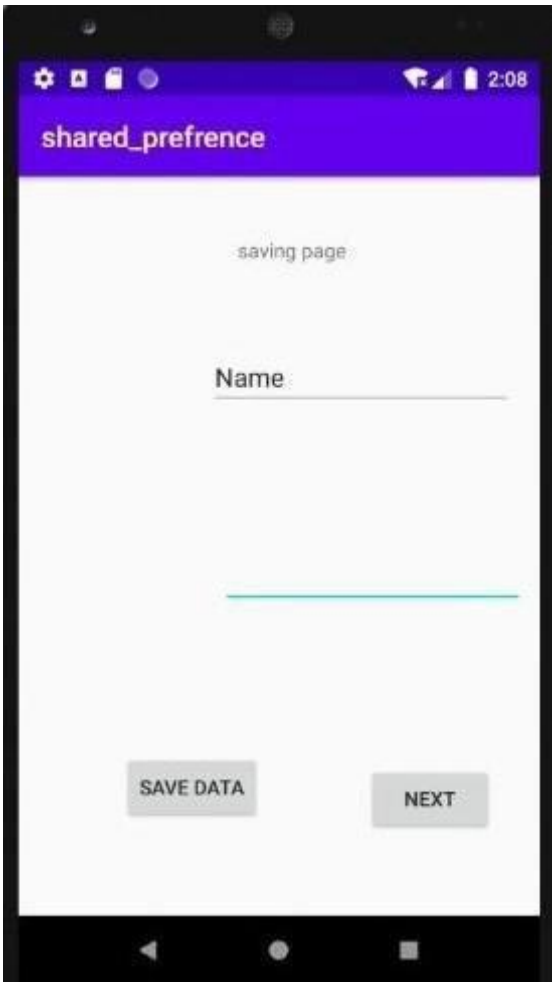

FIGURE-4.3

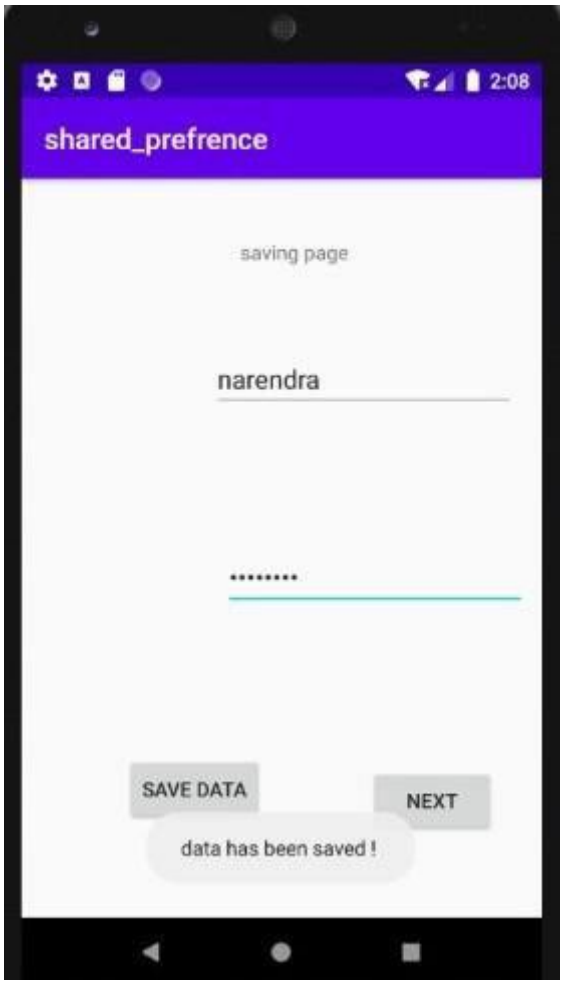

FIGURE-4.4

WIFI TURN On/Off

In this project we will deal with the hardware of our system.

Before turning on WIFI we have to ensure that it must be off and before turning it off we have to ensure that it must be on.

Creating the Project in Android Studio

STEP-1:CREATING UI

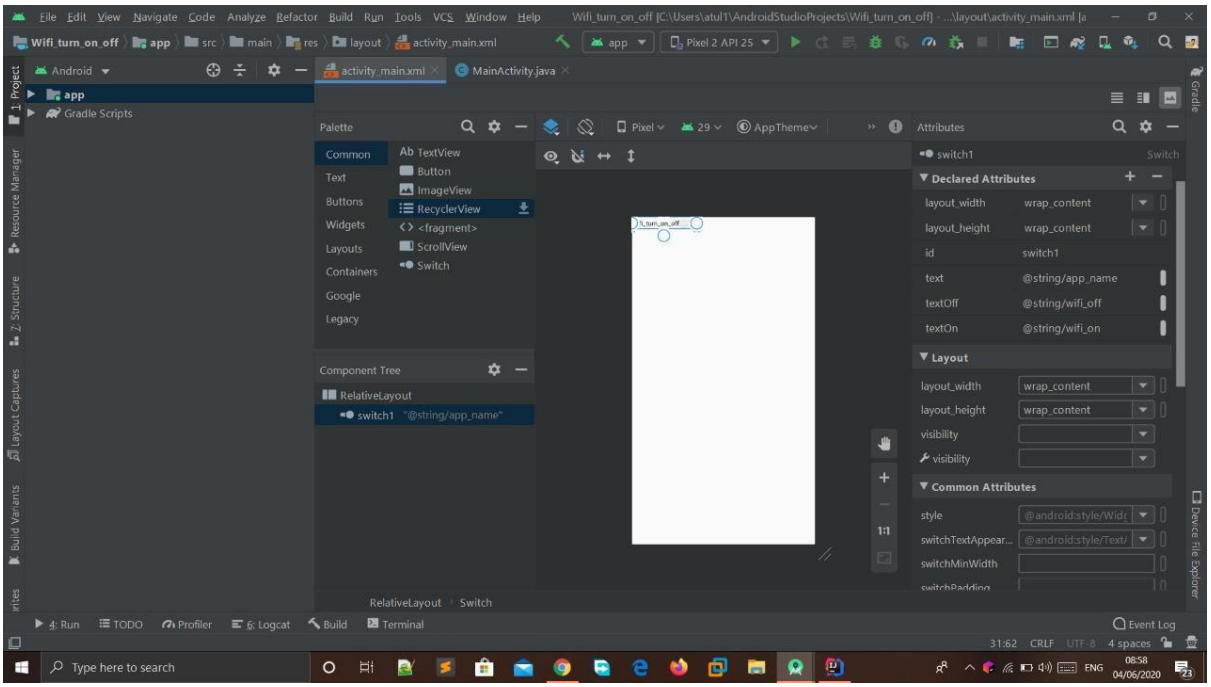

FIGURE-4.1

STEP-2: JAVA CODE

```
package com.example.wifi_turn_on_off;

import androidx.appcompat.app.AppCompatActivity;

import android.content.Context;

import android.net.wifi.WifiManager;

import android.os.Bundle;

import android.widget.CompoundButton;

import android.widget.Switch;
```

```
import android.widget.Toast;
public class MainActivity extends AppCompatActivity {
    Switch sw;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        sw=(Switch)findViewById(R.id.switch1);
        sw.setOnCheckedChangeListener(new CompoundButton.OnCheckedChangeListener() {
            @Override
            public void onCheckedChanged(CompoundButton buttonView, boolean isChecked) {
                if(isChecked)
                {
                    mywifiControl(true);
                    Toast.makeText(MainActivity.this,"Wifi On",Toast.LENGTH_SHORT).show();
                }
                else
                {
                    mywifiControl(false);
                    Toast.makeText(MainActivity.this,"Wifi Off",Toast.LENGTH_SHORT).show();
                }

            }
```

```
        });


    }


    public void mywifiControl(boolean b) {
        WifiManager wifiobj=(WifiManager) this.getSystemService(Context.WIFI_SERVICE);
        if((b==true) && !(wifiobj.isWifiEnabled()))
        {
            wifiobj.setWifiEnabled(true);
        }
        else if ((b==false ) &&wifiobj.isWifiEnabled())
        {
            wifiobj.setWifiEnabled(false);
        }
    }
}
```

STEP-3: RUN PROJECT

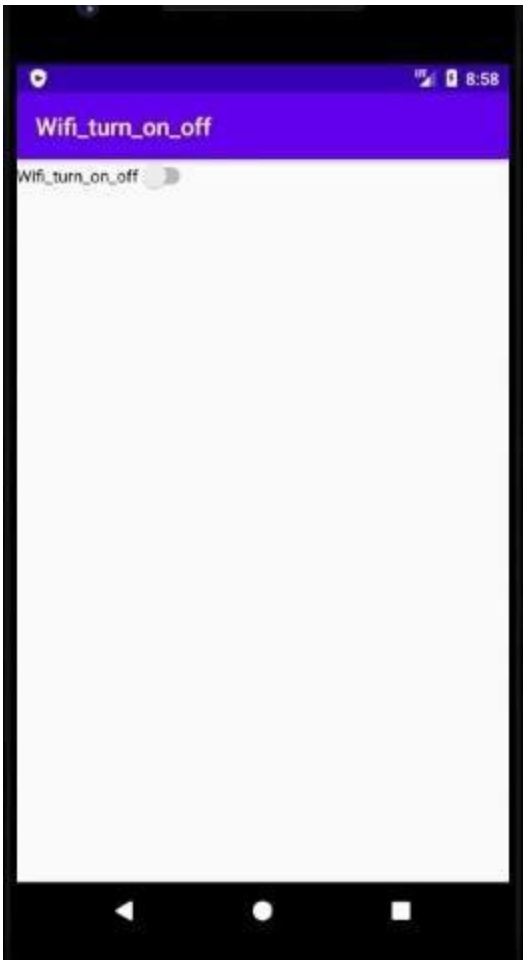

FIGURE-4.2

SUMMARY

The Project Consist of some apps taught to develop in the 30 hours of session in the internship by EIS.

Starting from the simple call dialler that help us to call anybody, A widely used checkbox app helps to sort out many problems, intent that allow us to have a connection between two activities, shared preference that is used to store data and at last we have some interaction with our hardware in the form of turning on and off a WIFI.