Lab 1

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

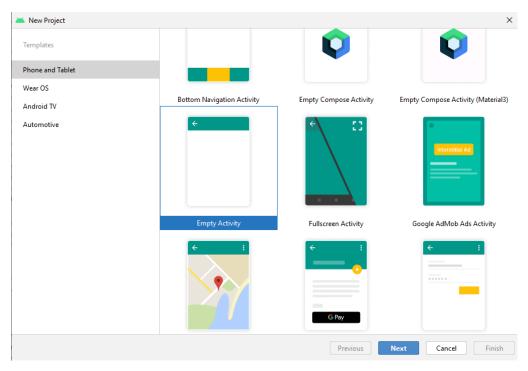


Figure 1 Selecting the Empty Activity

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

Note: Make sure you have an active internet connection.

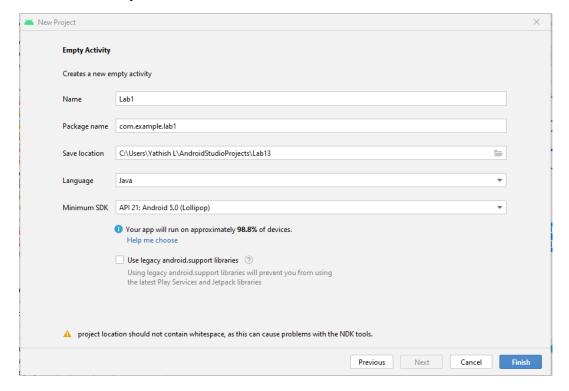
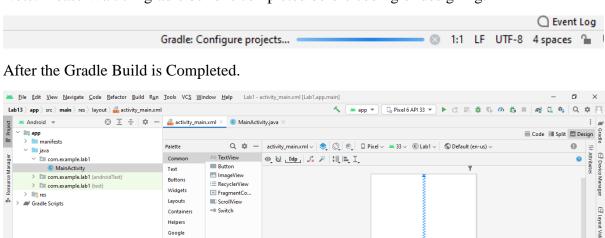


Figure 2 Name the project and select the Programming language.

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



-

1:1

☐ Event Log ☐ Layout Inspector

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

Legacy

Gradle sync finished in 44 s 82 ms (2 minutes ago)

ConstraintLayout

Ab TextView "Hello World

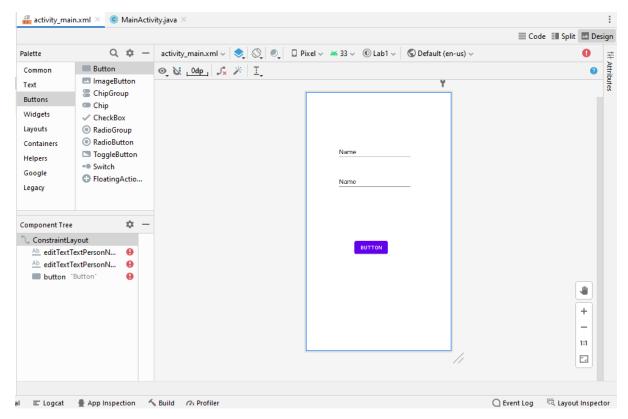
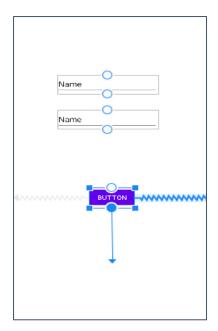
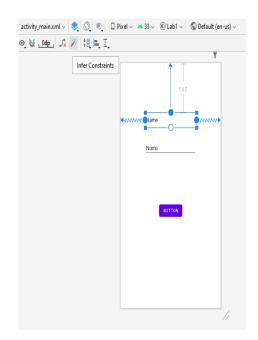


Figure There is No constraint assigned to Edit text or Button, you can do this is in two Ways one using the mapping the Constraint manually or using Infer Constraint

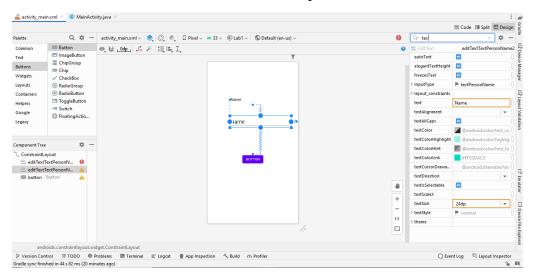




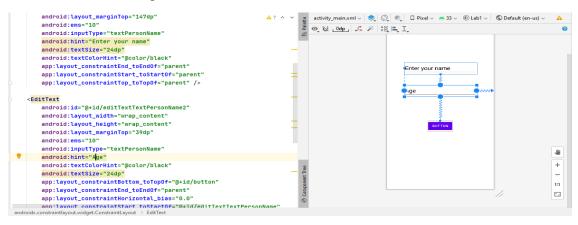
Manual Assigning

Infer Constraint

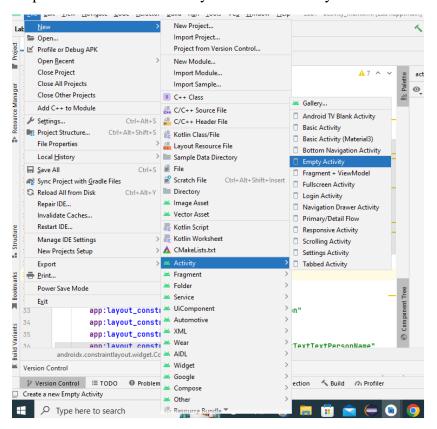
To change the Attributes of a component you can use the Attribute panel or in xml code and use can search the Attributes in the Attributes panel



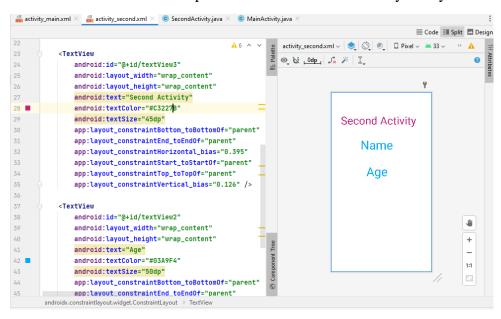
You can also do Change Attributes in the xml File.



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"</pre>
```

```
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);

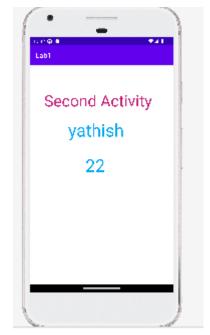
```
//The startActivity(Intent) method is used to start a new
activity,

// 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
```

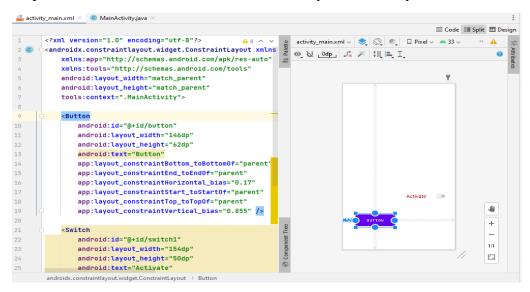




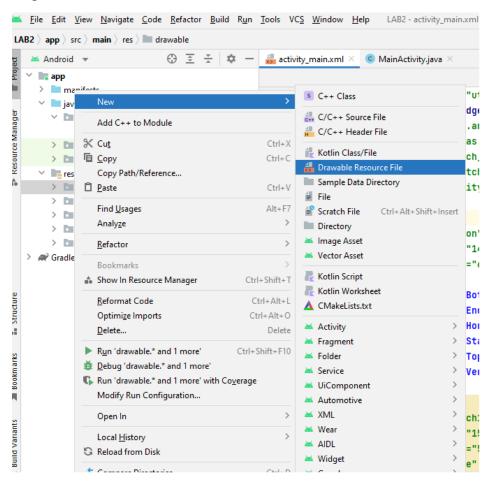
Lab 2

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

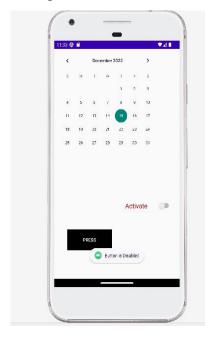


Write the code in the Drawable file(shape.xml)

```
<?xml version="1.0" encoding="utf-8"?>
<shape xmlns:android="http://schemas.android.com/apk/res/android"</pre>
android:shape="rectangle">
 <gradient android:startColor="#123456" android:type="linear"</pre>
android:endColor="#456875"></gradient>
    <corners android:radius="18dp"></corners>
</shape>
//This code is for design the graphics of the button
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"</pre>
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</pre>
    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="ldp"
    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





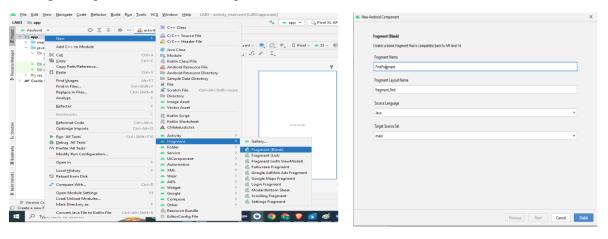


Lab 3

Follow the Step1 of Lab1 Program

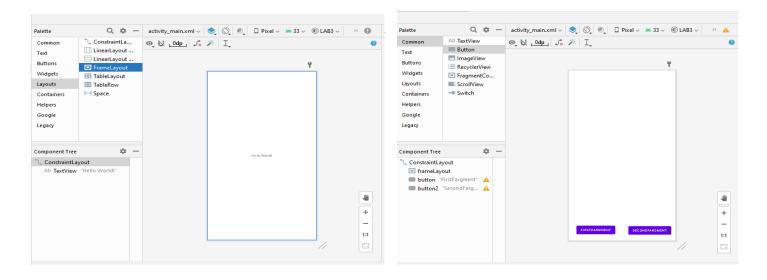
Step2 Create the two Fragment in the project.

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"</pre>
```

```
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" />
```

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
VOU
    // 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

