Lab 1

Step 1: Create a New Project and Select the Empty Activity.

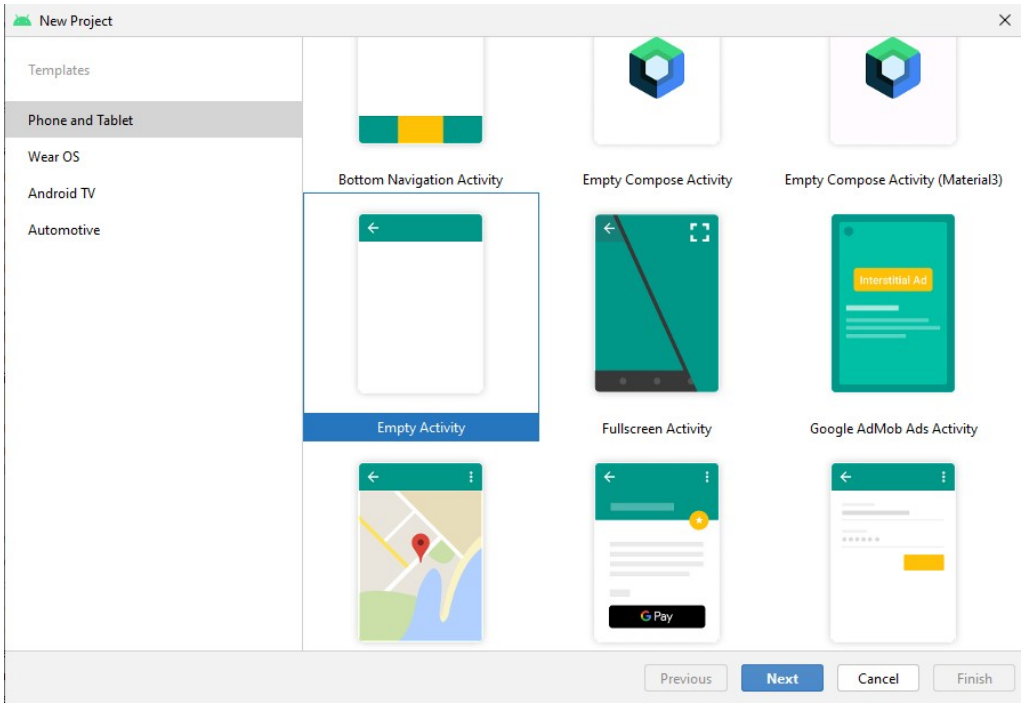


Figure 1 Selecting the Empty Activity

Step2: Name the Project and Select the Language as Java and click on finish.

Note: Make sure you have an active internet connection.

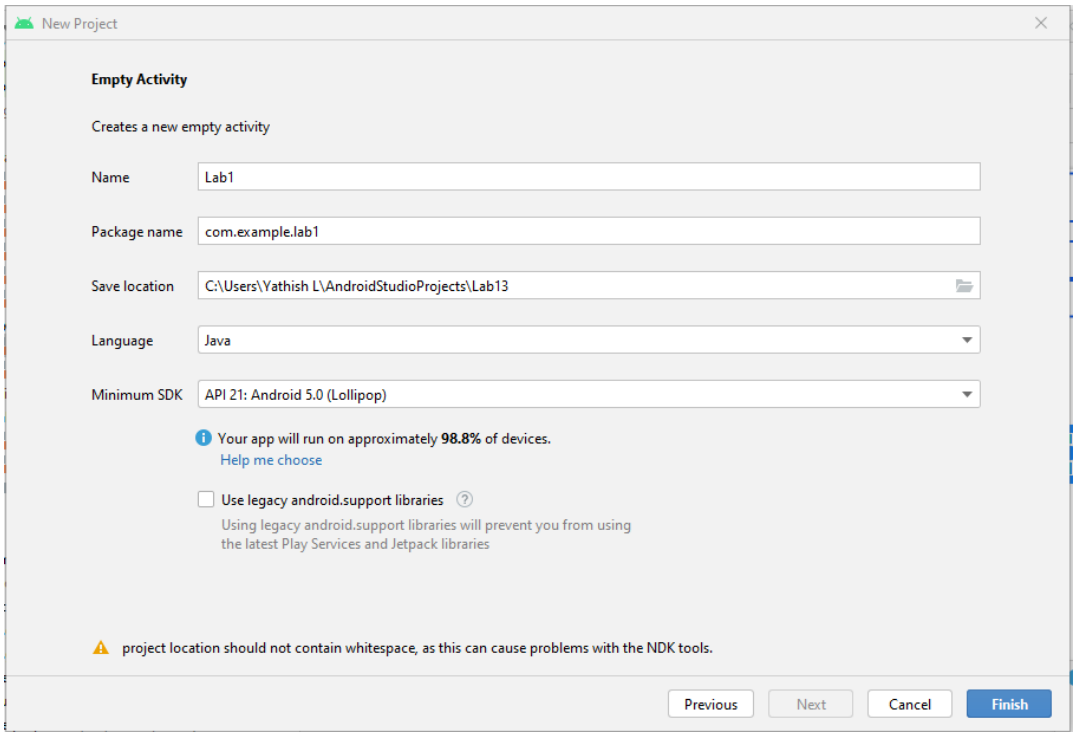
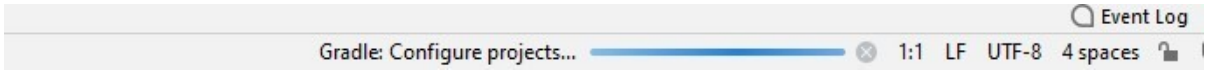
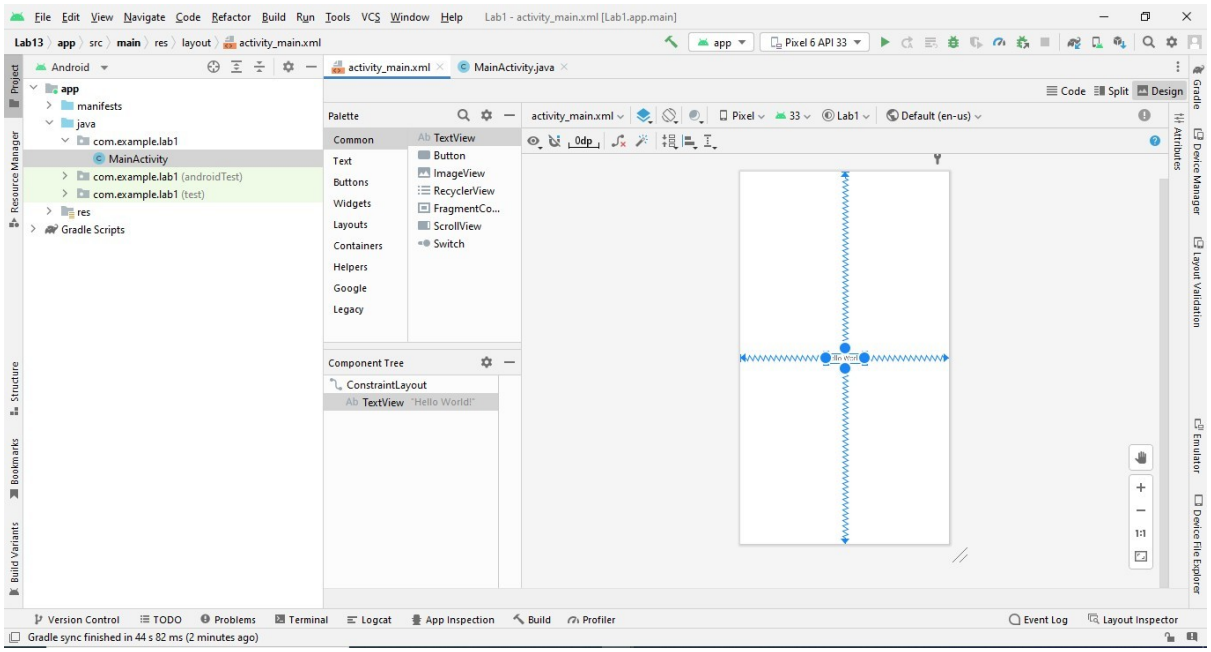


Figure 2 Name the project and select the Programming language.

Note: Please Wait till gradle build is completed before coding or designing.



After the Gradle Build is Completed.



Step3: Add the EditText and Button Components to Activity_main.xml

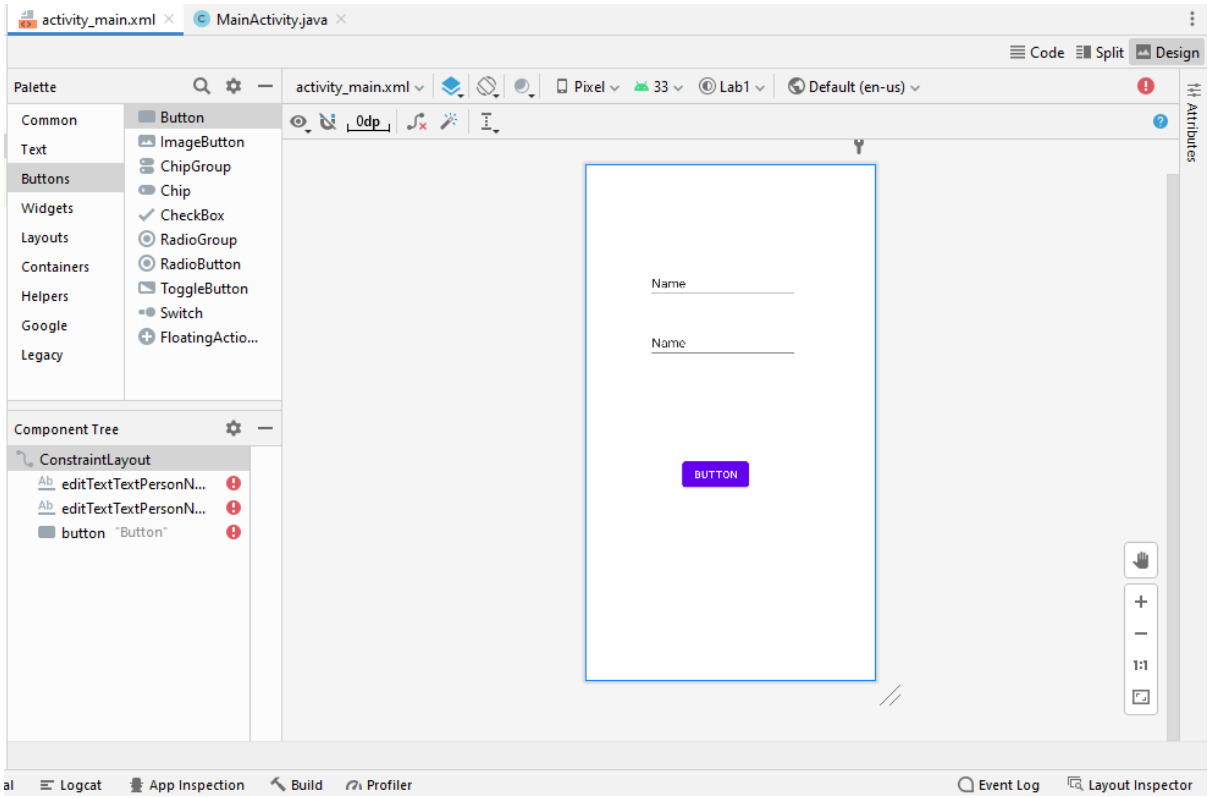
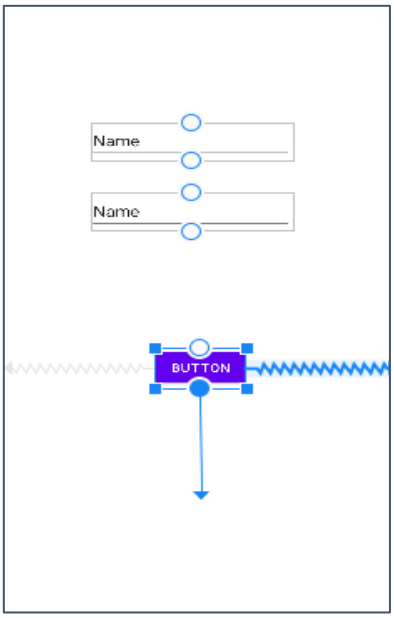
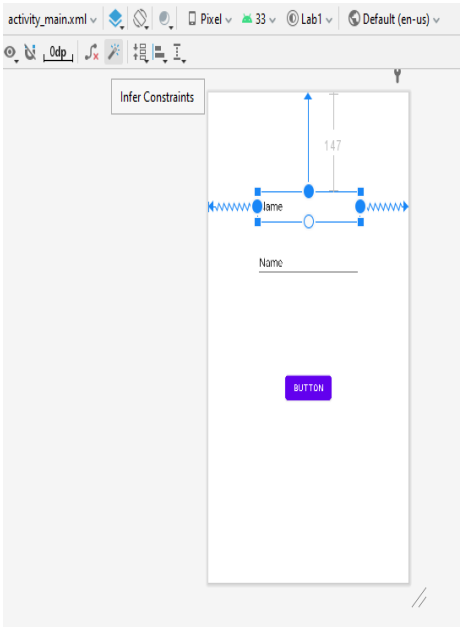


Figure There is No constraint assigned to Edit text or Button, you can do this is in two Ways one using the mapping the Constraint manually or using Infer Constraint

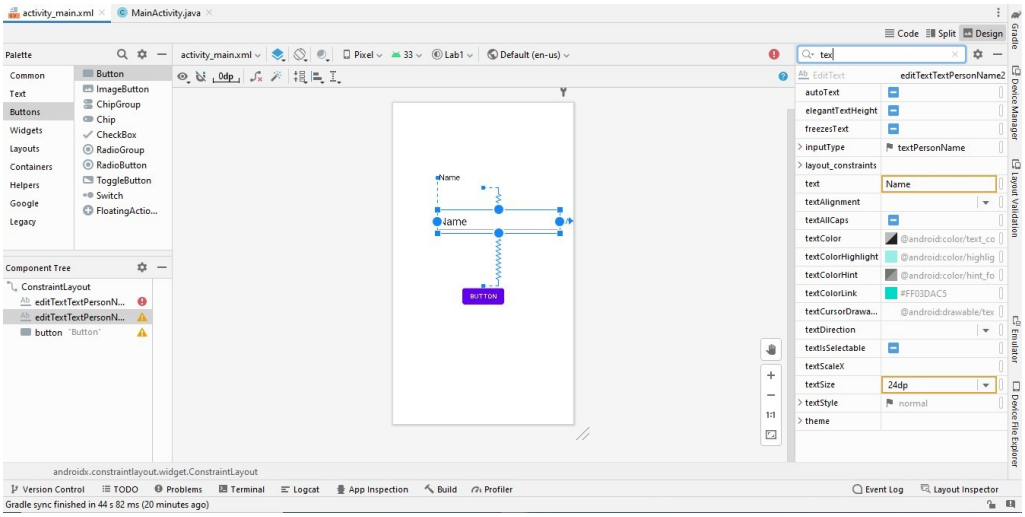


Manual Assigning

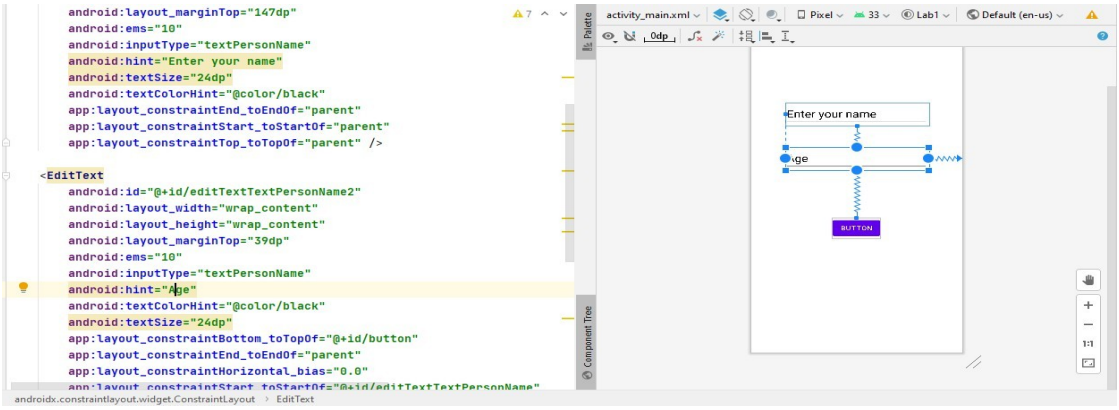


Infer Constraint

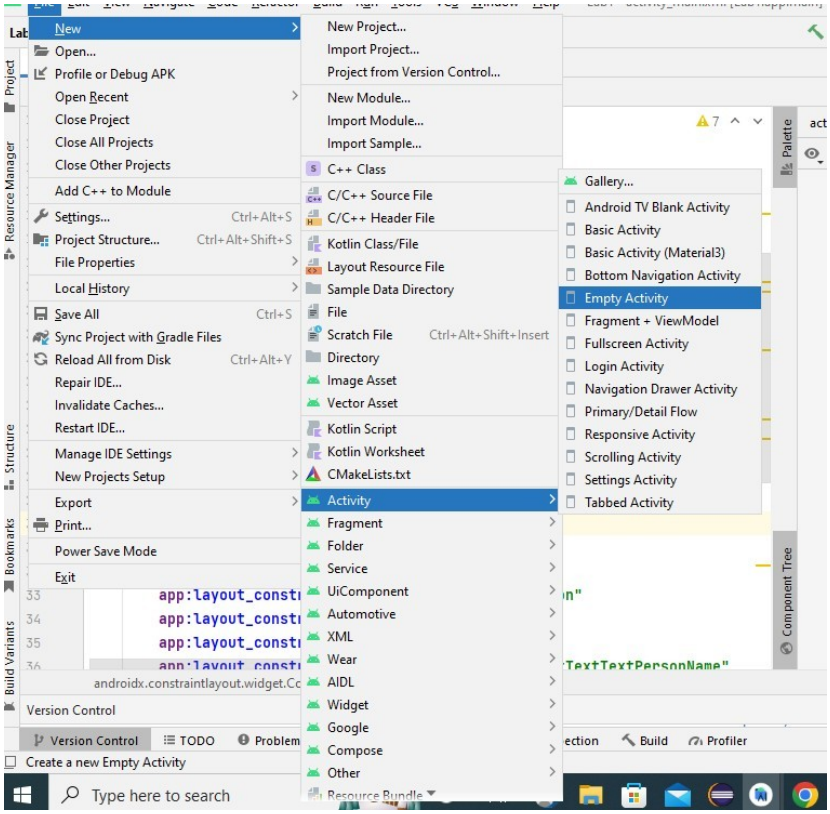
To change the Attributes of a component you can use the Attribute panel or in xml code and use can search the Attributes in the Attributes panel



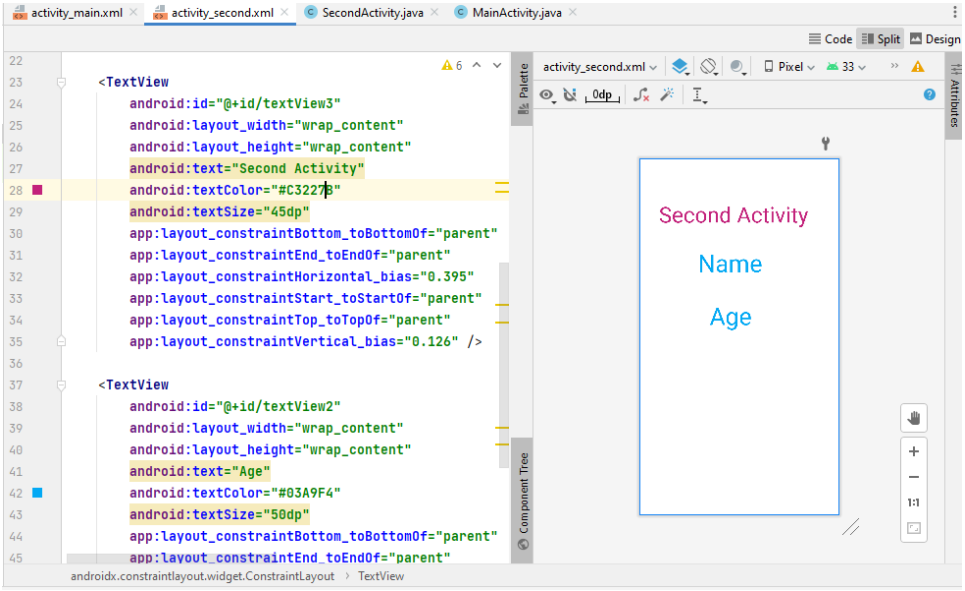
You can also do Change Attributes in the xml File.



Step4: create Another Activity as Second Activity



Please Follow the Above Step and create the Second Activity how you did for Main Activity.



Then You can Start Coding

XML code of Main Activity

```
<EditText
    android:id="@+id/editTextTextPersonName"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginTop="147dp"
    android:ems="10"
    android:inputType="textPersonName"
    android:hint="Enter your name"
    android:textSize="24dp"
    android:textColorHint="@color/black"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent" />

<EditText
    android:id="@+id/editTextTextPersonName2"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginTop="39dp"
    android:ems="10"
    android:inputType="textPersonName"
    android:hint="Age"
    android:textColorHint="@color/black"
    android:textSize="24dp"
    app:layout_constraintBottom_toTopOf="@+id/button"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.0"
    app:layout_constraintStart_toStartOf="@+id/editTextTextPersonName"
    app:layout_constraintTop_toBottomOf="@+id/editTextTextPersonName"
    app:layout_constraintVertical_bias="0.092" />

<Button
    android:id="@+id/button"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginBottom="267dp"
    android:text="Click"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent" />
```

Xml Code for Second Activity

```
<TextView
    android:id="@+id/textView"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Name"
    android:textSize="50dp"
    android:textColor="#03A9F4"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.436"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent"
```

```
    app:layout_constraintVertical_bias="0.272" />

<TextView
    android:id="@+id/textView3"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Second Activity"
    android:textColor="#C3227B"
    android:textSize="45dp"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.395"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintVertical_bias="0.126" />

<TextView
    android:id="@+id/textView2"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Age"
    android:textColor="#03A9F4"
    android:textSize="50dp"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.447"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/textView"
    app:layout_constraintVertical_bias="0.1" />
```

The Java Code of the Main Activity

```
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);
    Button b = findViewById(R.id.button);
    EditText name=findViewById(R.id.editTextTextPersonName);
    EditText age = findViewById(R.id.editTextTextPersonName2);
    // it will give you the reference to the view in XML layouts by
searching its ID
    b.setOnClickListener(new View.OnClickListener() {
        @Override
        /*These methods will be called by the Android framework when the
View
to      which the listener has been registered
        is triggered by user interaction with the item in the UI*/
        public void onClick(View view) {
            String Dname=name.getText().toString();
            String Dage =age.getText().toString();
            /* By using the gettext we can get the user input from the
activity*/
            Intent i = new Intent(MainActivity.this,SecondActivity.class);
            //Intent is the message that is passed between components such
as activities,
            // content providers, broadcast receivers, services etc
            i.putExtra("name",Dname);
            i.putExtra("age",Dage);
            //Add extended data to the intent.
            startActivity(i);
```

```
            //The startActivity(Intent) method is used to start a new
activity,
            // which will be placed at the top of the activity stack.
        }
    });

}
```

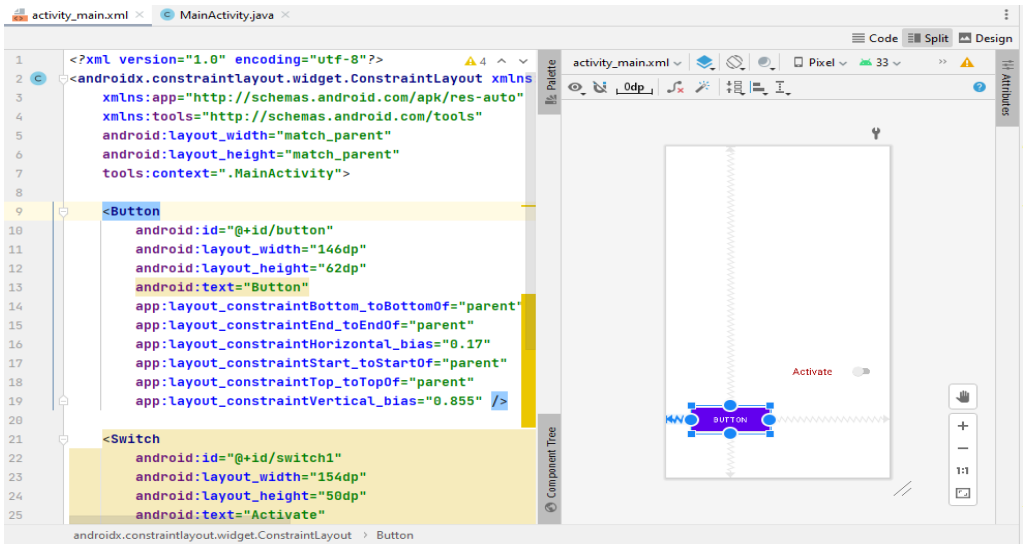
The Java Code of the Second Activity

```
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_second);
    Intent i = getIntent();
    //Return the intent that started this activity
    String name=i.getStringExtra("name");
    String age = i.getStringExtra("age");
    //this is used to get value which is passed in the Activity
    TextView t1=findViewById(R.id.textView);
    TextView t2 = findViewById(R.id.textView2);
    t1.setText(name);
    t2.setText(age);
    //this is used to send the data to the xml layout through java code
}
```

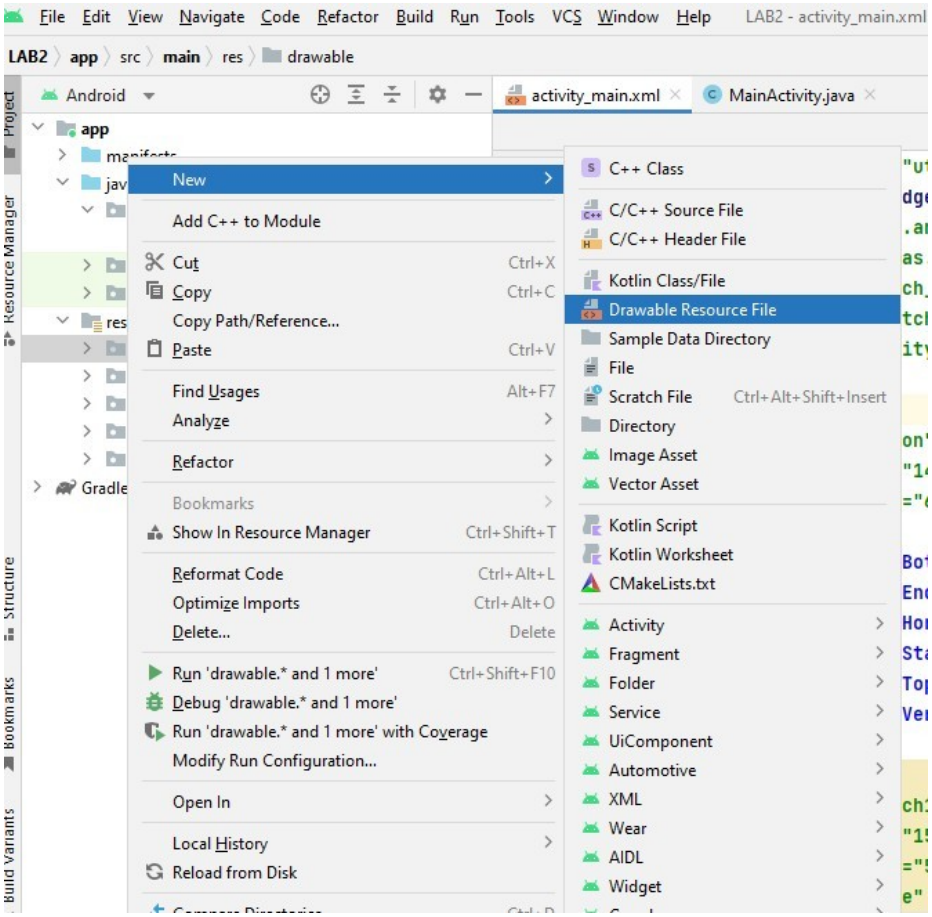
Lab 2

Follow the Step1 of Lab1 Program

Step 2: Add the button, switch and constraint layout in activity_main.xml



Step 3: Create A drawable file in **res->drawable** folder



Write the code in the Drawable file(shape.xml)

```
<?xml version="1.0" encoding="utf-8"?>
<shape xmlns:android="http://schemas.android.com/apk/res/android"
android:shape="rectangle">
 <gradient android:startColor="#123456" android:type="linear"
android:endColor="#456875"></gradient>
    <corners android:radius="18dp"></corners>
</shape>
//This code is for design the graphics of the button
//Please change in the theme if Drawable is not working
```

Repeat the step3 and create a new Drawable file

Write the code in the Drawable file (buttong)

```
<?xml version="1.0" encoding="utf-8"?>
<selector xmlns:android="http://schemas.android.com/apk/res/android">
<item android:state_pressed="true"
android:drawable="@drawable/shape"></item>
    <itemandroid:state_pressed="false"android:drawable="@color/black"/>
</selector>
```

//This for the button graphic changed when button is pressed

Xml code

```
<Button
    android:id="@+id/button"
    android:layout_width="146dp"
    android:layout_height="62dp"
    android:text="Press"
    android:background="@drawable/buttong"
```

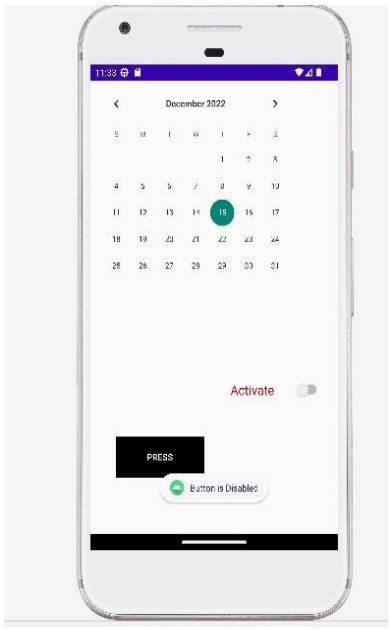
```
    android:textColor="@color/white"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.162"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintVertical_bias="0.863" />

<Switch
    android:id="@+id/switch1"
    android:layout_width="154dp"
    android:layout_height="0dp"
    android:layout_marginBottom="204dp"
    android:text="Activate"
    android:textColor="#AE0C0C"
    android:textSize="20dp"

    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.904"
    app:layout_constraintStart_toStartOf="parent"

    app:layout_constraintVertical_bias="0.903" />

<androidx.constraintlayout.widget.ConstraintLayout
    android:id="@+id/Layout"
```

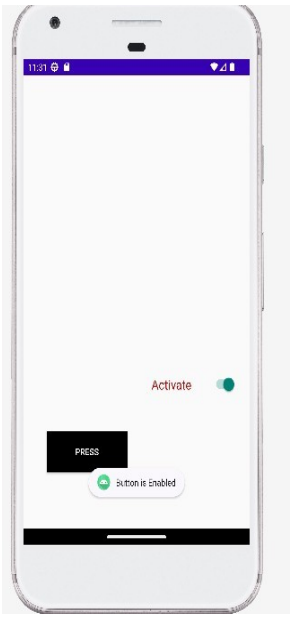
```
    android:layout_width="0dp"
    android:layout_height="0dp"
    android:layout_marginStart="1dp"
    android:layout_marginTop="1dp"
    android:layout_marginEnd="1dp"
    android:layout_marginBottom="1dp"
    app:layout_constraintBottom_toTopOf="@+id/switch1"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent">

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

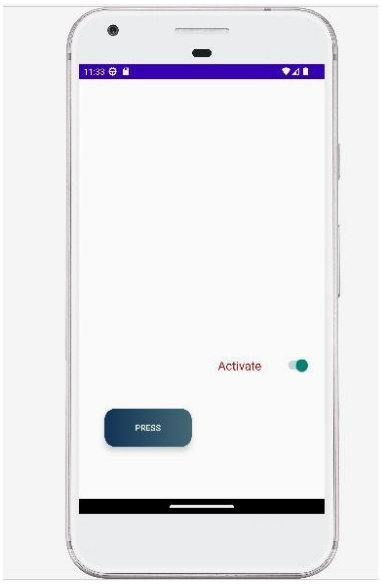
Java Code

```
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);
    Button button = findViewById(R.id.button);
    Switch switch1=findViewById(R.id.switch1);
// it will give you the reference to the view in XML layouts by searching
its ID

    ConstraintLayout layout = findViewById(R.id.Layout);
    CalendarView cal = new CalendarView(this);
// create the calenderview dynamically
    layout.addView(cal);
//adding the calenderView dyanamically to the constraintLayout
    switch1.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View view) {
            if(switch1.isChecked())
            { //if the switch is checked the calender view is removed
               // and button is enabled
               button.setEnabled(true);
               layout.removeView(cal);
               Toast.makeText(MainActivity.this,"Button is
        Enabled",Toast.LENGTH_LONG.show();



            }
            else
            {//if the switch is unchecked the calender view is removed
               // and button is disabled

        Toast.makeText(MainActivity.this,"Button is
        Disabled",Toast.LENGTH_LONG).show();


                button.setEnabled(false);
                layout.addView(cal);
            }
        }
    });

}
```

Output





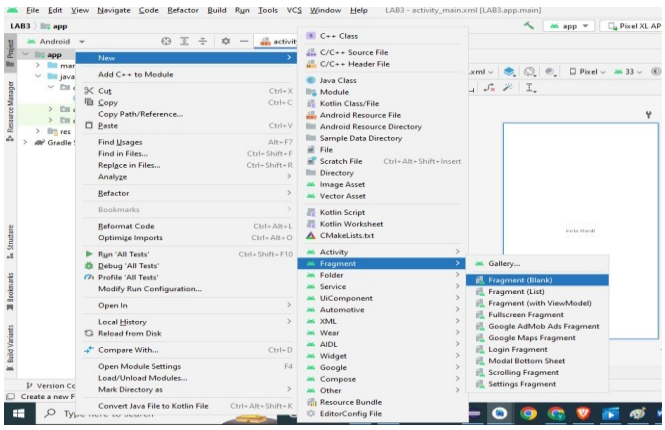


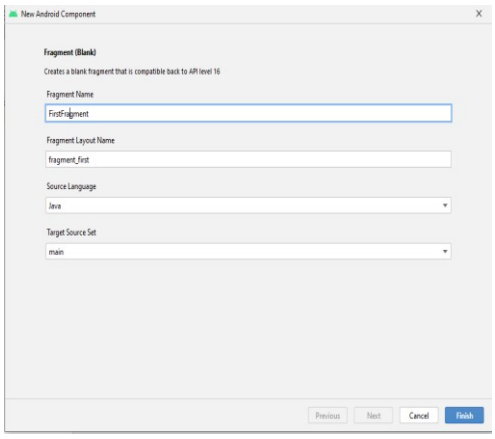
Lab 3

Follow the Step1 of Lab1 Program

Step2 Create the two Fragment in the project.

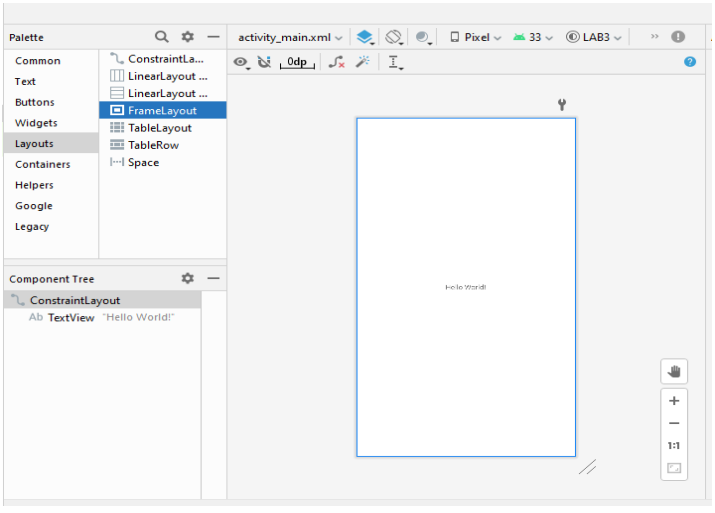
Note: please give valid name to fragment

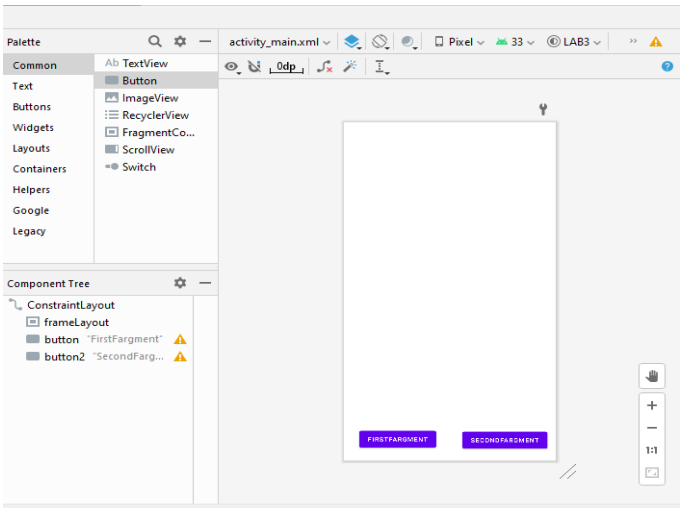




*please repeat the step once Again Create the second Fragment(blank)

Step4: Add the Frame Layout and two buttons from the Palette to the main activity





Xml Code for the activity_main.xml

```
<FrameLayout
    android:id="@+id/frameLayout"
    android:layout_width="411dp"
    android:layout_height="0dp"
    android:layout_marginStart="1dp"
    android:layout_marginEnd="1dp"
    android:layout_marginBottom="56dp"
    app:layout_constraintBottom_toTopOf="@+id/button"
    app:layout_constraintEnd_toEndOf="parent"
```

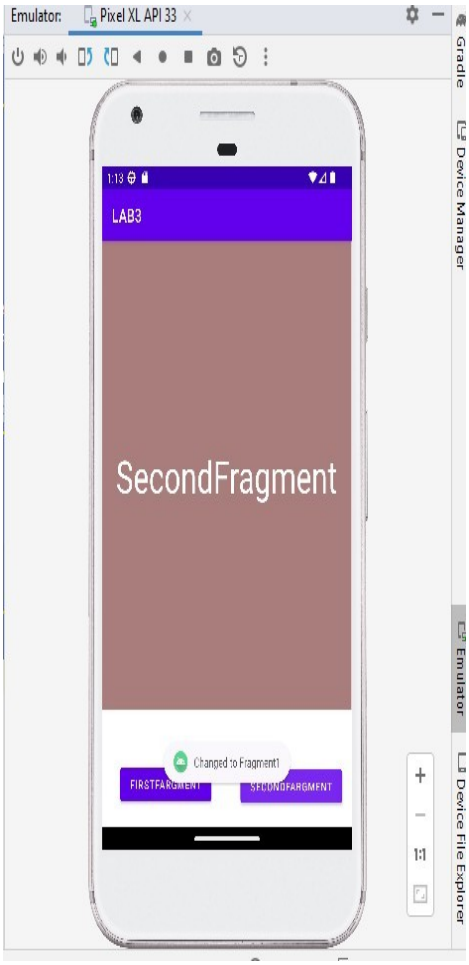
```
    app:layout_constraintHorizontal_bias="1.0"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintVertical_bias="0.0">

</FrameLayout>

<Button
    android:id="@+id/button"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginBottom="22dp"
    android:text="FirstFargment"
    android:textSize="13dp"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.114"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/frameLayout" />

<Button
    android:id="@+id/button2"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginEnd="10dp"
    android:layout_marginBottom="20dp"
    android:text="SecondFargment"
    android:textSize="13dp"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.89"
    app:layout_constraintStart_toEndOf="@+id/button" />

Fragment one xml code
<TextView
    android:id="@+id/textView2"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="FirstFragment"
    android:textSize="48dp"
    android:textColor="#FFFFFF"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent" />

Note Please change the Background of Fragment
android:background="#83E111"

Fragment Second xml code
<TextView
    android:id="@+id/textView"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="SecondFragment"
    android:textSize="48dp"
    android:textColor="#FFFFFF"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent" />
```

Note Please change the Background of Fragment
android:background="#AA7D7D"

Java Code for Main Activity

```
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);
    FirstFragment fragment1=new FirstFragment();
    secondFragment fragment2=new secondFragment();
    FragmentManager managerF = getSupportFragmentManager();
    // FragmentManager is the class responsible for performing actions
    // on your app's fragments,
    // such as adding, removing, or replacing them
    FragmentTransaction Ftransaction = managerF.beginTransaction();
    //a FragmentManager can add, remove, replace, and perform other actions
with fragments
    // in response to user interaction. Each set of fragment changes that
you
    // commit is called a transaction
    Ftransaction.add(R.id.frameLayout,fragment1);
    //add the  Fragment
    Button b1 = findViewById(R.id.button);
    Button b2 = findViewById(R.id.button2);
    b1.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View view) {
            FragmentManager managerF = getSupportFragmentManager();
            FragmentTransaction Ftransaction = managerF.beginTransaction();
            Ftransaction.replace(R.id.frameLayout,fragment1);
            //replace the  Fragment
            Toast.makeText(MainActivity.this,"Changed to
Fragment1",Toast.LENGTH_LONG).show();
            Ftransaction.commit();
            //Commit the changes in the Activity Main.
        }
    });
    b2.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View view) {
            FragmentManager managerF = getSupportFragmentManager();
            FragmentTransaction Ftransaction = managerF.beginTransaction();
            Ftransaction.replace(R.id.frameLayout,fragment2);
            Toast.makeText(MainActivity.this,"Changed to
Fragment1",Toast.LENGTH_LONG).show();
            Ftransaction.commit();
        }
    });



}
```

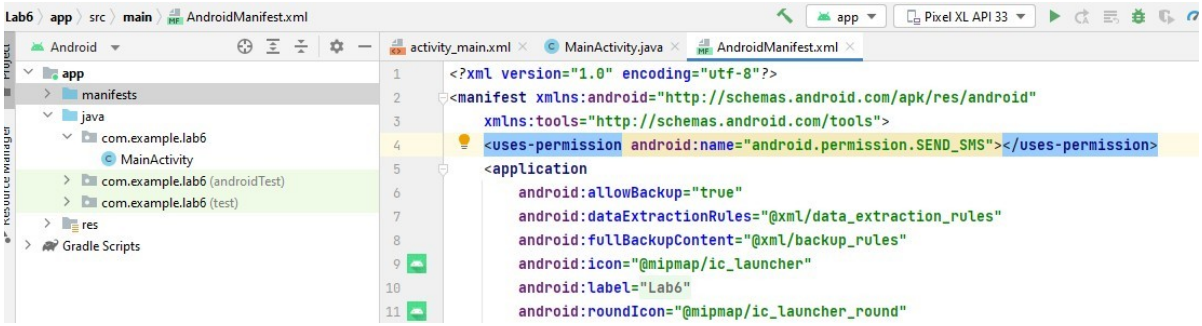
Output



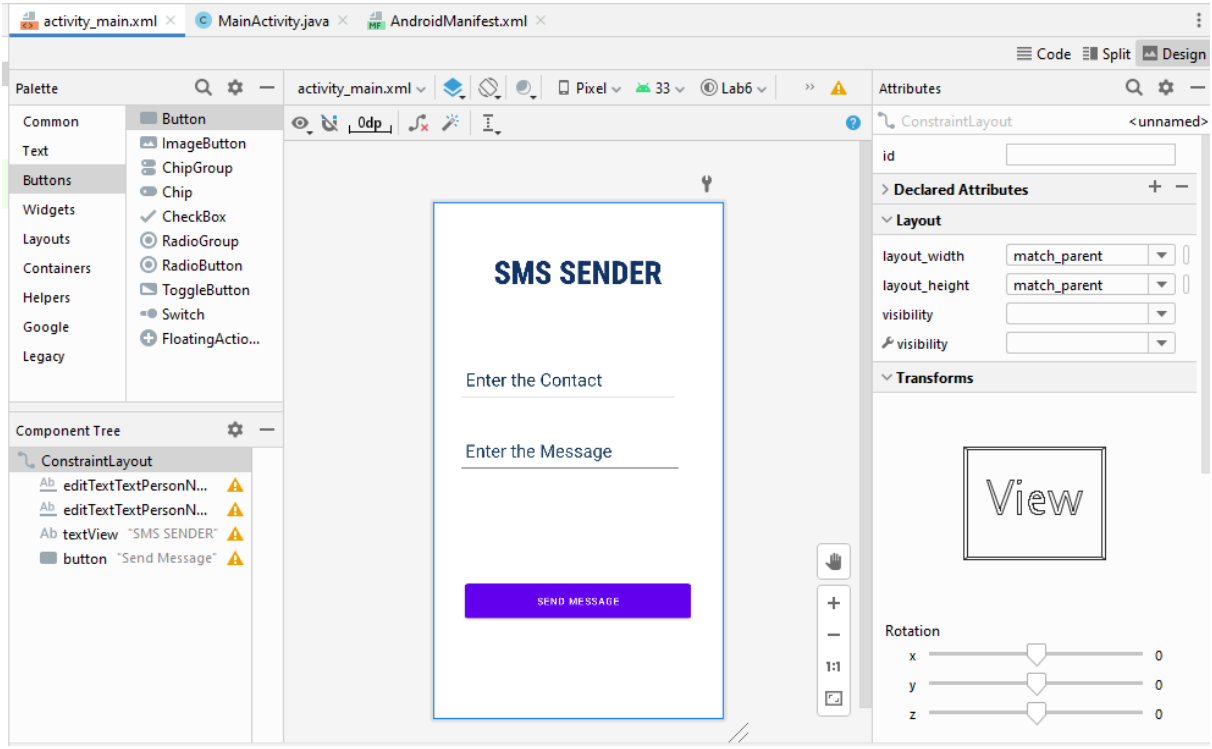
Lab 6

Follow the Step1 and create the Lab6 Project

Step2 : Wait till gradle Build is successful and open the Manifests file android studio add the send message Permission



Step3:Desgin the activity_main.xml for sending the message minimum requirement is to get phone number and message two EditText and one button to send Message.



Activity_main.xml code

```
<EditText
    android:id="@+id/EMessage"
    android:layout_width="316dp"
    android:layout_height="65dp"
    android:layout_marginTop="192dp"
    android:ems="10"
    android:hint="Enter the Message"
    android:inputType="textPersonName"
    android:paddingLeft="10dp"
    android:textColorHint="#234567"
    android:textSize="25dp"
```

```
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.375"
    app:layout_constraintStart_toStartOf="parent"
    />

<EditText
    android:id="@+id/Contact"
    android:layout_width="311dp"
    android:layout_height="65dp"
    android:ems="10"
    android:hint="Enter the Contact"
    android:inputType="textPersonName"
    android:paddingLeft="10dp"
    android:textColorHint="#123654"
    android:textSize="25dp"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.36"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintVertical_bias="0.329" />

<TextView
    android:id="@+id/Title"
    android:layout_width="match_parent"
    android:textAlignment="center"
    android:layout_height="wrap_content"
    android:layout_marginTop="65dp"
    android:layout_marginBottom="91dp"
    android:fontFamily="sans-serif-condensed-medium"
    android:text="SMS SENDER"
    android:textColor="#123465"
    android:textSize="45dp"
    android:textStyle="bold"

    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.261"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintVertical_bias="1.0" />

<Button
    android:id="@+id/Send"
    android:layout_width="match_parent"
    android:layout_margin="45dp"
    android:layout_height="60dp"
    android:text="Send Message"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.448"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintVertical_bias="0.844" />
```

Java Code for Main Activity

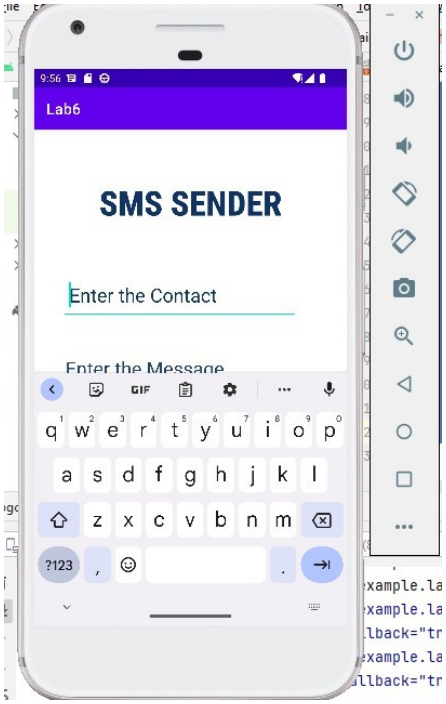
Note Please Add contacts before Running the Program in the AVD

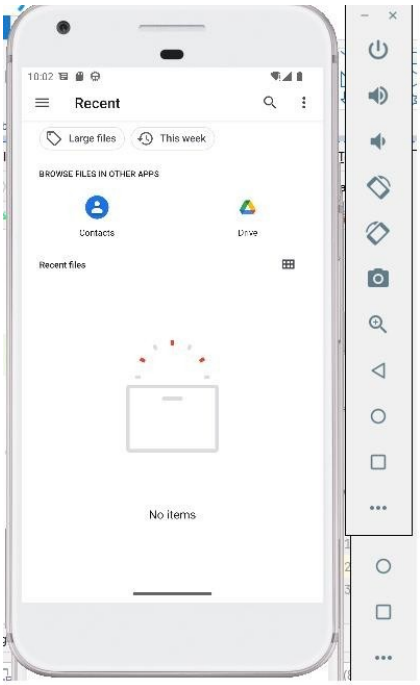
```
EditText Contact ,Emessage;
Button send;
@Override
```

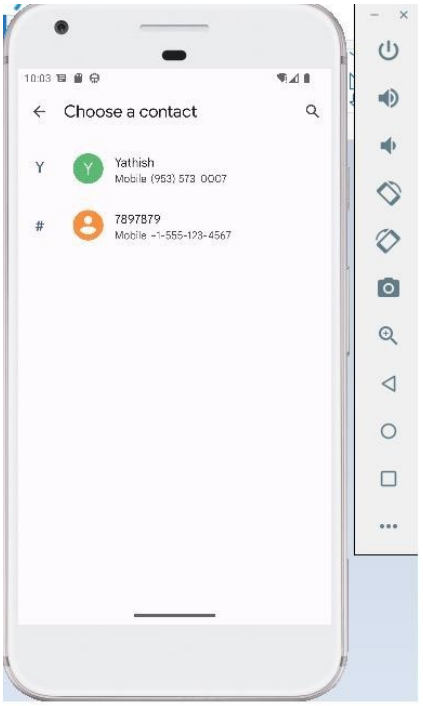
```
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);
    send = findViewById(R.id.Send);
    Contact=findViewById(R.id.Contact);
    Emessage=findViewById(R.id.EMessage);
    Contact.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View view) {
            Intent i = new Intent(Intent.ACTION_GET_CONTENT);


i.setType(ContactsContract.CommonDataKinds.Phone.CONTENT_ITEM_TYPE);
            // this is to Enable the intent to Access the Contacts present
in the Android Contact
            // and send the Contact details to app
            startActivityForResult(i,1);
        }
    });
    send.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View view) {
            try {
                SmsManager smg = SmsManager.getDefault();

        //This is used get the smsManager from the android Device

                smg.sendTextMessage(Contact.getText().toString(), null,
Emessage.getText().toString(), null, null);

//This is line will use  smsManager  to send mail from the app.

                Toast.makeText(MainActivity.this, "Message Send",
Toast.LENGTH_LONG).show();
            }catch (Exception e)
            {       Log.e("Message",e.toString());
                Toast.makeText(MainActivity.this, e.toString(),
Toast.LENGTH_LONG).show();
            }
        }
    });
}

@Override
protected void onActivityResult(int requestCode, int resultCode, @Nullable
Intent data) {
    super.onActivityResult(requestCode, resultCode, data);
    if(requestCode==1)
    {
        if(resultCode==RESULT_OK)
        {
            try{

                Uri contactData = data.getData();
//Accessing the data which is send by the intent Activity
                Cursor = managedQuery(contactData,null,null,null,null);
//Use to traversal the data which is got from activity
                cursor.moveToFirst();
//Accessing the First Data in Got data
                String number="Contact Numbere";
```

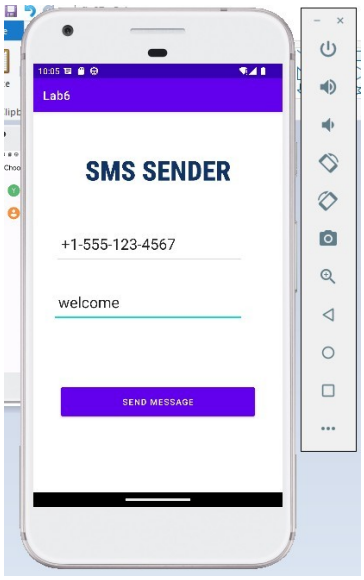
```
                int xz=cursor.getColumnIndex("data1");
//Getting the Index where Phone Number Exists
                number=cursor.getString(xz);
//Getting the phone number from the  Contact  Data;


                Contact.setText(number);

            }
            catch (Exception e )
            {
                Contact.setText(e.toString());
            }
        }
    }
}
```

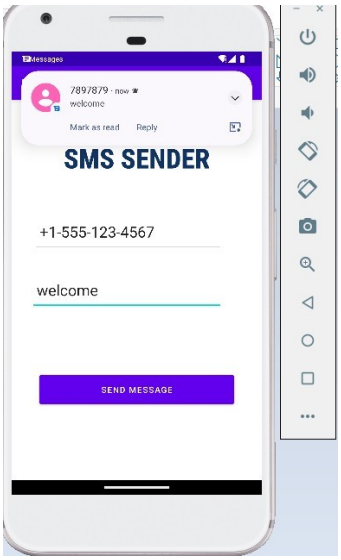
Output







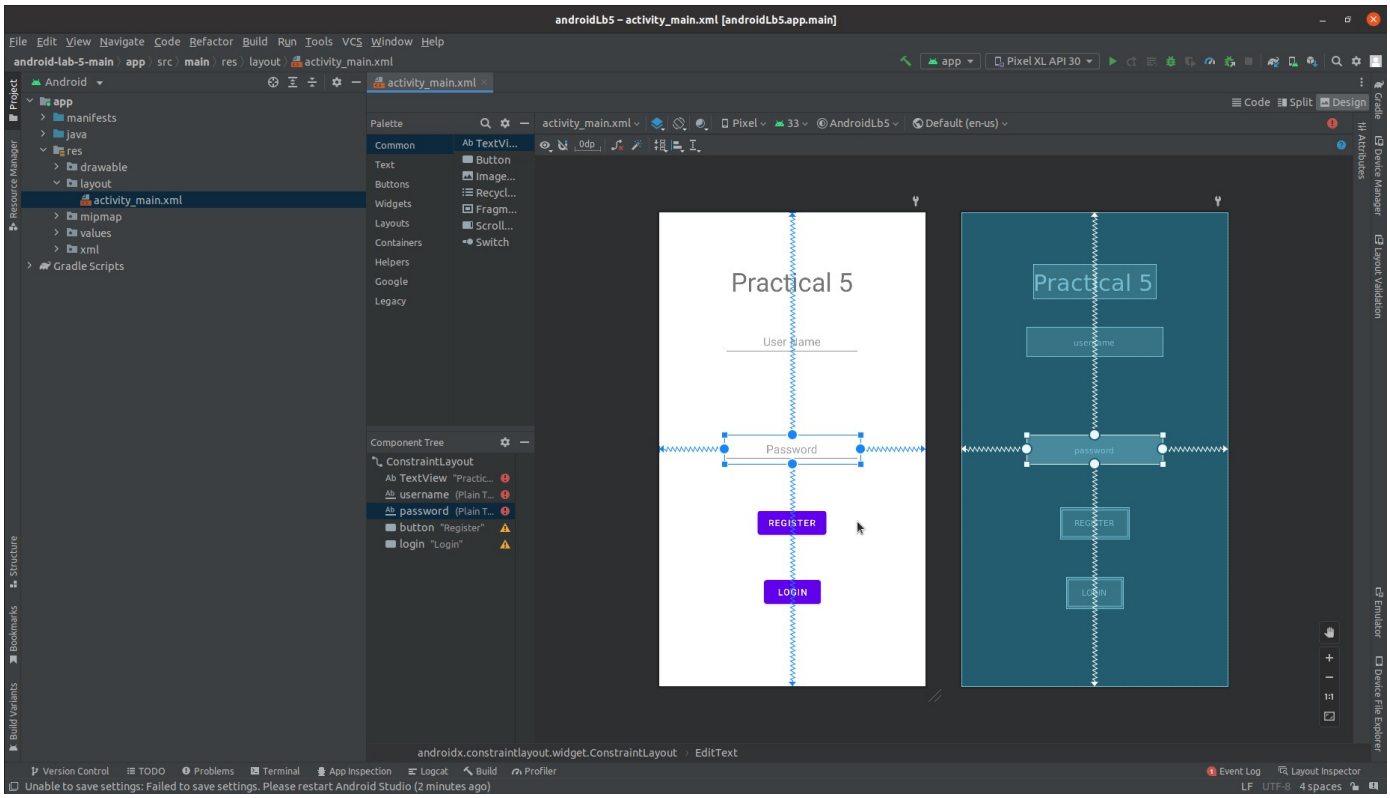

10:05
Lab6
SMS SENDER
+1-555-123-4567
welcome
SEND MESSAGE


Messages
7897879 · now
welcome
Mark as read
Reply
SMS SENDER
+1-555-123-4567
welcome
SEND MESSAGE

LAB 5

Follow the **Step1** and create the Lab6 Project

Step 2:Desgin the activity_main.xml for creating a database with table of user credentials and create a login portal system



Step 3:Activity_main.xml code

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

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:gravity="center"
        android:text="Practical 5"
        android:textSize="40dp"
        tools:layout_editor_absoluteX="111dp"
        tools:layout_editor_absoluteY="80dp" />

    <EditText
        android:id="@+id/username"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:ems="10"
        android:inputType="textPersonName"
        android:hint="User Name"
        android:gravity="center"
```

```
        android:text=""
        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="0.258" />

    <EditText
        android:id="@+id/password"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:ems="10"
        android:hint="Password"
        android:gravity="center"
        android:inputType="textPersonName"
        android:text=""
        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" />

    <Button
        android:id="@+id/button"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginBottom="228dp"
        android:text="Register"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.498"
        app:layout_constraintStart_toStartOf="parent" />

    <Button
        android:id="@+id/login"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Login"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        app:layout_constraintVertical_bias="0.822" />
</androidx.constraintlayout.widget.ConstraintLayout>
```

Step 4: Create a Java class for database connection and query methods.

```
package com.example.androidlb5;

import android.content.ContentValues;
import android.content.Context;
import android.database.Cursor;
import android.database.sqlite.SQLiteDatabase;
import android.database.sqlite.SQLiteOpenHelper;
import android.util.Log;
import android.widget.Toast;

import androidx.annotation.Nullable;
```

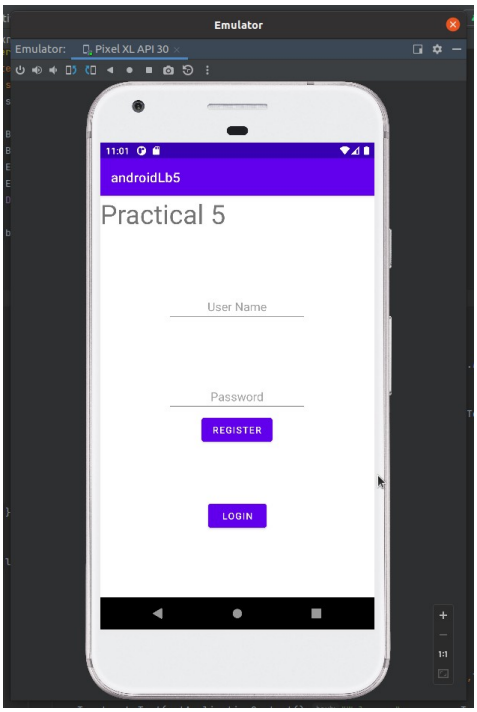
```
public class database extends SQLiteOpenHelper{


    public database( Context context) {
        super(context, "userdb",null,1);
    }

    @Override
    public void onCreate(SQLiteDatabase db) {
        db.execSQL("create table user(username TEXT primary key , password TEXT)");
    }

    @Override
    public void onUpgrade(SQLiteDatabase db, int i, int i1) {

    }


    public boolean insertData(String username , String password){
     SQLiteDatabase db=this.getWritableDatabase();
        ContentValues contentValues=new ContentValues();
        contentValues.put("username",username);
        contentValues.put("password",password);
        long result=db.insert("user",null,contentValues);

        if(result==-1){
            return false;
        }else{
            return true;
        }


    }


    public boolean getData(String username,String password) {

        SQLiteDatabase db = this.getReadableDatabase();
        Cursor cr = db.rawQuery("select * from user where username='" + username + "' and
password=='"+password+"'", null);
        if (cr.getCount() == 0) {
            return false;
        } else {
            return true;
        }

    }
}
```

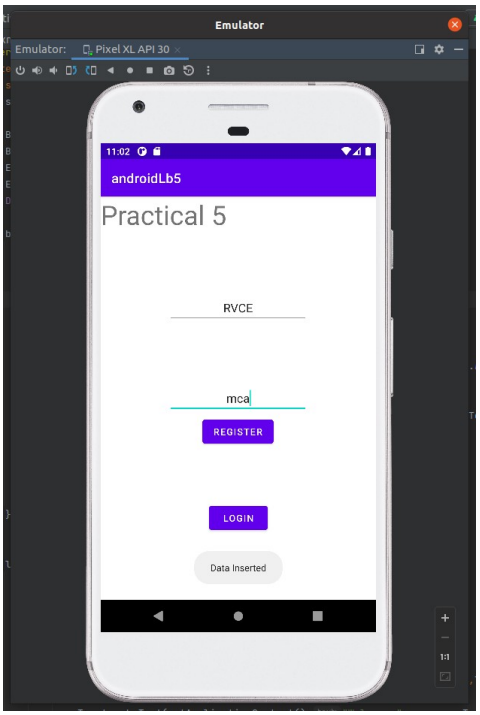
Step 5: Java Code for Main Activity

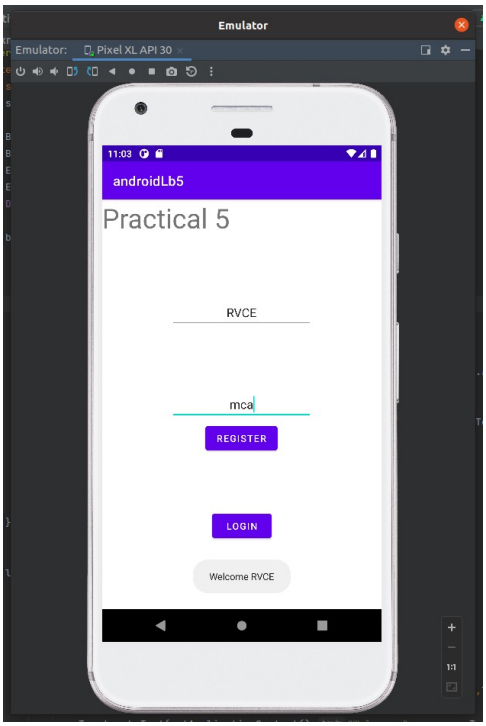
```
package com.example.androidlb5;


import androidx.appcompat.app.AppCompatActivity;

import android.database.Cursor;
import android.os.Bundle;
import android.util.Log;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
```

```
import android.widget.Toast;

public class MainActivity extends AppCompatActivity {

    String username,password;
    database DB;

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

        Button btn=findViewById(R.id.button);
        Button login=findViewById(R.id.login);
        EditText usernameText=findViewById(R.id.username);
        EditText passwordText=findViewById(R.id.password);
        DB=new database(this);

        btn.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                username=usernameText.getText().toString();
                password=passwordText.getText().toString();

                boolean result=DB.insertData(username,password);
                if(result==true){
                    Toast.makeText(getApplicationContext(),"Data Inserted",Toast.LENGTH_LONG).show();
                    Log.i("result","Data inserted");
                }else{
                    Toast.makeText(getApplicationContext(),"Cannot Insert
data",Toast.LENGTH_LONG).show();
                    Log.i("result","failed");
                }


            }
        });


        login.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                username=usernameText.getText().toString();
                password=passwordText.getText().toString();
                Boolean cr=DB.getData(username,password);
                if(cr==false){
                    Toast.makeText(getApplicationContext(),"Invalid
credentials",Toast.LENGTH_LONG).show();
                }else{
                    Toast.makeText(getApplicationContext(),"Welcome
"+username,Toast.LENGTH_LONG).show();

                }
            }
        });

    }
}
```

Output:


Emulator
Emulator: Pixel XL API 30
androidLb5
Practical 5
User Name
Password
REGISTER
LOGIN


Emulator
Emulator: Pixel XL API 30
androidLb5
Practical 5
RVCE
mca
REGISTER
LOGIN
Data Inserted


Emulator
Emulator: Pixel XL API 30
androidLb5
Practical 5
RVCE
mca
REGISTER
LOGIN
Welcome RVCE