LAB PRACTICALS

Practical 1(A)

Aim : Write a program using kotlin to implement control structures and loops.

☑ *Download IntelliJ IDEA and perform the following Practicals.*

(A) control statements

Kotlin flow control statements determine the next statement to be executed. For example, the statements if-else, if, when, while, for, and do are flow control statements.

(1) if condition: The If statement allows you to specify a section of code that is executed only if a given condition is true.

```
fun main() {
var n = 34
if(n % 2 == 0) {
        println("$n is even")
}
}
```

Output for above program:

```
        34 is even
```

(2) if else condition:if statement is executed and the given condition is checked. If this condition is evaluated to be true then the first code block is executed, otherwise the program goes into the else part and another code block is executed.

```
fun main() {
   val age:Int = 10

   if (age > 18) {
      print("Adult")
   } else {
      print("Minor")
   }
}
```

Output for above program is:

```
Minor
```

(3) if else if condition: else if condition is used to specify a new condition if the first condition is false.

```
fun main() {
   val age:Int = 13
   val result = if (age > 19) {
      "Adult"
   } else if ( age > 12 && age  < 20 ){
      "Teen"
   } else {
      "Minor"
   }
   print("The value of result : ")
   println(result)
}
```

Output for above program is:

```
The value of result : Teen
```

(4) When condition: when expression is similar to the switch statement in C, C++ and Java. when expression matches the supplied argument with all the branches one by one until a match is found. Once a match is found, it executes the matched branch. If none of the branches match, the else branch is executed.

```
fun main() {
  val day = 2
  val result = when (day) {
    1 -> "Monday"
    2 -> "Tuesday"
    3 -> "Wednesday"
    4 -> "Thursday"
    5 -> "Friday"
    6 -> "Saturday"
    7 -> "Sunday"
    else -> "Invalid day."
  }
  println(result)
}
```

Output for above program is:

```
Tuesday
```

(B) Loops

Loops are used in programming to repeat a specific block of code until a certain condition is met.

1) **For Loop:**for loop iterates through anything that provides an iterator ie. that contains a countable number of values, for example arrays, ranges, maps or any other collection available in Kotlin.

```
fun main() {
  for (item in 1..5) {
    println(item)
  }
}
```

Output for above program is:

```
1
2
3
4
5
```

2) **while loop:**while loop executes its body continuously as long as the specified condition is true.

Example:

```
{
  var i = 5;
  while (i > 0) {
    println(i)
    i--
  }
  }
```

Output for above program is:

```
5
4
3
2
1
```

(3) Break and continue statements: The break and continue statements are jump statements that can be used to omit some statements within a loop or end a loop instantly without first checking the test expression.

Break statement:

Kotlin break statement is used to come out of a loop once a certain condition is met. This loop could be a for, while or do...while loop.

```
fun main() {
  var i = 0;
  while (i++ < 100) {
    println(i)
    if( i == 3 ){
      break
    }

  }
}
```

Output for above program is:

```
1
2
3
```

Continue statement

The Kotlin continue statement breaks the loop iteration in between(skips the part next to the continue statement till end of the loop) and continues with the next iteration in the loop.

```
fun main() {
  var i = 0;
  while (i++ < 6) {
    if( i == 3 ){
      continue
    }
    println(i)
  }
}
```

Output for above program is:

```
1
2
4
5
6
```

Practical 1(B)

Aim : Write a program to implement object-oriented concepts in kotlin.

☑ ***Kotlin supports both functional and object-oriented programming. Object oriented programming (OOP) allows us to solve the complex problem by using objects.***

(1) Example: Kotlin Class and Object

```
class Lamp {
    // property (data member)
    private var isOn: Boolean = false

    // member function
    fun turnOn() {
        isOn = true
    }
    // member function
    fun turnOff() {
        isOn = false
    }
    fun displayLightStatus(lamp: String) {
        if (isOn == true)
            println("$lamp lamp is on.")
        else
            println("$lamp lamp is off.")
    }
}
fun main() {
    val l1 = Lamp() // create l1 object of Lamp class
    val l2 = Lamp() // create l2 object of Lamp class
    l1.turnOn()
    l2.turnOff()
    l1.displayLightStatus("l1")
    l2.displayLightStatus("l2")
}
```

Output for above program is

```
l1 lamp is on.
l2 lamp is off.
```

(2) Example: Kotlin Inheritance

```
fun main() {
    var student_1 = Student("Arjun")
    var teacher_1 = Teacher("Amit")

    println("\n\nAbout "+student_1.name+"\n----------")
    student_1.doAll()

    println("\n\nAbout "+teacher_1.name+"\n----------")
    teacher_1.doAll()
}

/**
 * Person is a Parent Class
 */
open class Person(var role: String = "Person", var name: String = "X") {
    fun eat(){
        println(name + " is eating.")
    }
    fun sleep(){
        println(name + " is sleeping.")
    }
}

/**
 * Student class inherits Person class
 */
class Student(name: String): Person("Student", name) {
    // activity function belongs to Student only
    fun activity(){
        println("$name is a $role. $name is studying in school.")
    }

    fun doAll(){
        eat()
        sleep()
```

```
        activity()
    }
}

/**
 * Student class inherits Person class
 */
class Teacher(name: String): Person("Teacher", name) {
    fun profession(){
        println("$name is a $role. $name teaches at school.")
    }

    fun doAll(){
        eat()
        sleep()
        profession()
    }
}
```

Output for above program is:

```
About Arjun
---------------
Arjun is eating.
Arjun is sleeping.
Arjun is a Student. Arjun is studying in school.

About Amit
---------------
Amit is eating.
Amit is sleeping.
Amit is a Teacher. Amit teaches at school.
```

(3) Constructor

☑ ***A constructor is a concise way to initialize class properties. A constructor is like a special function, and it is defined by using two parentheses () after the class name.***

Example

```
fun main() {
    val person1 = Person("Sumit", 26)
    println("First Name = ${person1.firstName}")
    println("Age = ${person1.age}")
}
class Person(val firstName: String, var age: Int) {
}
```

Output for above program is:

```
First Name = Sumit
Age = 26
```

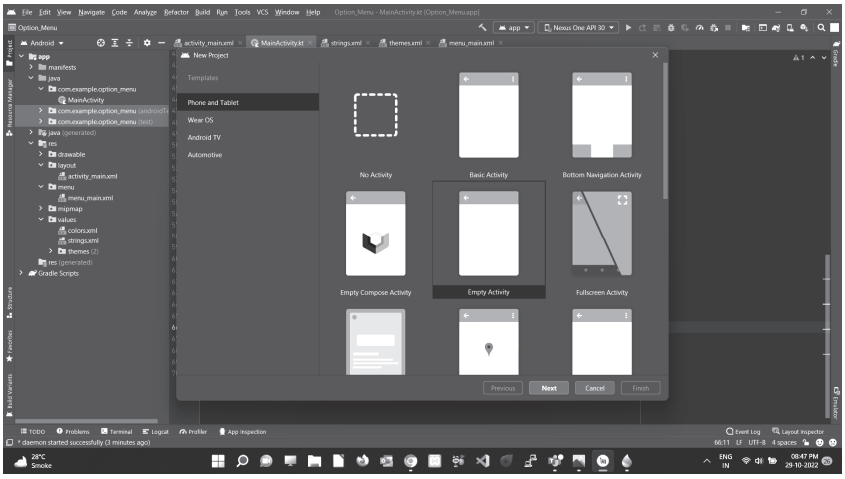
Practical 2(A)

Aim : Create an android application to design screens using different layouts and UI including Button, Edittext,Textview,Radio Button etc.

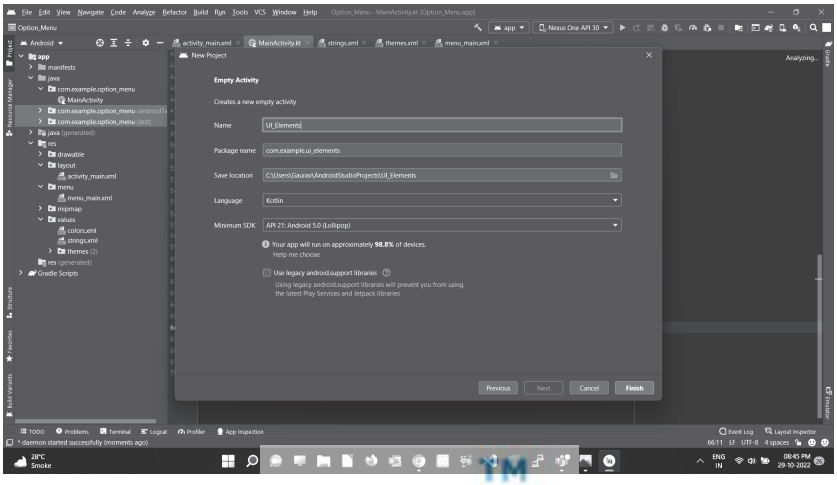
☑ ***To create the graphical user interface for any app, android offers a selection of pre-built UI elements, including UI controllers and structured layout objects. However, Android offers additional UI modules for various interfaces like dialogue, notifications, and menus.***

Install Android studio and follow the steps given below.

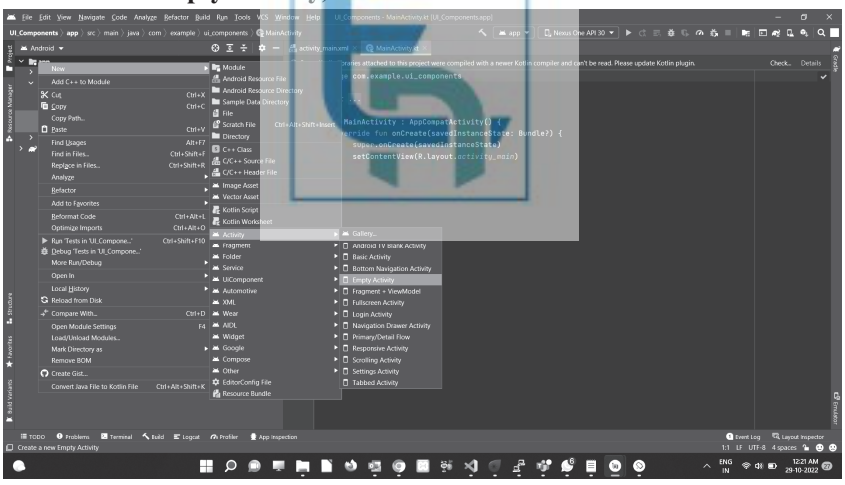
▶ **Step 1 : Go to file - New - New Project - select empty activity- Next**



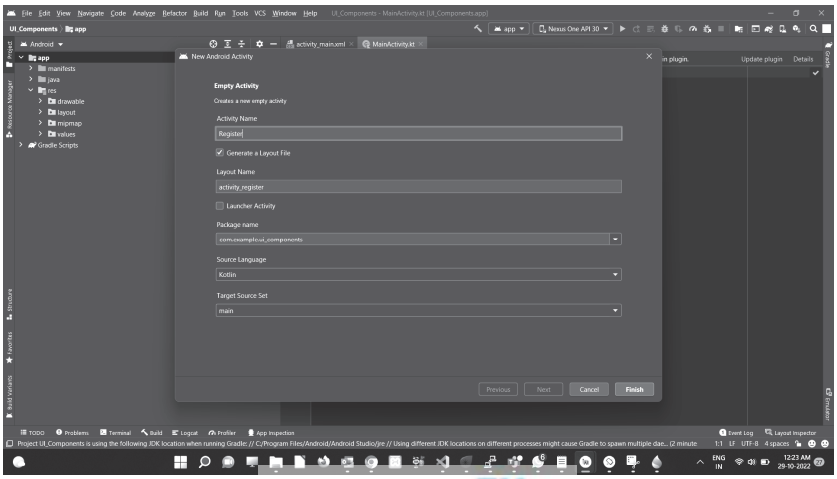
☞ **Set Name for your project as UI_components and click on finish**



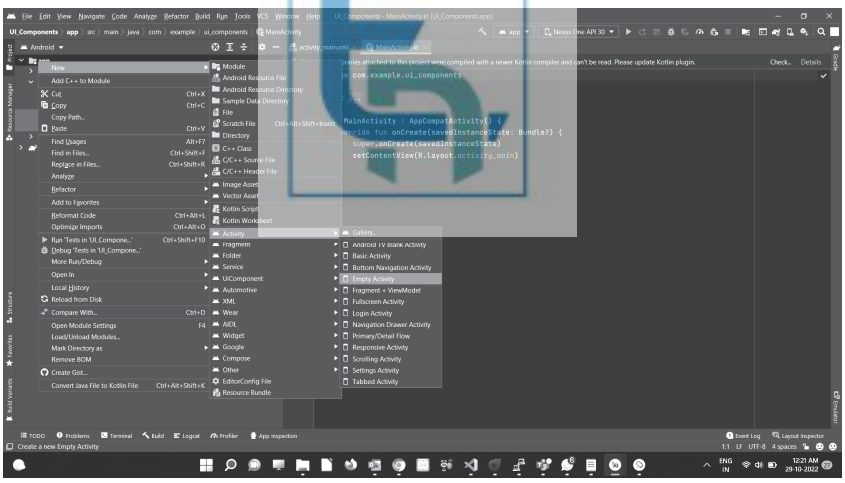
▸ **Step 2 : Create Register activity(Right click on app-New-Activity-Empty Activity)**



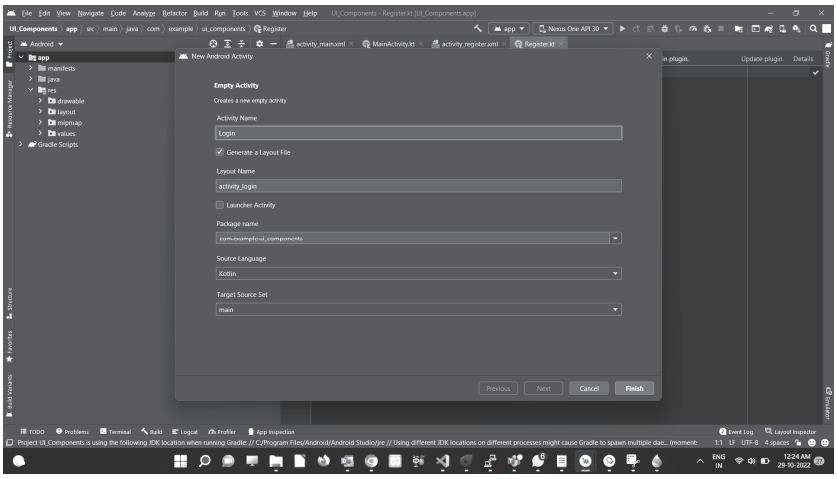
☞ **Set activity name as Register and click on finish.**



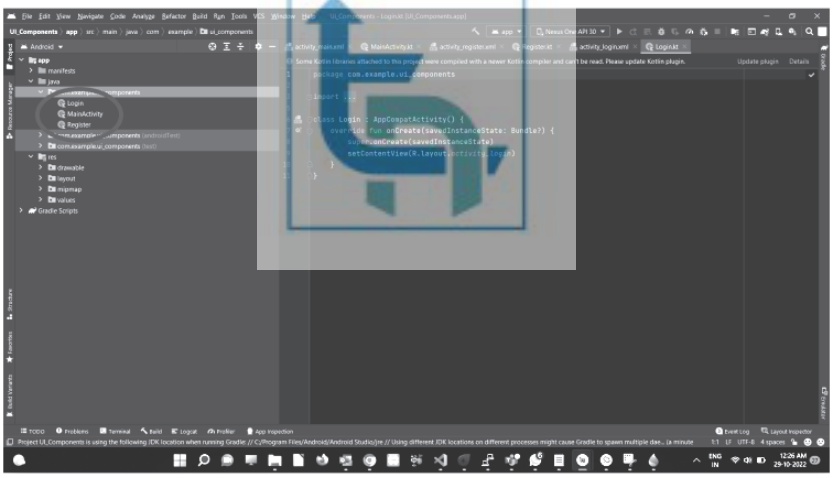
- **Step 3 : Create Login activity(Right click on app-New-Activity-Empty Activity**



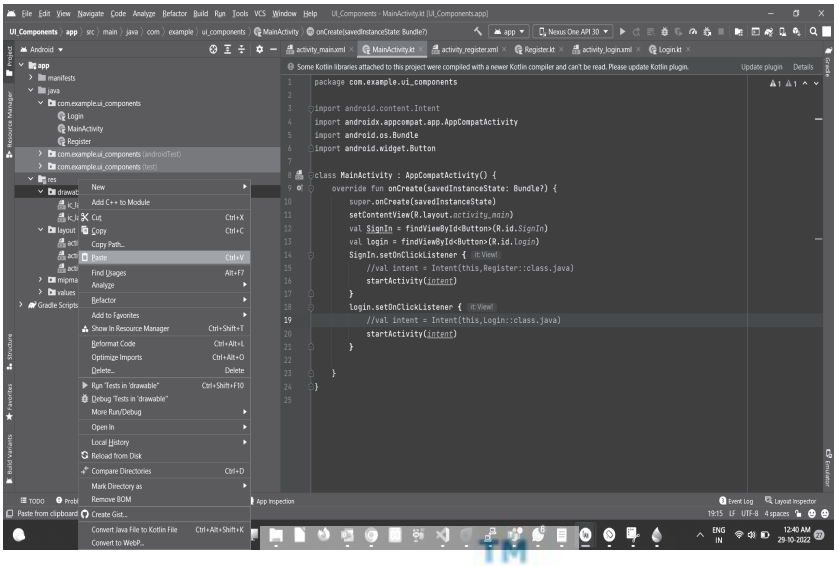
☞ **Set the activity name as Login and click on finish.**

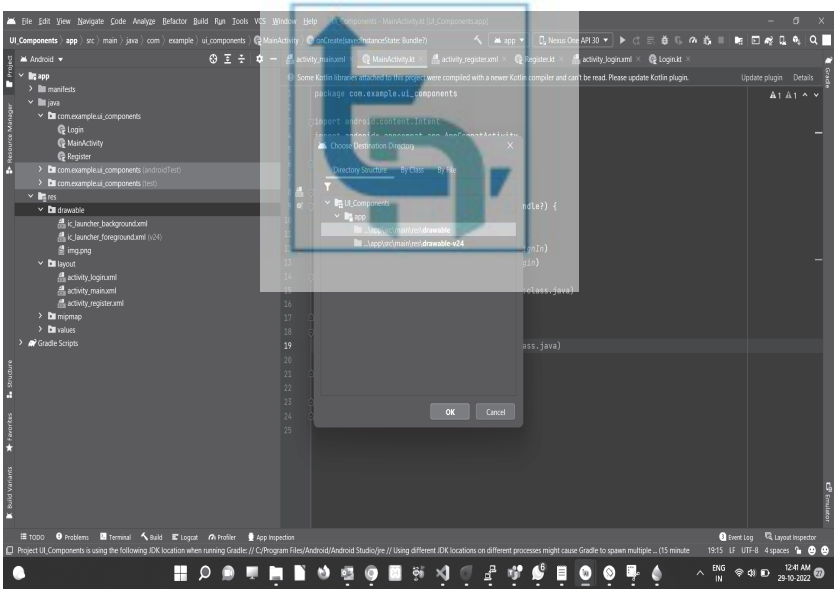


☞ **After creating Register and Login activities GUI will be:**

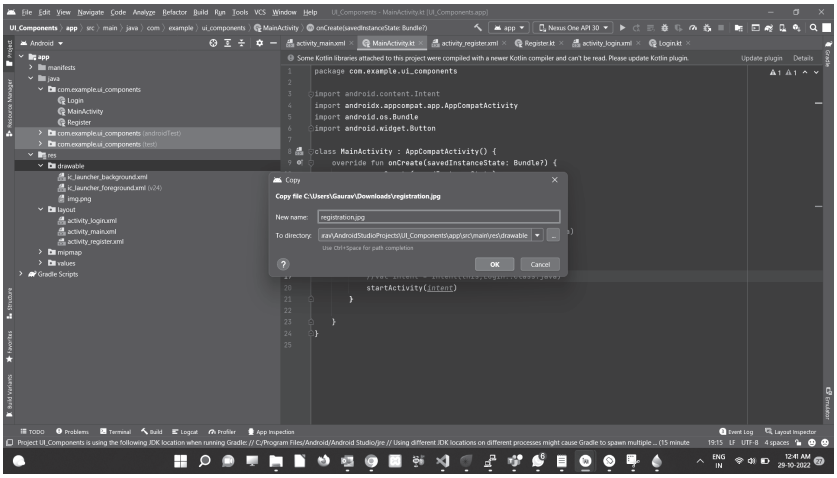


☞ **Now, Copy 3 images from the download folder which you are going to use in your application(images for background) and paste inside the drawable as shown below.**





Click on OK.



Click on OK.

☞ **activity_main.xml**

```
<?xml version="1.0" encoding="utf-8"?>
<AbsoluteLayout
xmlns:android="http://schemas.android.com/apk/res/android"
xmlns:tools="http://schemas.android.com/tools"
xmlns:app="http://schemas.android.com/apk/res-auto"
android:layout_width="match_parent"
android:layout_height="match_parent"
tools:context=".MainActivity"
android:background="@drawable/signin"
android:orientation="vertical">
<TextView
android:text="Welcome Page"
android:layout_width="wrap_content"
android:layout_height="150dp"
android:id="@+id/textView"
android:textColor="#000"
android:layout_x="120dp"
android:layout_y="200dp"
android:textSize="25sp" />

<Button android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:text="Login"
```

```
android:id="@+id/login"
android:background="@color/design_default_color_primary"
android:textColor="@color/design_default_color_on_secondary"
android:textSize="15sp"
android:layout_x="80dp"
android:layout_y="300dp"
android:textColorLink="@android:color/darker_gray"/>

<Button android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:id="@+id/SignIn"
android:background="@color/design_default_color_primary"
android:textColor="@color/design_default_color_on_secondary"
android:textColorLink="@android:color/darker_gray"
android:text="Sign In"
android:textSize="15sp"
android:layout_x="200dp"
android:layout_y="300dp"/>
</AbsoluteLayout>
```

☞ **activity_login.xml**

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout
xmlns:android="http://schemas.android.com/apk/res/android"
xmlns:tools="http://schemas.android.com/tools"
xmlns:app="http://schemas.android.com/apk/res-auto"
android:layout_width="match_parent"
android:layout_height="match_parent"
tools:context=".Login"
android:background="@drawable/img">
<TextView android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:text="Welcome to Login Screen!!"
android:layout_above="@id/edittxt2"
android:layout_marginBottom="50dp"
android:layout_marginLeft="30dp"
android:textColor="@color/design_default_color_primary"
android:textSize="30dp"/>
<EditText android:layout_width="wrap_content"
android:layout_height="wrap_content"
```

```
android:hint="Username"
android:id="@+id/edittxt2"
android:layout_above="@id/edittxtPass"
android:layout_marginLeft="70dp"
android:ems="8"
android:textColorHint="@color/design_default_color_on_secondary"/>

<EditText android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:hint="Password"

android:id="@+id/edittxtPass"
android:layout_above="@id/btnlogin"
android:layout_marginLeft="70dp"
android:ems="8"
android:textColorHint="@color/design_default_color_on_secondary"/>
<Button android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:id="@+id/btnlogin"
android:text="Submit"
android:background="@color/design_default_color_on_secondary"
android:layout_alignTop="@id/cancel1"
android:layout_marginLeft="40dp"/>

<Button
android:id="@+id/cancel1"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_centerInParent="true"
android:background="@color/design_default_color_on_secondary"
android:text="Cancel" />
</RelativeLayout>
```

☞ **activity_register.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"
xmlns:app="http://schemas.android.com/apk/res-auto"
android:layout_width="match_parent"
```

```
android:layout_height="match_parent"
tools:context=".Register"
android:background="@drawable/registration"
android:orientation="vertical"
android:layout_marginLeft="20dp">
<TextView
android:text="Welcome to Registration Form"
android:layout_width="match_parent"
android:layout_height="wrap_content" android:id="@+id/textView5"
android:layout_centerHorizontal="true"
android:textSize="20sp"
android:textColor="@color/design_default_color_primary" />
<TextView
android:text="Enter your Name"
android:layout_width="wrap_content"
android:layout_height="wrap_content" android:id="@+id/textView4"/>
<EditText
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:inputType="textPersonName"
android:ems="10"
android:id="@+id/editText3"/>
<TextView
android:id="@+id/textView3"
android:text="Enter your Phone No"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
/>
<EditText
android:id="@+id/editText2"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:inputType="textPersonName"
android:ems="10"
/>
<TextView
android:text="Enter Your Address"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:id="@+id/textView2"/>
```

```
<EditText
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:inputType="textPersonName"
android:ems="10"
android:id="@+id/editText"/>
<RadioGroup
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:orientation="horizontal"
android:id="@+id/radioGroup">
<RadioButton
android:text="Male"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:id="@+id/radioMale"
android:layout_weight="1"/>
<RadioButton
android:text="Female"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:id="@+id/radioFemale"
android:layout_weight="1"/>
</RadioGroup>
<CheckBox
android:text="Reading"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:id="@+id/checkRead"/>
<CheckBox
android:text="Singing"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:id="@+id/checkSing"
/>
<CheckBox
android:text="Dancing"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:id="@+id/checkDance"
```

```
style="?android:attr/starStyle"/>
<LinearLayout android:layout_width="match_parent"
android:layout_height="match_parent"
android:orientation="horizontal">
<Button
android:text="Submit"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:id="@+id/btnSubmit"/>
<Button
android:text="Cancel"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:id="@+id/btnCancel"
android:layout_marginLeft="40dp"/>
</LinearLayout>
</LinearLayout>
```

☞ **MainActivity.kt**

```
package com.example.ui_components
import android.content.Intent
import androidx.appcompat.app.AppCompatActivity
import android.os.Bundle
class MainActivity : AppCompatActivity() {
override fun onCreate(savedInstanceState: Bundle?) {
super.onCreate(savedInstanceState)
      setContentView(R.layout.activity_main)
val SignIn = findViewById<Button>(R.id.SignIn)
val login= findViewById<Button>(R.id.login)

SignIn.setOnClickListener{
val i = Intent(applicationContext, Register::class.java)
  startActivity(i)
}
 login.setOnClickListener{
val i = Intent(applicationContext, Login::class.java)
    startActivity(i)
 }
  }
}
```

☞ **Register.kt**

```
package com.example.ui_components
import android.content.Intent
import androidx.appcompat.app.AppCompatActivity
import android.os.Bundle
import android.widget.Button

class Register : AppCompatActivity() {
override fun onCreate(savedInstanceState: Bundle?) {
super.onCreate(savedInstanceState)
    setContentView(R.layout.register)
val btnCancel = findViewById<Button>(R.id.btnCancel)
    btnCancel.setOnClickListener{
val i = Intent(applicationContext, MainActivity::class.java)
      startActivity(i)
}
}
}
```

☞ **Login.kt**

```
package com.example.ui_components
import android.content.Intent
import androidx.appcompat.app.AppCompatActivity
import android.os.Bundle
import android.widget.Button

class Login : AppCompatActivity() {
override fun onCreate(savedInstanceState: Bundle?) {
super.onCreate(savedInstanceState)
    setContentView(R.layout.activity_login)
val cancel1 = findViewById<Button>(R.id.cancel1)
    cancel1.setOnClickListener{
val i = Intent(applicationContext, MainActivity::class.java)
      startActivity(i)
}
}
}
```


1:16
UI_Components
Welcome Page
LOGIN
SIGN IN
1:20
UI_Components
Welcome to Login Screen!!
Username
Password
SUBMI CANCEL
1:21
UI_Components
Welcome to Registration Form
Enter your Name
Enter your Phone No
Enter Your Address
Male Female
Reading
Singing
Dancing
SUBMIT
CANCEL

Practical 2(B)

Aim : Write an android application demonstrating response to event/user interaction for

- **Checkbox**
- **Radio button**
- **Button**
- **Spinner**

(1) activity_main.xml

Note : Copy background image for your registration page and paste inside drawable folder.

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
xmlns:android="http://schemas.android.com/apk/res/a
ndroid"
xmlns:tools="http://schemas.android.com/tools"
xmlns:app="http://schemas.android.com/apk/res-auto"
android:layout_width="match_parent"
android:layout_height="match_parent"
android:background="@drawable/registration"

android:orientation="vertical"
android:layout_marginLeft="20dp">
<TextView
android:text="Welcome to Registration Form"
android:layout_width="match_parent"
android:layout_height="wrap_content"
android:id="@+id/textView5"
android:layout_centerHorizontal="true"
android:textSize="20sp"
android:textColor="@color/design_default_color_prim
ary" />

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

```
android:inputType="textPersonName"
android:ems="10"
android:id="@+id/editText3"
android:hint="Enter Your Name"
/>

<EditText
android:id="@+id/editText2"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:inputType="textPersonName"
android:ems="10"
android:hint="Enter your Phone No"
/>

<EditText
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:inputType="textPersonName"
android:ems="10"
android:id="@+id/editText"
android:hint="Enter Your Address"/>
<RadioGroup
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:orientation="horizontal"
android:id="@+id/radioGroup">
<RadioButton
android:text="Male"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:id="@+id/radioMale"
android:layout_weight="1"/>
<RadioButton
android:text="Female"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:id="@+id/radioFemale"
android:layout_weight="1"/>
</RadioGroup>
<LinearLayout
```

```
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:orientation="horizontal">
<TextView
android:id="@+id/txtView"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:text="Select language:"
android:textSize = "20dp" />

<Spinner
android:id="@+id/spinner"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
/>
</LinearLayout>
<LinearLayout
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:orientation="horizontal">
<CheckBox
android:text="Reading"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:id="@+id/checkRead"/>
<CheckBox
android:text="Singing"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:id="@+id/checkSing"
/>
<CheckBox
android:text="Dancing"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:id="@+id/checkDance"
style="?android:attr/starStyle"/>
</LinearLayout>
<LinearLayout android:layout_width="match_parent"
android:layout_height="match_parent"
android:orientation="horizontal">
```

```
<Button
android:text="Submit"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:id="@+id/btnSubmit"/>
<Button
android:text="Cancel"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:id="@+id/btnCancel"
android:layout_marginLeft="40dp"/>
</LinearLayout>
</LinearLayout>
```

(2) String.xml

```
<resources>
<string name="app_name">Registration_form</string>
<string name="selected_item">Selected item:</string>
<string-array name="Languages">
<item>Java</item>
<item>Kotlin</item>
<item>Swift</item>
<item>Python</item>
<item>Scala</item>
<item>Perl</item>
</string-array>
</resources>
```

(3) MainActivity.kt

```
package com.example.option_menu
import android.os.Bundle
import android.view.View
import android.widget.*
import androidx.appcompat.app.AppCompatActivity
class MainActivity : AppCompatActivity() {
override fun onCreate(savedInstanceState: Bundle?) {
super.onCreate(savedInstanceState)
    setContentView(R.layout.activity_main)
val btnSubmit = findViewById<Button>(R.id.btnSubmit)
val btnCancel = findViewById<Button>(R.id.btnCancel)
val checkRead = findViewById<CheckBox>(R.id.checkRead)
```

```
val checkSing = findViewById<CheckBox>(R.id.checkSing)
val checkDance = findViewById<CheckBox>(R.id.checkDance)
val radioGroup = findViewById<RadioGroup>(R.id.radioGroup)
val languages = resources.getStringArray(R.array.Languages)
// access the spinner
val spinner = findViewById<Spinner>(R.id.spinner)
if (spinner != null) {
val adapter = ArrayAdapter(this,
android.R.layout.simple_spinner_item, languages)
        spinner.adapter = adapter
        spinner.onItemSelectedListener = object :
           AdapterView.OnItemSelectedListener {
override fun onItemSelected(parent: AdapterView<*>,
view: View, position: Int, id: Long) {
              Toast.makeText(this@MainActivity,
getString(R.string.selected_item) + " " +
"" + languages[position], Toast.LENGTH_SHORT).show()
           }
override fun onNothingSelected(parent: AdapterView<*>) {
// write code to perform some action
}
        }
     }
     btnSubmit.setOnClickListener {
        Toast.makeText(this,"Submit button is clicked!!",
Toast.LENGTH_SHORT).show()
     }
     btnCancel.setOnClickListener {
        Toast.makeText(this,"Cancel button is clicked!!",
Toast.LENGTH_SHORT).show()
     }
     radioGroup.setOnCheckedChangeListener { group, checkedId->
var text = "You selected:"
text += if (R.id.radioMale == checkedId) "male" else "female"
Toast.makeText(applicationContext,text,Toast.LENGTH_SHORT).show()
     }
     checkRead.setOnClickListener(View.OnClickListener {
if (checkRead.isChecked) {
           Toast.makeText(this, "Reading Selected!!",
Toast.LENGTH_SHORT).show()
```

```
        }
    })
    checkSing.setOnClickListener(View.OnClickListener {
if (checkSing.isChecked) {
        Toast.makeText(this, "Singing Selected!!",
Toast.LENGTH_SHORT).show()
        }
    })
    checkDance.setOnClickListener(View.OnClickListener {
if (checkDance.isChecked) {
        Toast.makeText(this, "dancing Selected!!",
Toast.LENGTH_SHORT).show()
        }
    })
  }
}
```

☞ **Output**


4:23
Registration_form
Welcome to Registration Form
Amay
8899776655
Palghar
Male
Female
Select language:
Java
Kotlin
Swift
Python
Scala
Perl
Reading
SUBMIT
4:24
Registration_form
Welcome to Registration Form
Amay
8899776655
Palghar
Male
Female
Select language: Kotlin
Reading
Singing
Dancing
SUBMIT
CANCEL
Selected item: Kotlin
4:24
Registration_form
Welcome to Registration Form
Amay
8899776655
Palghar
Male
Female
Select language: Kotlin
Reading
Singing
Dancing
SUBMIT
CANCEL
Reading Selected!!
4:25
Registration_form
Welcome to Registration Form
Amay
8899776655
Palghar
Male
Female
Select language: Kotlin
Reading
Singing
Dancing
SUBMIT
CANCEL
Submit button is clicked!!

Practical No 3

(I) Aim : Create an application to create Image Flipper and Image Gallery. On click on the image display the information about the image.

- **Step 1 : Add the following code to res/layout/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:background="#fff"
    android:orientation="vertical"
    android:weightSum="3"
    tools:context=".MainActivity">
    <!-- create a ImageView and Gallery -->
    <ImageView
    android:id="@+id/imageView"
    android:layout_width="fill_parent"
    android:layout_height="0dp"
    android:layout_weight="2"
    android:scaleType="fitXY" />
    <!-- By using android:spacing we can give spacing between images
    android:animationDuration="3000" -> for animation running -->
    <Gallery
    android:id="@+id/languagesGallery"
    android:layout_width="match_parent"
    android:layout_height="0dp"
    android:layout_weight="1"    android:layout_marginTop="100dp"
    android:animationDuration="2000"
    android:padding="10dp"
    android:spacing="5dp"
    android:unselectedAlpha="50" />
</LinearLayout>
```

Step 2 : Add the following code to src/MainActivity.kt

```
import android.os.Bundle
import android.support.v7.app.AppCompatActivity
import android.view.View
import android.widget.Gallery
import android.widget.ImageView
class MainActivity : AppCompatActivity() {
        private lateinit var simpleGallery: Gallery
        // CustomizedGalleryAdapter is a java class which extends BaseAdapter
        // and implement the override methods.
        private lateinit var customGalleryAdapter: CustomizedGalleryAdapter
        private lateinit var selectedImageView: ImageView
        // To show the selected language, we need this
        // array of images, here taken 10 different kind of
        // most popular programming languages private var images = intArrayOf(
          R.drawable.python,
        R.drawable.javascript,
        R.drawable.python,
        R.drawable.r,
        R.drawable.javascript,
        R.drawable.python,
        R.drawable.r,
        R.drawable.javascript
        )
        override fun onCreate(savedInstanceState: Bundle?)
        {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_main)
        // Our layout is activity_main
        // get the reference of Gallery. As we are showing
        // languages it is named as languagesGallery
        // meaningful names will be good for easier understanding
simpleGallery = findViewById<View>(R.id.languagesGallery) as Gallery
        // get the reference of ImageView
selectedImageView = findViewById<View>(R.id.imageView) as ImageView
        // initialize the adapter
customGalleryAdapter = CustomizedGalleryAdapter(applicationContext,
images)
```

```
        // set the adapter for gallery
        simpleGallery.adapter = customGalleryAdapter
      // Let us do item click of gallery and image can be identified by its position
        simpleGallery.setOnItemClickListener { parent, view, position, id ->
        // Whichever image is clicked, that is set in the selectedImageView
        // position will indicate the location of image
        selectedImageView.setImageResource(images[position])
        }
        }
}
```

- **Step 3 : Add the following code to src/CustomizedGalleryAdapter.kt**

```
package com.example.myapplication
import android.content.Context
import android.util.Log
import android.view.View
import android.view.ViewGroup
import android.widget.BaseAdapter
import android.widget.Gallery
import android.widget.ImageView
class CustomizedGalleryAdapter(private val context: Context, private val
images: IntArray) : BaseAdapter()
{
        // returns the number of images, in our example it is 10
        override fun getCount(): Int
        {
        return images.size
        }
        // returns the Item of an item, i.e. for our example we can get the image
        override fun getItem(position: Int): Any
        {
        return position
        }
        // returns the ID of an item override fun getItemId(position: Int): Long
        {
        return position.toLong()
        }
        // returns an ImageView view
override fun getView(position: Int, convertView: View?, parent: ViewGroup):
View
```

```
        {
        // position argument will indicate the location of image
        // create a ImageView programmatically
        val imageView = ImageView(context)
        // set image in ImageView
        imageView.setImageResource(images[position])
        // set ImageView param
        imageView.layoutParams = Gallery.LayoutParams(200, 200)
        return imageView
        }
}
```

☞ **Output**

(II) Aim : Create an application to use Gridview for shopping cart application.

- **Step 1 : Create a new project in Android Studio, go to File? New Project and fill all required details to create a new project.**
- **Step 2 : Add the following code to res/layout/activity_main.xml.**

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
   android:layout_width="match_parent"
   android:layout_height="match_parent"
       android:orientation="vertical">
       <!--
       GridView with 3 value for numColumns attribute
       -->
       <GridView
      android:id="@+id/simpleGridView"
      android:layout_width="fill_parent"
      android:layout_height="wrap_content"
       android:footerDividersEnabled="false"
       android:padding="1dp"
       android:numColumns="3" />
</LinearLayout>
```

- **Step 3 : Add the following code to src/MainActivity.kt**

```
import android.os.Bundle
import android.support.v7.app.AppCompatActivity
import android.widget.AdapterView.OnItemClickListener
import android.widget.GridView
import android.widget.Toast
class MainActivity : AppCompatActivity() {
        lateinit var gridView: GridView
        private var playerNames = arrayOf("Bike", "Dog", "Joker", "lion",
"Micky", "Teddy", "Dog", "Bike", "Joker","Bike", "Dog", "Joker", "lion",
"Micky", "Teddy", "Dog", "Bike", "Joker")
private var playerImages = intArrayOf(R.drawable.bike, R.drawable.dog,
R.drawable.joker,
R.drawable.lion, R.drawable.micky, R.drawable.teddy, R.drawable.dog,
R.drawable.bike, R.drawable.joker, R.drawable.bike, R.drawable.dog,
R.drawable.joker,
```

```
R.drawable.lion, R.drawable.micky, R.drawable.teddy, R.drawable.dog,
R.drawable.bike, R.drawable.joker)
        override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_main)
        title = "KotlinApp"
        gridView = findViewById(R.id.simpleGridView)
val mainAdapter = MainAdapter(this@MainActivity, playerNames,
playerImages)
gridView.adapter = mainAdapter
gridView.onItemClickListener = OnItemClickListener { _, _, position, _ ->
Toast.makeText(applicationContext, "You CLicked " +
playerNames[+position],
        Toast.LENGTH_SHORT).show()
        }
        }
}
```

- **Step 4 : Create a Kotlin class (MyAdapter.kt) and add the following code.**

```
import android.content.Context
import android.view.LayoutInflater
import android.view.View
import android.view.ViewGroup
import android.widget.BaseAdapter
import android.widget.ImageView
import android.widget.TextView
internal class MainAdapter(
        private val context: Context,
        private val numbersInWords: Array<String>,
        private val numberImage: IntArray
) :
        BaseAdapter() {
        private var layoutInflater: LayoutInflater? = null
        private lateinit var imageView: ImageView
        private lateinit var textView: TextView
        override fun getCount(): Int {
        return numbersInWords.size
        }
        override fun getItem(position: Int): Any? {
        return null
```

```
        }
        override fun getItemId(position: Int): Long {
        return 0
        }
        override fun getView(
        position: Int,
        convertView: View?,
        parent: ViewGroup
        ): View? {
        var convertView = convertView
        if (layoutInflater == null) {
        layoutInflater =
        context.getSystemService(Context.LAYOUT_INFLATER_SERVICE) as
LayoutInflater
        }
        if (convertView == null) {
            convertView = layoutInflater!!.inflate(R.layout.row_item, null)
        }
        imageView = convertView!!.findViewById(R.id.imageView)
        textView = convertView.findViewById(R.id.textView)
        imageView.setImageResource(numberImage[position])
        textView.text = numbersInWords[position]
        return convertView
        }
}
```

- **Step 5 : Create a Layout Resource file (row_item.xml) and add the following code.**

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:orientation="vertical"
        android:gravity="center"
        android:padding="8dp">
        <ImageView
        android:id="@+id/imageView"
          android:layout_width="100dp"
        android:layout_height="100dp" />
        <TextView
        android:textAlignment="center"
```

```
        android:gravity="center"
        android:id="@+id/textView"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginTop="16dp"
        android:text="Numbers"
        android:layout_marginBottom="10dp"
        android:textColor="@android:color/background_dark"
        android:textSize="24sp"
        android:textStyle="bold" />
</LinearLayout>
```

☞ **Output**

Practical 4(A)

Aim : Create an android application to demonstrate implicit and explicit intents

There are two types of intents in android

(1) Implicit Intent (2) Explicit Intent

(1) Implicit Intent

Implicit Intent doesn't specify the component. In such a case, intent provides information on available components provided by the system that is to be invoked. For example, you may write the following code to view the webpage.

MainActivity.kt

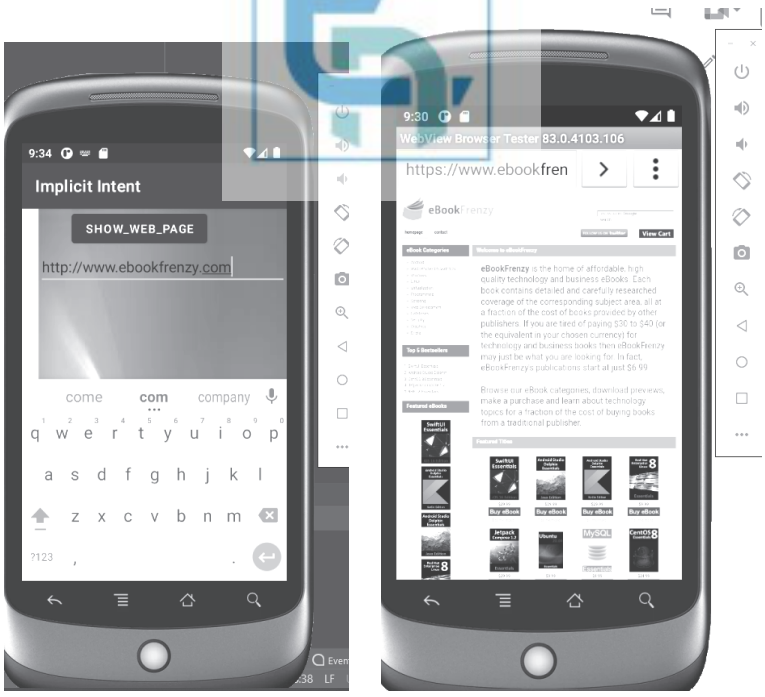
```
package com.example.option_menu
import android.content.Intent
import android.net.Uri
import android.os.Bundle
import android.view.View
import android.widget.*
import androidx.appcompat.app.AppCompatActivity
class MainActivity : AppCompatActivity() {
override fun onCreate(savedInstanceState: Bundle?) {
super.onCreate(savedInstanceState)
      setContentView(R.layout.activity_main)
val btnshow = findViewById<Button>(R.id.btnshow)
val edt = findViewById<EditText>(R.id.edt)
    btnshow.setOnClickListener{
val uri: String = edt.getText().toString()
val intent = Intent(Intent.ACTION_VIEW, Uri.parse(uri))
         startActivity(intent)
    }
  }
}
```

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"
```

```
xmlns:app="http://schemas.android.com/apk/res-auto"
android:layout_width="match_parent"
android:layout_height="match_parent"
android:background="@drawable/img"
android:orientation="vertical"
android:layout_marginLeft="20dp">
<Button
android:text="Show_Web_Page"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:id="@+id/btnshow"
android:layout_marginLeft="40dp"/>
<EditText
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:id="@+id/edt"
android:ems="15"
/>
</LinearLayout>
```

☞ **Output :** Enter url http://www.ebookfrenzy.com in edit text



(2) Explicit Intent

Explicit Intent specifies the component. In such a case, intent provides the external class to be invoked. In the below example, there are two activities (FirstActivity, and SecondActivity). When you click on the 'GO TO NEXT ACTIVITY' Button in the FirstActivity, then you move to the SecondActivity.

Example on Explicit Intent

☞ **Part 1: activity_main.xml**

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout
xmlns:android="http://schemas.android.com/apk/res/android"
xmlns:tools="http://schemas.android.com/tools"
xmlns:app="http://schemas.android.com/apk/res-auto"
android:layout_width="match_parent"
android:layout_height="match_parent"
android:orientation="vertical"
android:background="@color/teal_700">
<TextView
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:text="This is First Activity"
android:layout_above="@+id/btn"
android:layout_marginBottom="100dp"
android:layout_marginLeft="40dp"
android:textSize="30sp"/>
<Button
android:text="GoTo Next Activity"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:id="@+id/btn"
android:layout_marginLeft="40dp"
android:layout_centerInParent="true"/>
</RelativeLayout>
```

2) activity_next.xml

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

```
xmlns:tools="http://schemas.android.com/tools"
android:layout_width="match_parent"
android:layout_height="match_parent"
tools:context=".Next_Activity"
android:background="@color/purple_200">

<TextView
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:text="This is Second Activity"
android:layout_centerInParent="true"
android:textSize="30sp"/>
</RelativeLayout>
```

(3) MainActivity.kt

```
package com.example.option_menu
import android.content.Intent
import android.net.Uri
import android.os.Bundle
import android.view.View
import android.widget.*
import androidx.appcompat.app.AppCompatActivity
class MainActivity : AppCompatActivity() {
override fun onCreate(savedInstanceState: Bundle?) {
super.onCreate(savedInstanceState)
      setContentView(R.layout.activity_main)
val btn = findViewById<Button>(R.id.btn)
    btn.setOnClickListener{
val i = Intent(applicationContext, Next_Activity::class.java)
       startActivity(i)
    }
  }
}
```

(3) NextActivity.kt (Keep as it is)

```
package com.example.option_menu
import android.content.Intent
import androidx.appcompat.app.AppCompatActivity
import android.os.Bundle
import android.widget.Button
```

```
class Next_Activity : AppCompatActivity() {
override fun onCreate(savedInstanceState: Bundle?) {
super.onCreate(savedInstanceState)
      setContentView(R.layout.activity_next)
  }
}
```

☞ **Output**

Practical 4(B)

✍ **Aim : Create an android application to demonstrate shared preferences.**

- **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.

- In the following example when a user enters the data inside the text fields and when the user closes the application completely and again when the user opens the application the data entered by the user will be saved and set for the text fields.

☞ **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"
xmlns:app="http://schemas.android.com/apk/res-auto"
android:layout_width="match_parent"
android:layout_height="match_parent"
tools:context=".MainActivity"
android:orientation="vertical"
android:gravity="center_horizontal"
android:background="@color/teal_200">
<TextView
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:text="Example of Shared Preference"
android:layout_marginTop="50dp"
android:textSize="25sp"
android:textColor="@color/purple_700"
android:layout_marginLeft="30dp"/>
<EditText
android:id="@+id/editName"
android:layout_width="250dp"
android:layout_height="wrap_content"
android:inputType="textPersonName"
android:hint="Email_ID"
android:layout_marginTop="80dp"
/>
<EditText
android:id="@+id/editEmail"
android:layout_width="250dp"
android:layout_height="wrap_content"
android:inputType="textEmailAddress"
android:hint="Password"
android:layout_marginTop="20dp"
```

```
/>
<Button
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:text="Login"
android:layout_marginTop="20dp"
android:onClick="saveData"/>
</LinearLayout>
```

☞ **MainActivity.kt**

```
package com.example.option_menu
import android.os.Bundle
import android.preference.PreferenceManager
import android.view.Gravity
import android.view.View
import android.widget.EditText
import android.widget.Toast
import androidx.appcompat.app.AppCompatActivity
class MainActivity : AppCompatActivity() {
override fun onCreate(savedInstanceState: Bundle?) {
super.onCreate(savedInstanceState)
    setContentView(R.layout.activity_main)
val editEmail = findViewById<EditText>(R.id.editEmail)
val editName = findViewById<EditText>(R.id.editName)
val pref = PreferenceManager.getDefaultSharedPreferences(this)
    pref.apply {
val name = getString("NAME", "")
val email = getString("EMAIL", "")
        editName.setText(name)
        editEmail.setText(email)
    }
  }
fun saveData(v: View) {
val pref = PreferenceManager.getDefaultSharedPreferences(this)
val editor = pref.edit()
val editEmail = findViewById<EditText>(R.id.editEmail)
val editName = findViewById<EditText>(R.id.editName)
    editor
        .putString("NAME", editName.text.toString())
        .putString("EMAIL", editEmail.text.toString())
        .apply()
```

```
val toast = Toast.makeText(applicationContext, "Saved",
Toast.LENGTH_LONG)
        toast.setGravity(Gravity.TOP, 0, 140)
        toast.show()
    }
}
```

☞ **Output**

Practical No 5

I) Create an Android application to demonstrate the use of Broadcast listeners.

- **Step 1 : Create a New Project**
- **Step 2 : Working with the activity_main.xml file**

  Go to the activity_main.xml file and refer to the following code. Below is the code for the activity_main.xml file.

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

- **Step 3: Working with the MainActivity file.**

  Go to the MainActivity file and refer to the following code. Below is the code for the MainActivity file. Comments are added inside the code to understand the code in more detail.

```
import android.content.Intent
import android.content.IntentFilter
import android.os.Bundle
import android.support.v7.app.AppCompatActivity
class MainActivity : AppCompatActivity() {
    // register the receiver in the main activity in order
    // to receive updates of broadcasts events if they occur
    lateinit var receiver: AirplaneModeChangeReceiver
    override fun onCreate(savedInstanceState: Bundle?) {
    super.onCreate(savedInstanceState)
    setContentView(R.layout.activity_main)
    receiver = AirplaneModeChangeReceiver()
    // Intent Filter is useful to determine which apps wants to receive
    // which intents,since here we want to respond to change of
    // airplane mode
    IntentFilter(Intent.ACTION_AIRPLANE_MODE_CHANGED).also {
    // registering the receiver
    // it parameter which is passed in registerReceiver() function
    // is the intent filter that we have just created
    registerReceiver(receiver, it)
```

```
        }
        }
        // since AirplaneModeChangeReceiver class holds a instance of Context
        // and that context is actually the activity context in which
        // the receiver has been created
        override fun onStop() {
        super.onStop()
        unregisterReceiver(receiver)
        }
}
```

- **Step 4 : Create a new class**

  Go to app > java > your package name(in which the MainActicity is present) > right-click > New > Kotlin File/Class and name the files as AirplaneModeChangeReceiver. Below is the code for the AirplaneModeChangeReceiver file. Comments are added inside the code to understand the code in more detail.

```
import android.content.BroadcastReceiver
import android.content.Context
import android.content.Intent
import android.widget.Toast
// AirplaneModeChangeReceiver class extending BroadcastReceiver class
class AirplaneModeChangeReceiver : BroadcastReceiver() {
        // this function will be executed when the user changes his
        // airplane mode
        override fun onReceive(context: Context?, intent: Intent?) {
        // intent contains the information about the broadcast
        // in our case broadcast is change of airplane mode
        // if getBooleanExtra contains null value,it will directly return back
 val isAirplaneModeEnabled = intent?.getBooleanExtra("state", false) ?: return
        // checking whether airplane mode is enabled or not
        if (isAirplaneModeEnabled) {
        // showing the toast message if airplane mode is enabled
        Toast.makeText(context, "Airplane Mode Enabled",
Toast.LENGTH_LONG).show()
        } else {
        // showing the toast message if airplane mode is disabled
        Toast.makeText(context, "Airplane Mode Disabled",
Toast.LENGTH_LONG).show()
        }
        }
}
```

II) Create an Android application to create and use services.

- **Step 1 : Create a new project**
  1. Click on File, then New => New Project.
  2. Choose Empty activity
  3. Select language as Java/Kotlin
  4. Select the minimum SDK as per your need.
- **Step 2 : Modify strings.xml file**

  All the strings which are used in the activity are listed in this file.

```
<resources>
    <string name="app_name">Services_In_Android</string>
    <string name="heading">Services In Android</string>
    <string name="startButtonText">Start the Service</string>
    <string name="stopButtonText">Stop the Service</string>
</resources>
```

- **Step 3 : Working with the activity_main.xml file**

  Open the activity_main.xml file and add 2 Buttons in it which will start and stop the service. Below is the code for designing a proper activity layout.

```
<?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:background="#168BC34A"
    tools:context=".MainActivity">
    <LinearLayout
    android:id="@+id/linearLayout"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_centerVertical="true"
    android:orientation="vertical"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintVertical_bias="1.0"
```

```
        tools:ignore="MissingConstraints">
        <TextView
        android:id="@+id/textView1"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_marginBottom="170dp"
        android:text="@string/heading"
        android:textAlignment="center"
android:textAppearance="@style/TextAppearance.AppCompat.Large"
        android:textColor="@android:color/holo_green_dark"
        android:textSize="36sp"
        android:textStyle="bold" />
        <Button
        android:id="@+id/startButton"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:layout_marginStart="20dp"
        android:layout_marginTop="10dp"
        android:layout_marginEnd="20dp"
        android:layout_marginBottom="20dp"
        android:background="#4CAF50"
        android:text="@string/startButtonText"
        android:textAlignment="center"
      android:textAppearance="@style/TextAppearance.AppCompat.Display1"
        android:textColor="#FFFFFF"
        android:textStyle="bold" />
        <Button
        android:id="@+id/stopButton"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:layout_marginStart="20dp"
        android:layout_marginTop="10dp"
        android:layout_marginEnd="20dp"
        android:layout_marginBottom="20dp"
        android:background="#4CAF50"
        android:text="@string/stopButtonText"
        android:textAlignment="center"
      android:textAppearance="@style/TextAppearance.AppCompat.Display1"
        android:textColor="#FFFFFF"
```

```
        android:textStyle="bold" />
        </LinearLayout>
</LinearLayout>
```

- **Step 4 : Creating the custom service class**

  A custom service class will be created in the same directory where the MainActivity class resides and this class will extend the Service class. The callback methods are used to initiate and destroy the services. To play music, the MediaPlayer object is used. Below is the code to carry out this task.

```
package com.example.firebasedtataexample
import android.support.v7.app.AppCompatActivity
import android.os.Bundle
import android.app.Service
import android.content.Intent
import android.media.MediaPlayer
import android.os.IBinder
import android.provider.Settings
class NewService : Service() {
        // declaring object of MediaPlayer
        private lateinit var player:MediaPlayer
        // execution of service will start
        // on calling this method
        override fun onStartCommand(intent: Intent, flags: Int, startId: Int): Int {
        // creating a media player which
        // will play the audio of Default
        // ringtone in android device
        player = MediaPlayer.create(this,
Settings.System.DEFAULT_RINGTONE_URI)
        // providing the boolean
        // value as true to play
        // the audio on loop
        player.setLooping(true)
        // starting the process
        player.start()
        // returns the status
        // of the program
        return START_STICKY
        }
        // execution of the service will
        // stop on calling this method
```

```
        override fun onDestroy() {
        super.onDestroy()
        // stopping the process
        player.stop()
        package com.example.firebasedtataexample
import android.content.Intent
import android.content.IntentFilter
import android.os.Bundle
import android.support.v7.app.AppCompatActivity
import android.view.View
import android.widget.Button
class MainActivity : AppCompatActivity(), View.OnClickListener {
        // declaring objects of Button class
        private var start: Button? = null
        private var stop: Button? = null
        override fun onCreate(savedInstanceState: Bundle?) {
super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_main)
        // assigning ID of startButton
        // to the object start
        start = findViewById<View>(R.id.startButton) as Button
        // assigning ID of stopButton
        // to the object stop
        stop = findViewById<View>(R.id.stopButton) as Button
        // declaring listeners for the
        // buttons to make them respond
        // correctly according to the process
        start!!.setOnClickListener(this)
        stop!!.setOnClickListener(this)
        }
        override fun onClick(view: View) {
        // process to be performed
        // if start button is clicked
        if (view === start) {
        // starting the service
        startService(Intent(this, NewService::class.java))
        }
        // process to be performed
        // if stop button is clicked
        else if (view === stop) {
```

```
        // stopping the service
        stopService(Intent(this, NewService::class.java))
        }
        }
}
}
        override fun onBind(intent: Intent): IBinder? {
        return null
        }
}
```

- **Step 5 : Working with the MainActivity file**

```
package com.example.firebasedtataexample
import android.content.Intent
import android.content.IntentFilter
import android.os.Bundle
import android.support.v7.app.AppCompatActivity
import android.view.View
import android.widget.Button
class MainActivity : AppCompatActivity(), View.OnClickListener {
        // declaring objects of Button class
        private var start: Button? = null
        private var stop: Button? = null
        override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_main)
        // assigning ID of startButton
        // to the object start
        start = findViewById<View>(R.id.startButton) as Button
        // assigning ID of stopButton
        // to the object stop
        stop = findViewById<View>(R.id.stopButton) as Button
        // declaring listeners for the
        // buttons to make them respond
        // correctly according to the process
        start!!.setOnClickListener(this)
        stop!!.setOnClickListener(this)
        }
        override fun onClick(view: View) {
        // process to be performed
        // if start button is clicked
```

```
        if (view === start) {
        // starting the service
        startService(Intent(this, NewService::class.java))
        }
        // process to be performed
        // if stop button is clicked
        else if (view === stop) {
        // stopping the service
        stopService(Intent(this, NewService::class.java))
        }
        }
}
```

☞ **Output**

Practical 6(A)

Aim : Create an android application to demonstrate XML based animation.

- Adding a motion effect to any view, image, or text is called animation. An animation can be used to change the contour of a particular view or to provide motion.
- In Android, animation is typically utilized to give your UI a rich appearance and feel.

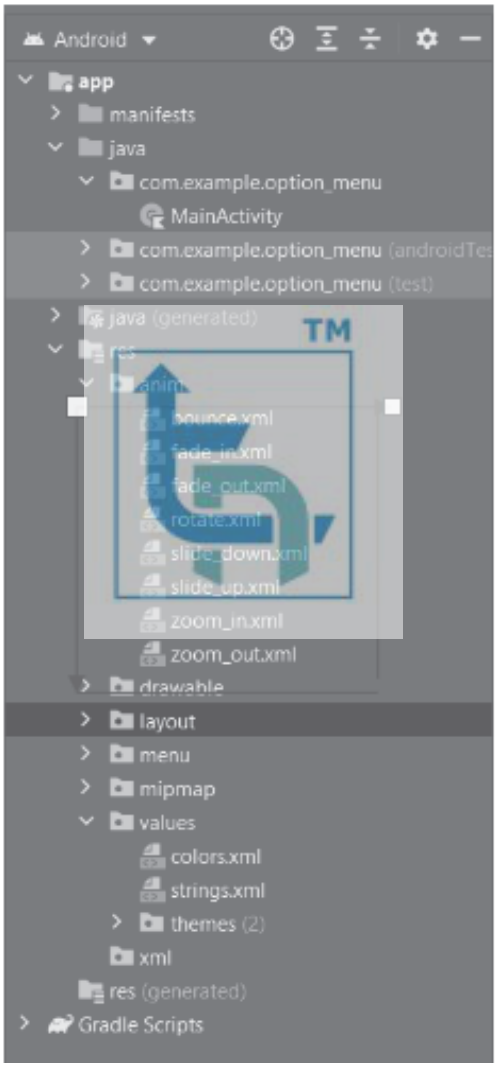
Activity_main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout
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:background="@color/cardview_shadow_start_color">
<TextView
android:id="@+id/textView"
android:layout_width="match_parent"
android:layout_height="match_parent"
android:layout_above="@+id/linearLayout"
android:gravity="center"
android:text="Geeks for Geeks"
android:textSize="32sp"
android:textColor="@color/design_default_color_primary"
android:textStyle="bold" />
<LinearLayout
android:id="@+id/linearLayout"
android:layout_width="match_parent"
android:layout_height="wrap_content"
android:layout_alignParentBottom="true"
android:orientation="vertical">
<LinearLayout
android:layout_width="match_parent"
android:layout_height="wrap_content"
```

```
android:orientation="horizontal"
android:weightSum="2">
<Button
android:id="@+id/fade_in"
android:layout_width="0dp"
android:layout_height="match_parent"
android:layout_weight="1"
android:text="Fade In"
android:textAllCaps="false" />
<Button
android:id="@+id/fade_out"
android:layout_width="0dp"
android:layout_height="match_parent"
android:layout_weight="1"
android:text="Fade Out"
android:textAllCaps="false" />
</LinearLayout>
<LinearLayout
android:layout_width="match_parent"
android:layout_height="wrap_content"
android:orientation="horizontal"
android:weightSum="2">
<Button
android:id="@+id/zoom_in"
android:layout_width="0dp"
android:layout_height="match_parent"
android:layout_weight="1"
android:text="Zoom In"
android:textAllCaps="false" />
<Button
android:id="@+id/zoom_out"
android:layout_width="0dp"
android:layout_height="match_parent"
android:layout_weight="1"
android:text="Zoom Out"
android:textAllCaps="false" />
</LinearLayout>
<LinearLayout
android:layout_width="match_parent"
android:layout_height="wrap_content"
```

```
android:orientation="horizontal"
android:weightSum="2">
<Button
android:id="@+id/slide_down"
android:layout_width="0dp"
android:layout_height="match_parent"
android:layout_weight="1"
android:text="Slide Down"
android:textAllCaps="false" />
<Button
android:id="@+id/slide_up"
android:layout_width="0dp"
android:layout_height="match_parent"
android:layout_weight="1"
android:text="Slide Up"
android:textAllCaps="false" />
</LinearLayout>
<LinearLayout
android:layout_width="match_parent"
android:layout_height="wrap_content"
android:orientation="horizontal"
android:weightSum="2">
<Button
android:id="@+id/bounce"
android:layout_width="0dp"
android:layout_height="match_parent"
android:layout_weight="1"
android:text="Bounce"
android:textAllCaps="false" />
<Button
android:id="@+id/rotate"
android:layout_width="0dp"
android:layout_height="match_parent"
android:layout_weight="1"
android:text="Rotate"
android:textAllCaps="false" />
</LinearLayout>
</LinearLayout>
</RelativeLayout>
```

- We will first create a folder name *anim*.In this folder, we will be adding the XML files which will be used to produce the animations.
- For this to happen, go to app/res right click and then select Android *Resource Directory* and name it as anim. Then right click on the anim folder – new – animation resource file then create the following xml file one by one.



(1) fade_in.xml

```
<?xml version="1.0" encoding="utf-8"?>
<set xmlns:android="http://schemas.android.com/apk/res/android"
android:interpolator="@android:anim/linear_interpolator">
<alpha
android:duration="1000"
android:fromAlpha="0.1"
android:toAlpha="1.0" />
</set>
```

(2) fade_out.xml

```
<?xml version="1.0" encoding="utf-8"?>
<set xmlns:android="http://schemas.android.com/apk/res/android"
android:interpolator="@android:anim/linear_interpolator">
<alpha
android:duration="1000"
android:fromAlpha="1.0"
android:toAlpha="0.1" />
</set>
```

(3) bounce.xml

```
<?xml version="1.0" encoding="utf-8"?>
<set
   xmlns:android="http://schemas.android.com/apk/res/android"
   android:interpolator="@android:anim/linear_interpolator"
   android:fillAfter="true">
<translate
      android:fromYDelta="100%"
      android:toYDelta="-20%"
      android:duration="300" />
<translate
      android:startOffset="500"
      android:fromYDelta="-20%"
      android:toYDelta="10%"
      android:duration="150" />
<translate
      android:startOffset="1000"
      android:fromYDelta="10%"
      android:toYDelta="0"
      android:duration="100" />
</set>
```

(4) rotate.xml

```
<?xml version="1.0" encoding="utf-8"?>
<rotate xmlns:android="http://schemas.android.com/apk/res/android"
   android:duration="1000"
   android:fromDegrees="0"
   android:interpolator="@android:anim/linear_interpolator"
   android:pivotX="50%"
   android:pivotY="50%"
   android:startOffset="0"
   android:toDegrees="360" />
```

(5) slide_down.xml

```
<?xml version="1.0" encoding="utf-8"?>
<set xmlns:android="http://schemas.android.com/apk/res/android">
<translate
      android:duration="1000"
      android:fromYDelta="-100%"
      android:toYDelta="0" />
</set>
```

(6) slide_up.xml

```
<set xmlns:android="http://schemas.android.com/apk/res/android">
<translate
      android:duration="1000"
      android:fromYDelta="0"
      android:toYDelta="-100%" />
</set>
```

(7) zoom_in.xml

```
<?xml version="1.0" encoding="utf-8"?>
<set xmlns:android="http://schemas.android.com/apk/res/android"
   android:fillAfter="true">
<scale xmlns:android="http://schemas.android.com/apk/res/android"
      android:duration="1000"
      android:fromXScale="1"
      android:fromYScale="1"
      android:pivotX="50%"
      android:pivotY="50%"
      android:toXScale="1.5"
      android:toYScale="1.5">
</scale>
</set>
```

(8) zoom_out.xml

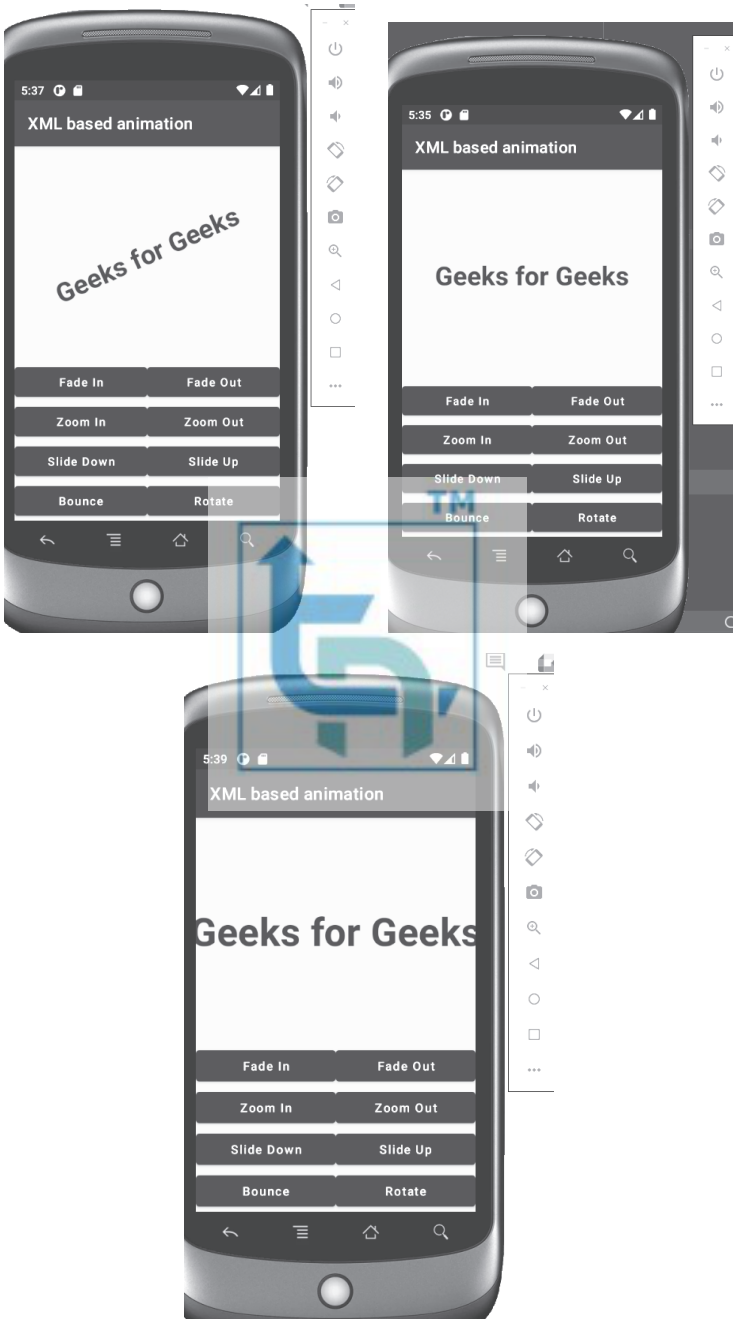
```
<?xml version="1.0" encoding="utf-8"?>
<set xmlns:android="http://schemas.android.com/apk/res/android"
   android:fillAfter="true" >
<scale
      xmlns:android="http://schemas.android.com/apk/res/android"
      android:duration="1000"
      android:fromXScale="1.0"
      android:fromYScale="1.0"
      android:pivotX="50%"
      android:pivotY="50%"
      android:toXScale="0.5"
      android:toYScale="0.5" >
</scale>
</set>
```

☞ **MainActivity.kt**

```
package com.example.option_menu
import androidx.appcompat.app.AppCompatActivity
import android.os.Bundle
import android.os.Handler
import android.view.View
import android.view.animation.AnimationUtils
import android.widget.Button
import android.widget.TextView
class MainActivity : AppCompatActivity() {
override fun onCreate(savedInstanceState: Bundle?) {
super.onCreate(savedInstanceState)
      setContentView(R.layout.activity_main)
val bounce = findViewById<Button>(R.id.bounce)
val textView=findViewById<TextView>(R.id.textView)
val fade_in = findViewById<Button>(R.id.fade_in)
val fade_out = findViewById<Button>(R.id.fade_out)
val zoom_in = findViewById<Button>(R.id.zoom_in)
val zoom_out = findViewById<Button>(R.id.zoom_out)
val slide_down = findViewById<Button>(R.id.slide_down)
val slide_up = findViewById<Button>(R.id.slide_up)
val rotate = findViewById<Button>(R.id.rotate)
      fade_in.setOnClickListener {
```

```
        textView.visibility = View.VISIBLE
val animationFadeIn = AnimationUtils.loadAnimation(this, R.anim.fade_in)
        textView.startAnimation(animationFadeIn)
    }
    fade_out.setOnClickListener {
val animationFadeOut = AnimationUtils.loadAnimation(this,
R.anim.fade_out)
        textView.startAnimation(animationFadeOut)
        Handler().postDelayed({
            textView.visibility = View.GONE
}, 1000)
    }
    zoom_in.setOnClickListener {
val animationZoomIn = AnimationUtils.loadAnimation(this, R.anim.zoom_in)
        textView.startAnimation(animationZoomIn)
    }
    zoom_out.setOnClickListener {
val animationZoomOut = AnimationUtils.loadAnimation(this,
R.anim.zoom_out)
        textView.startAnimation(animationZoomOut)
    }
    slide_down.setOnClickListener {
val animationSlideDown = AnimationUtils.loadAnimation(this,
R.anim.slide_down)
        textView.startAnimation(animationSlideDown)
    }
    slide_up.setOnClickListener {
val animationSlideUp = AnimationUtils.loadAnimation(this, R.anim.slide_up)
        textView.startAnimation(animationSlideUp)
    }
    bounce.setOnClickListener {
val animationBounce = AnimationUtils.loadAnimation(this, R.anim.bounce)
        textView.startAnimation(animationBounce)
    }
    rotate.setOnClickListener {
val animationRotate = AnimationUtils.loadAnimation(this, R.anim.rotate)
        textView.startAnimation(animationRotate)
    }
  }
}
```

☞ **Output**


5:37
XML based animation
Geeks for Geeks
Fade In
Fade Out
Zoom In
Zoom Out
Slide Down
Slide Up
Bounce
Rotate
5:35
XML based animation
Geeks for Geeks
Fade In
Fade Out
Zoom In
Zoom Out
Slide Down
Slide Up
Bounce
Rotate
5:39
XML based animation
Geeks for Geeks
Fade In
Fade Out
Zoom In
Zoom Out
Slide Down
Slide Up
Bounce
Rotate

Practical 6(B)

Aim : Create an android application to display canvas and allow the user to draw on it.

activity_main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout
xmlns:android="http://schemas.android.com/apk/res/android"
xmlns:app="http://schemas.android.com/apk/res-auto"
xmlns:tools="http://schemas.android.com/tools"
android:layout_width="match_parent"
android:layout_height="match_parent"
tools:context=".MainActivity">
<ImageView
android:id="@+id/image_view_1"
android:layout_width="match_parent"
android:layout_height="match_parent"
tools:ignore="ContentDescription"
android:background="@color/black"/>
</RelativeLayout>
```

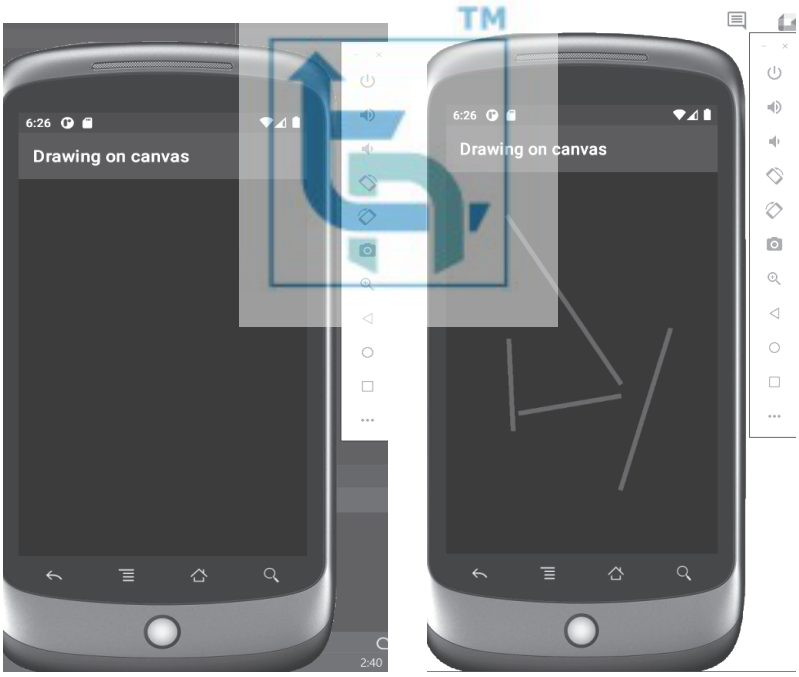
MainActivity.ky

```
package com.example.option_menu
import android.annotation.SuppressLint
import android.graphics.Bitmap
import android.graphics.Canvas
import android.graphics.Color
import android.graphics.Paint
import android.os.Build
import androidx.appcompat.app.AppCompatActivity
import android.os.Bundle
import android.view.MotionEvent
import android.view.View
import android.widget.ImageView
import androidx.annotation.RequiresApi
class MainActivity : AppCompatActivity(), View.OnTouchListener {
// Declaring ImageView, Bitmap, Canvas, Paint,
  // Down Coordinates and Up Coordinates
```

```
private lateinit var mImageView: ImageView
private lateinit var bitmap: Bitmap
private lateinit var canvas: Canvas
private lateinit var paint: Paint
private var downX = 0f
private var downY = 0f
private var upX = 0f
private var upY = 0f
@RequiresApi(Build.VERSION_CODES.R)
@SuppressLint("ClickableViewAccessibility")
override fun onCreate(savedInstanceState: Bundle?) {
super.onCreate(savedInstanceState)
    setContentView(R.layout.activity_main)
// Initializing the ImageView
mImageView = findViewById(R.id.image_view_1)
// Getting the current window dimensions
val currentDisplay = windowManager.currentWindowMetrics
val dw = currentDisplay.bounds.width()
val dh = currentDisplay.bounds.height()
// Creating a bitmap with fetched dimensions
bitmap = Bitmap.createBitmap(dw, dh, Bitmap.Config.ARGB_8888)
// Storing the canvas on the bitmap
canvas = Canvas(bitmap)
// Initializing Paint to determine
    // stoke attributes like color and size
paint = Paint()
paint.color = Color.RED
paint.strokeWidth = 10F
// Setting the bitmap on ImageView
mImageView.setImageBitmap(bitmap)
// Setting onTouchListener on the ImageView
mImageView.setOnTouchListener(this)
  }
// When Touch is detected on the ImageView,
  // Initial and final coordinates are recorded
  // and a line is drawn between them.
  // ImagView is updated
@SuppressLint("ClickableViewAccessibility")
override fun onTouch(v: View?, event: MotionEvent?): Boolean {
when (event!!.action) {
```

```
        MotionEvent.ACTION_DOWN -> {
downX = event.x
downY = event.y
}
        MotionEvent.ACTION_UP -> {
upX = event.x
upY = event.y
canvas.drawLine(downX, downY, upX, upY, paint)
mImageView.invalidate()
        }
      }
return true
}
}
```

☞ **Output**



Practical 7

I) Create a media player application in android that plays audio. Implement play, pause, and loop features.

- **Step 1 :** Create an empty activity project
- **Step 2 :** Create a raw resource folder under res.
- **Step 3 : Invoke the following code inside the activity_main.xml file to implement the UI.**

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout
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"
tools:ignore="HardcodedText">
<TextView
android:id="@+id/headingText"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_centerHorizontal="true"
android:layout_marginTop="32dp"
android:text="MEDIA PLAYER"
android:textSize="18sp"
android:textStyle="bold" />
<LinearLayout
android:layout_width="match_parent"
android:layout_height="wrap_content"
android:layout_below="@id/headingText"
android:layout_marginTop="16dp"
android:gravity="center_horizontal">
<Button
android:id="@+id/stopButton"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_marginEnd="8dp"
android:backgroundTint="#9C27B0"
android:text="STOP"
android:textColor="@android:color/white" />
```

```
<Button
android:id="@+id/playButton"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_marginEnd="8dp"
android:backgroundTint="#9C27B0"
android:text="PLAY"
android:textColor="@android:color/white" />
<Button
android:id="@+id/pauseButton"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:backgroundTint="#9C27B0"
android:text="PAUSE"
android:textColor="@android:color/white" />

</LinearLayout>

</RelativeLayout>
```

- **Step 4 : Invoke the following inside the** MainActivity.kt **file.**

```
package com.example.firebasedtataexample
import android.media.MediaPlayer
import android.os.Bundle
import android.support.v7.app.AppCompatActivity
import android.widget.Button
import com.example.firebasedtataexample.R
class MainActivity : AppCompatActivity() {
override fun onCreate(savedInstanceState: Bundle?) {
super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_main)
// create an instance of mediplayer for audio playback
val mediaPlayer: MediaPlayer = MediaPlayer.create(applicationContext,
R.raw.happybday)
// register all the buttons using their appropriate IDs
val bPlay: Button = findViewById(R.id.playButton)
val bPause: Button = findViewById(R.id.pauseButton)
val bStop: Button = findViewById(R.id.stopButton)
// handle the start button to
        // start the audio playback
bPlay.setOnClickListener {
// start method is used to start
            // playing the audio file
mediaPlayer.start()
```

```
}
// handle the pause button to put the
      // MediaPlayer instance at the Pause state
bPause.setOnClickListener {
// pause() method can be used to
         // pause the mediaplyer instance
mediaPlayer.pause()
}
// handle the stop button to stop playing
      // and prepare the mediaplayer instance
      // for the next instance of play
bStop.setOnClickListener {
// stop() method is used to completely
         // stop playing the mediaplayer instance
mediaPlayer.stop()
// after stopping the mediaplayer instance
         // it is again need to be prepared
         // for the next instance of playback
mediaPlayer.prepare()
}
}
}
```

☞ **Output**

ii) Create an Android application to use a camera and capture image/video and display them on the screen.

activity_main.xml

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

<!-- add Camera Button to open the Camera -->

<!-- add ImageView to display the captured image -->

<Button
android:id="@+id/camera_button"
android:layout_width="100dp"
android:layout_height="50dp"
android:layout_marginLeft="150dp"
android:background="#673AB7"
android:text="Open Camera"
android:layout_marginTop="10dp"
android:textColor="#ffffff" />

<ImageView
android:id="@+id/click_image"
android:layout_width="350dp"
android:layout_height="450dp"
android:layout_marginLeft="25dp"
android:layout_marginTop="200dp"
android:layout_marginBottom="10dp" />
</RelativeLayout>
```

MainActivity.ky

```
import android.content.Intent
import android.graphics.Bitmap
import android.os.Build
import android.os.Bundle
import android.provider.MediaStore
import android.support.v7.app.AppCompatActivity
```

```
import android.widget.Button
import android.widget.ImageView
class MainActivity : AppCompatActivity() {
// Define the button and imageview type variable
lateinit var camera_open_id: Button
lateinit var click_image_id: ImageView
companion object {
// Define the pic id
private const val pic_id = 123
}
override fun onCreate(savedInstanceState: Bundle?) {
super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_main)
// By ID we can get each component which id is assigned in XML file get
Buttons and imageview.
camera_open_id = findViewById(R.id.camera_button)
click_image_id = findViewById(R.id.click_image)

// Camera_open button is for open the camera and add the setOnClickListener
in this button
camera_open_id.setOnClickListener {
// Create the camera_intent ACTION_IMAGE_CAPTURE it will open the
camera for capture the image
val camera_intent = if (Build.VERSION.SDK_INT >=
Build.VERSION_CODES.CUPCAKE) {
            Intent(MediaStore.ACTION_IMAGE_CAPTURE)
        } else {
TODO("VERSION.SDK_INT < CUPCAKE")
        }
// Start the activity with camera_intent, and request pic id
startActivityForResult(camera_intent, pic_id)
}
}
// This method will help to retrieve the image
override fun onActivityResult(requestCode: Int, resultCode: Int, data: Intent?)
{
super.onActivityResult(requestCode, resultCode, data)
// Match the request 'pic id with requestCode
if (requestCode == pic_id) {
// BitMap is data structure of image file which store the image in memory
val photo = data!!.extras!!["data"] as Bitmap?
// Set the image in imageview for display
click_image_id.setImageBitmap(photo)
```

```
    }
  }
}
```

☞ **Output**

Practical 8(A)

✍ **Aim : Create an android application to implement Asynctask and threading concept**

☞ **AsyncTask**

The AsyncTask helps us to perform some operations in the background and update the results or status of the work to the main thread.

☞ **activity_main.xml**

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

```
xmlns:tools="http://schemas.android.com/tools"
android:layout_width="match_parent"
android:layout_height="match_parent"
tools:context=".MainActivity"
android:orientation="vertical">
<TextView
android:layout_width="wrap_content"
android:layout_height="280dp"
android:text="Hello World!"
android:id="@+id/statusText"
app:layout_constraintLeft_toLeftOf="parent"
app:layout_constraintRight_toRightOf="parent"
app:layout_constraintTop_toTopOf="parent"
android:gravity="bottom"/>
<Button
android:layout_width="match_parent"
android:layout_height="wrap_content"
android:id="@+id/btn_async"
android:text="Start Async Task"
app:layout_constraintBottom_toBottomOf="parent"
app:layout_constraintLeft_toLeftOf="parent"
app:layout_constraintRight_toRightOf="parent"
app:layout_constraintTop_toTopOf="@id/statusText"/>
<ProgressBar
android:id="@+id/progressBar"
style="?android:attr/progressBarStyleHorizontal"
android:layout_width="match_parent"
android:layout_height="wrap_content"
app:layout_constraintBottom_toBottomOf="parent"
app:layout_constraintLeft_toLeftOf="parent"
app:layout_constraintRight_toRightOf="parent"
app:layout_constraintTop_toTopOf="@id/btn_async"
android:visibility="gone"/>
</LinearLayout>
```

☞ **MainActivity.kt**

```
package com.example.option_menu
import android.os.AsyncTask
//import android.support.v7.app.AppCompatActivity
import android.os.Bundle
```

```
//import android.support.v4.widget.AutoScrollHelper
import android.text.method.ScrollingMovementMethod
import android.util.Log
import android.view.View
import android.widget.Button
import android.widget.ProgressBar
import android.widget.TextView
import androidx.appcompat.app.AppCompatActivity
//import kotlinx.android.synthetic.main.activity_main.*
class MainActivity : AppCompatActivity() {
private var btn: Button?=null ;
  private var statusText: TextView ?=null;
  private var statusProgress: ProgressBar?=null;
  private var stringBuilder:StringBuilder?=null

  override fun onCreate(savedInstanceState: Bundle?) {
super.onCreate(savedInstanceState)
    setContentView(R.layout.activity_main)
//Get the UI element from layout and initialize the UI element
btn= findViewById<Button>(R.id.btn_async);
    val progressBar = findViewById<ProgressBar>(R.id.progressBar)
statusText=findViewById<TextView>(R.id.statusText)
statusText?.movementMethod=ScrollingMovementMethod()
statusProgress=findViewById(R.id.progressBar)
    progressBar.max=2;
progressBar.progress = 0
stringBuilder = StringBuilder("Async task Demo\n----------")
stringBuilder?.append("\nWaiting to start the Async Jobs\n")
statusText?.text = "${stringBuilder.toString()}"
// Set on Click listener on start async button
btn?.setOnClickListener(View.OnClickListener {
// Declare the AsyncTask and start the execution
var  task:MyAsyncTask = MyAsyncTask()
      task.execute("www.nplix.com","www.debugandroid.com")
    })
  }
// Create inner class by extending the AsyncTask
inner class MyAsyncTask : AsyncTask<String, Int, Int>(){
//Override the doInBackground method
override fun doInBackground(vararg params: String?): Int {
val count:Int=params.size
var index=0
while (index<count){
```

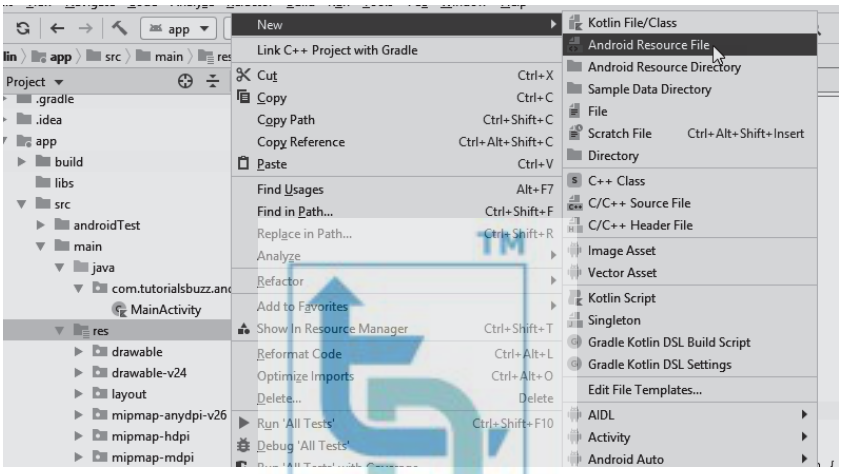
```
            Log.d("Kotlin","In doInBackground Method and Total parameter
passed is :$count " +
"and processing $index with value: ${params[index]}")
// Publish the progress
publishProgress(index+1)
//Sleeping for 1 seconds
Thread.sleep(1000)
            index++
        }
return count;
}
// Override the onProgressUpdate method to post the update on main thread
override fun onProgressUpdate(vararg values: Int?) {
super.onProgressUpdate(*values)
if(values[0]!=null) {
statusProgress?.progress = values[0] as Int
stringBuilder?.append("Publish post called with ${values[0]}\n")
statusText?.text = stringBuilder.toString()
        }
    }
// Setup the intial UI before execution of the background task
override fun onPreExecute() {
super.onPreExecute()
val progressBar = findViewById<ProgressBar>(R.id.progressBar)
stringBuilder?.append("Async task started... \n In PreExecute Method \n")
statusText?.text = "${stringBuilder.toString()}"
progressBar.visibility=View.VISIBLE
progressBar.progress = 0
Log.d("Kotlin","On PreExecute Method")
    }
// Update the final status by overriding the OnPostExecute method.
override fun onPostExecute(result: Int?) {
super.onPostExecute(result)
stringBuilder?.append("Task Completed.\n")
statusText?.text = "${stringBuilder.toString()}"
Log.d("Kotlin","On Post Execute and size of String is:$result")
    }
  }
}
```

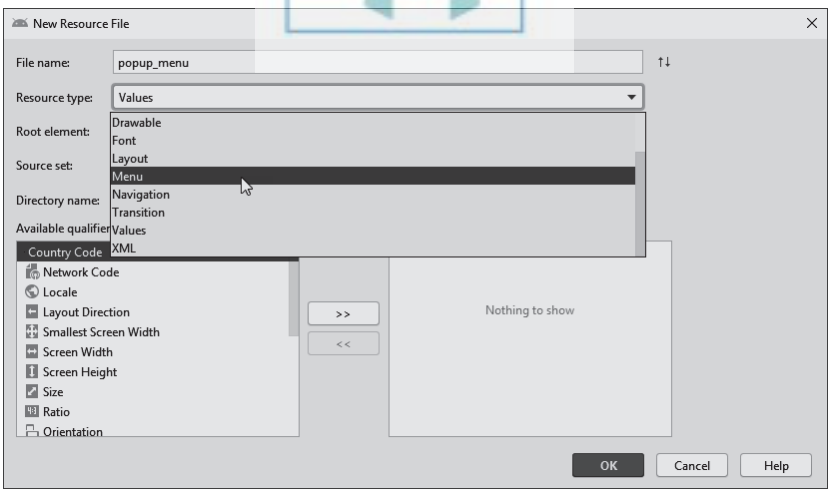

7:03
AsyncTask Example
Async task Demo
Waiting to start the Async Jobs
START ASYNC TASK
7:03
AsyncTask Example
Async task Demo
Waiting to start the Async Jobs
Async task started...
In PreExecute Method
Publish post called with 1
Publish post called with 2
Task Completed.
START ASYNC TASK
7:04
AsyncTask Example
START ASYNC TASK

Practical 8(B)

Aim : Create an android application to demonstrate the different types of menus.

(1) Popup menu : Android popup menu displays a list of items in a vertical list which presents the view that invoked the menu and is useful to provide an overflow of actions related to specific content.





☞ **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/my_button"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:text="Popup Menu"
app:layout_constraintBottom_toBottomOf="parent"
app:layout_constraintLeft_toLeftOf="parent"
app:layout_constraintRight_toRightOf="parent"
app:layout_constraintTop_toTopOf="parent" />
</androidx.constraintlayout.widget.ConstraintLayout>
```

☞ **Popup_menu.xml**

```
<?xml version="1.0" encoding="utf-8"?>
<menu 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"
tools:context="com.tutorialsbuzz.androidoptionmenu.MainActivity">
<item
android:id="@+id/header1"
android:title="Item 1 " />
<item
android:id="@+id/header2"
android:title="Item 2" />
<item
android:id="@+id/header3"
android:title="Item 3" />
</menu>
```

☞ **MainActivity.kt**

```
package com.example.popupmenu
import androidx.appcompat.app.AppCompatActivity
import android.os.Bundle
import android.view.MenuItem
import android.view.View
import android.widget.PopupMenu
import android.widget.Toast
import android.widget.Button
class MainActivity : AppCompatActivity() {
override fun onCreate(savedInstanceState: Bundle?) {
super.onCreate(savedInstanceState)
      setContentView(R.layout.activity_main)
val my_button = findViewById<Button>(R.id.my_button)
      my_button.setOnClickListener {
         showPopup(my_button)
      }
   }
private fun showPopup(view: View) {
val popup = PopupMenu(this, view)
      popup.inflate(R.menu.popup_menu)
popup.setOnMenuItemClickListener(PopupMenu.OnMenuItemClickListener {
item: MenuItem? ->
when (item!!.itemId) {
            R.id.header1 -> {
               Toast.makeText(this@MainActivity, "Item 1 selected",
Toast.LENGTH_SHORT).show()
            }
            R.id.header2 -> {
               Toast.makeText(this@MainActivity, "Item 2 selected",
Toast.LENGTH_SHORT).show()
            }
            R.id.header3 -> {
               Toast.makeText(this@MainActivity, "Item 3 selected",
Toast.LENGTH_SHORT).show()
            }
         }
true
})
      popup.show()
   }
}
```

☞ **Output**


11:50
PopUpMenu
POPUP MENU
11:59
PopUpMenu
POPUP MENU
Item 1
Item 2
Item 3
12:00
PopUpMenu
POPUP MENU
Item 1 selected

(2) Context Menu: Android context menu is a floating menu that only appears when the user clicks for a long time on an element and is useful for elements that affect the selected content or context frame.

☞ **activity_main.xml**

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
xmlns:app="http://schemas.android.com/apk/res-auto"
xmlns:tools="http://schemas.android.com/tools"
android:layout_width="match_parent"
android:layout_height="match_parent"
android:orientation="vertical"
tools:context=".MainActivity">
<TextView
android:layout_width="match_parent"
android:layout_height="40sp"
android:gravity="center"
android:text="Long Press On List Item to Open Context Menu"
android:textStyle="bold"
app:layout_constraintBottom_toBottomOf="parent"
app:layout_constraintLeft_toLeftOf="parent"
app:layout_constraintRight_toRightOf="parent"
app:layout_constraintTop_toTopOf="parent" />
<ListView
android:id="@+id/listView"
android:layout_width="match_parent"
android:layout_height="match_parent" />
</LinearLayout>
```

☞ **MainActivity.kt**

```
package com.example.context_menu
import android.os.Bundle
//import android.support.v7.app.AppCompatActivity
import android.view.ContextMenu
import android.view.MenuItem
import android.view.View
import android.widget.AdapterView.AdapterContextMenuInfo
import android.widget.ArrayAdapter
import android.widget.ListView
import android.widget.Toast
import androidx.appcompat.app.AppCompatActivity
```

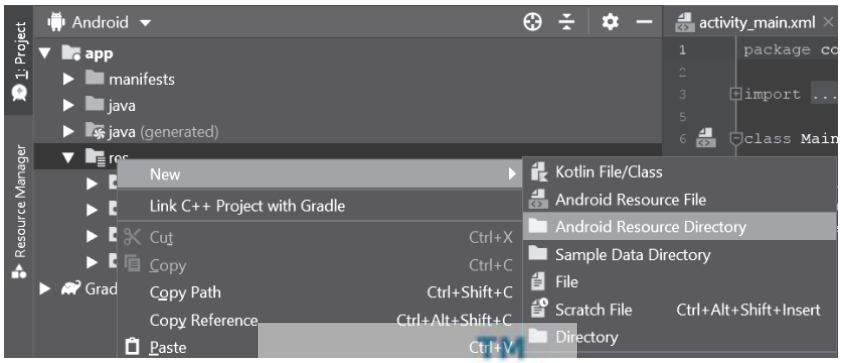
```
class MainActivity : AppCompatActivity() {
lateinit var array: Array<String>
override fun onCreate(savedInstanceState: Bundle?) {
super.onCreate(savedInstanceState)
      setContentView(R.layout.activity_main)
//Find View By Id For ListView
val listView = findViewById<ListView>(R.id.listView) as ListView
//Array of String Which Contain Name of Persons.
array = arrayOf("Sachin", "Vishal", "Rishu", "Krishank", "Vivek", "Jatin",
"Raj", "Rajan", "Nikhil")
//Creating Adapter
val adp = ArrayAdapter(this@MainActivity,
android.R.layout.simple_list_item_1, array)
//Set Adapter to ListView
listView.adapter = adp
//Register ListView for Context Menu
registerForContextMenu(listView)
  }
override fun onCreateContextMenu(menu: ContextMenu?, v: View?, menuInfo:
ContextMenu.ContextMenuInfo?) {
super.onCreateContextMenu(menu, v, menuInfo)
//Set Header of Context Menu
menu!!.setHeaderTitle("Select Option")
menu.add(0, v!!.id, 0, "Call")
menu.add(0, v.id, 1, "SMS")
menu.add(0, v.id, 2, "Email")
menu.add(0, v.id, 3, "WhatsApp")
  }
override fun onContextItemSelected(item: MenuItem): Boolean {
//Get Order of Selected Item
val selectedItemOrder = item!!.order
//Get Title Of Selected Item
val selectedItemTitle = item.title
//To get Name of Person Click on ListView
val info = item.menuInfo as AdapterContextMenuInfo
val listPosition = info.position
val name = array[listPosition]

      Toast.makeText(this@MainActivity, " " + selectedItemTitle + " " +
name, Toast.LENGTH_LONG).show()
return true
}
}
```

☞ **Output**

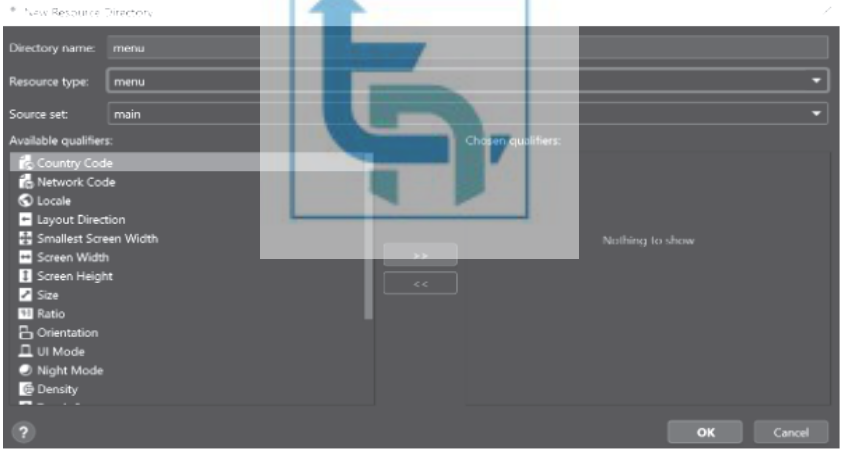

1:01
Context_menu
Long Press On List Item to Open Context Menu
Sachin
Vishal
Rishu
Krishank
Vivek
Jatin
Raj
Rajan
1:03
Context_menu
Long Press On List Item to Open Context Menu
Select Option
Call
SMS
Email
WhatsApp
1:05
Context_menu
Long Press On List Item to Open Context Menu
Call Sachin

(3) Option_menu : Android options menu is a primary collection of menu items in an android application and is useful for actions that have a global impact on the searching application.

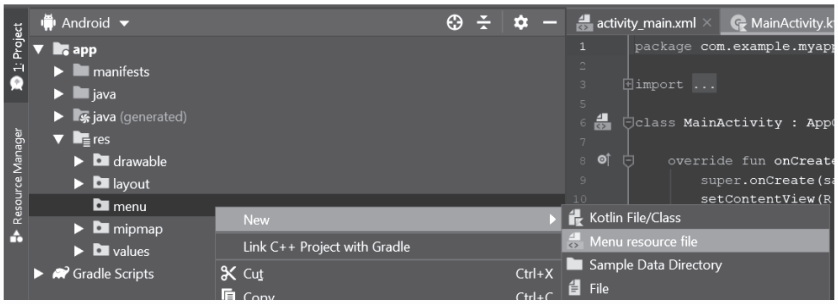
- To create the menu directory, right-click the **res** folder (found by navigating through **Project>app**) and select **New>Android Resource Directory**



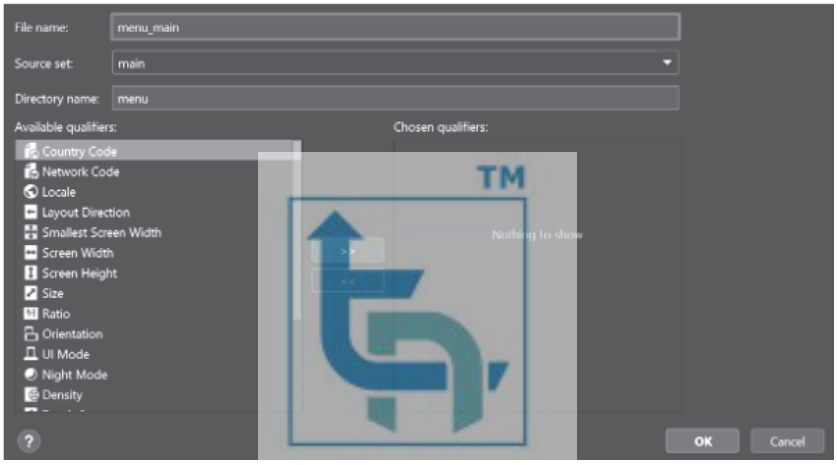
- Set both the directory name and resource type to menu then press OK



- Once the menu directory is in place, you can create the layout. To create a new layout resource, right-click the menu directory then select New > Menu resource file.



- Set the file name to menu_main then press OK.



☞ **Menu_main.xml**

```
<?xml version="1.0" encoding="utf-8"?>
<menu xmlns:android="http://schemas.android.com/apk/res/android">
<item
android:id="@+id/menu1"
android:title="Save"></item>
<item
android:id="@+id/menu2"
android:title="Edit"></item>
<item
android:id="@+id/menu3"
android:title="Delete"></item>
<item
android:id="@+id/menu4"
android:title="Search"></item>
```

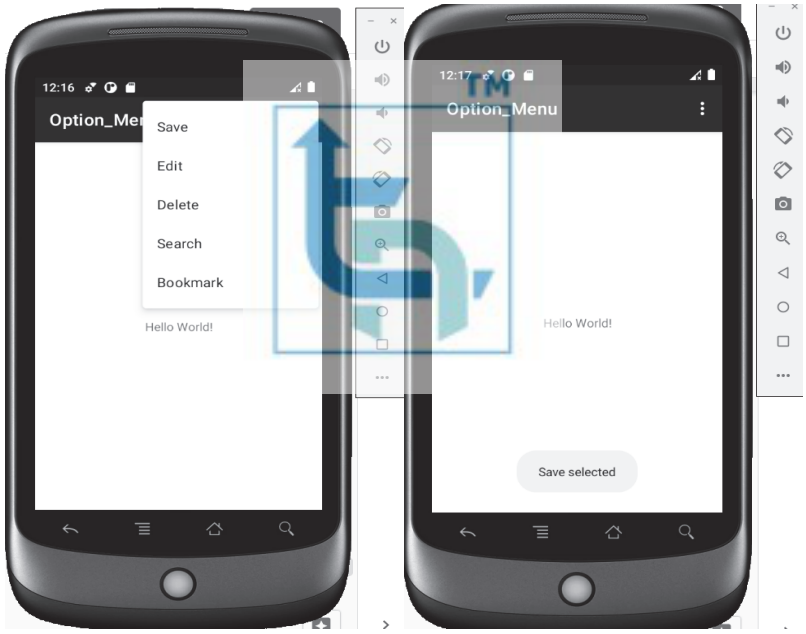
```
<item
android:id="@+id/menu5"
android:title="Bookmark"></item>
</menu>
```

☞ **MainActivity.kt**

```
package com.example.option_menu
import android.os.Bundle
import androidx.appcompat.app.AppCompatActivity
import android.view.Menu
import android.view.MenuItem
import android.widget.Toast

class MainActivity : AppCompatActivity() {
override fun onCreate(savedInstanceState: Bundle?) {
super.onCreate(savedInstanceState)
      setContentView(R.layout.activity_main) }
override fun onCreateOptionsMenu(menu: Menu): Boolean {
menuInflater.inflate(R.menu.menu_main, menu)
return true
}
override fun onOptionsItemSelected(item: MenuItem): Boolean {
when (item.itemId) {
        R.id.menu1 -> {
           Toast.makeText(this, "Save selected",
Toast.LENGTH_SHORT).show()
return true
}
        R.id.menu2 -> {
           Toast.makeText(this, "Edit selected",
Toast.LENGTH_SHORT).show()
return true
}
        R.id.menu3 -> {
           Toast.makeText(this, "Delete selected",
Toast.LENGTH_SHORT).show()
return true
}
        R.id.menu4 -> {
```

```
            Toast.makeText(this, "Search selected",
Toast.LENGTH_SHORT).show()
return true
}
        R.id.menu5 -> {
            Toast.makeText(this, "Bookmark selected",
Toast.LENGTH_SHORT).show()
return true
}
else ->return super.onOptionsItemSelected(item)
      } }
}
```

☞ **Output**



Practical 9

I) Aim : Create an Android application to record the current location. Based on the current location allow the user to use some useful services/applications

- **Step 1 : add below permission in AndroidManifest.xml file**

```
<uses-permission
android:name="android.permission.ACCESS_COARSE_LOCATION" />
<uses-permission
android:name="android.permission.ACCESS_FINE_LOCATION" />
<uses-permission android:name="android.permission.INTERNET" />
```

- **Step 2 : Edit activity_main.xml as below**

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout
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:padding="4dp"
tools:context=".MainActivity">
<TextView
android:id="@+id/text"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_centerHorizontal="true"
android:layout_marginTop="70dp"
android:background="#008080"
android:padding="5dp"
android:text="Click below button to get current Location"
android:textColor="#fff"
android:textSize="24sp"
android:textStyle="bold" />
<TextView
android:id="@+id/textView"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_centerInParent="true"
android:text="Get GPS Location"
android:textColor="@android:color/holo_red_dark"
```

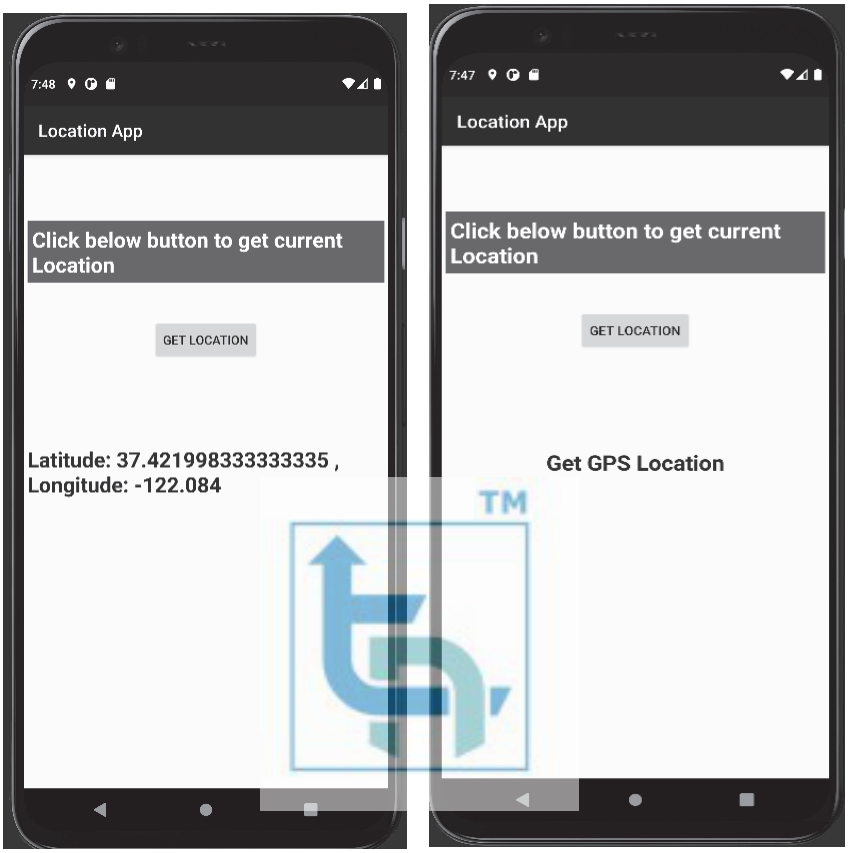
```
android:textSize="24sp"
android:textStyle="bold" />
<Button
android:id="@+id/getLocation"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_below="@+id/text"
android:layout_centerInParent="true"
android:layout_marginTop="40dp"
android:text="Get location" />
</RelativeLayout>
```

- **Step 3 : Edit MainActivity.xml as below**

```
package com.example.myapplication
import android.support.v7.app.AppCompatActivity
import android.os.Bundle
import android.Manifest
import android.content.Context
import android.content.pm.PackageManager
import android.location.Location
import android.location.LocationListener
import android.location.LocationManager
import android.support.v4.app.ActivityCompat
import android.support.v4.content.ContextCompat
import android.widget.Button
import android.widget.TextView
import android.widget.Toast
import android.content.Intent
import android.net.Uri
import java.util.*
class MainActivity : AppCompatActivity(), LocationListener {
private lateinit var locationManager: LocationManager
private lateinit var tvGpsLocation: TextView
private val locationPermissionCode = 2
override fun onCreate(savedInstanceState: Bundle?) {
super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_main)
title = "Location App"
val button: Button = findViewById(R.id.getLocation)
        button.setOnClickListener {
getLocation()
}
}
private fun getLocation() {
```

```
locationManager = getSystemService(Context.LOCATION_SERVICE) as
LocationManager
if ((ContextCompat.checkSelfPermission(this,
Manifest.permission.ACCESS_FINE_LOCATION) !=
PackageManager.PERMISSION_GRANTED)) {
        ActivityCompat.requestPermissions(this,
arrayOf(Manifest.permission.ACCESS_FINE_LOCATION),
locationPermissionCode)
    }
locationManager.requestLocationUpdates(LocationManager.GPS_PROVIDER,
5000, 5f, this)
  }
override fun onLocationChanged(location: Location) {
tvGpsLocation = findViewById(R.id.textView)
tvGpsLocation.text = "Latitude: " + location.latitude + " , Longitude: " +
location.longitude

//call openMap function to open google map
openMap(location)
  }
fun openMap(location: Location)
  {
val uri: String = java.lang.String.format(Locale.ENGLISH, "geo:%f,%f",
location.latitude, location.longitude)
val intent = Intent(Intent.ACTION_VIEW, Uri.parse(uri))
this.startActivity(intent)
  }
override fun onRequestPermissionsResult(requestCode: Int, permissions:
Array<out String>, grantResults: IntArray) {
super.onRequestPermissionsResult(requestCode, permissions, grantResults)
if (requestCode == locationPermissionCode) {
if (grantResults.isNotEmpty() && grantResults[0] ==
PackageManager.PERMISSION_GRANTED) {
        Toast.makeText(this, "Permission Granted",
Toast.LENGTH_SHORT).show()
      }
else {
        Toast.makeText(this, "Permission Denied",
Toast.LENGTH_SHORT).show()
      }
    }
  }
}
```

☞ **Output**



Practical 10

Aim : Create an android application to store and retrieve data in the SQLite database.

- In order to perform database operations on Android devices, such as storing, altering, or retrieving permanent data from the database, SQLite is an open-source relational database.

☞ **activity_main.xml**

```
<LinearLayout
xmlns:android="http://schemas.android.com/apk/res/android"
android:layout_width="match_parent"
```

```
android:layout_height="match_parent"
android:gravity="center"
android:orientation="vertical">
<EditText android:id="@+id/editID"
android:layout_width="match_parent"
android:layout_height="wrap_content"
android:hint="Enter Unique ID"/>
<EditText android:id="@+id/editName"
android:layout_width="match_parent"
android:layout_height="wrap_content"
android:hint="Enter your Name here"/>
<EditText android:id="@+id/editEmail"
android:layout_width="match_parent"
android:layout_height="wrap_content"
android:hint="Enter your Email Address"/>
<EditText android:id="@+id/editCourse"
android:layout_width="match_parent"
android:layout_height="wrap_content"
android:hint="Enter your Course Name"/>
<Button android:id="@+id/btnInsert"
android:layout_width="match_parent"
android:layout_height="wrap_content"
android:text="Insert Data"/>
<Button android:id="@+id/btnUpdate"
android:layout_width="match_parent"
android:layout_height="wrap_content"
android:text="Update Data"/>
<Button android:id="@+id/btnDelete"
android:layout_width="match_parent"
android:layout_height="wrap_content"
android:text="Delete Data"/>
<Button android:id="@+id/btnViewAll"
android:layout_width="match_parent"
android:layout_height="wrap_content"
android:text="View All Data"/>
</LinearLayout>
```

☞ Creating a new class for SQLite operations

- Navigate to app > java > your project's package name > Right-click on it > New > Kotlin class and name it as DatabaseHelper and add the below code to it.

☞ DatabaseHelper.kt

```
package com.example.popupmenu
import android.content.Context
import android.database.sqlite.SQLiteOpenHelper
import android.content.ContentValues
import android.database.Cursor
import android.database.sqlite.SQLiteDatabase
class DatabaseHelper(context: Context) :
  SQLiteOpenHelper(context, DATABASE_NAME, null, 1) {
override fun onCreate(db: SQLiteDatabase) {
      db.execSQL( "CREATE TABLE $TABLE_NAME (ID INTEGER
PRIMARY KEY " + "AUTOINCREMENT,NAME TEXT,EMAIL
TEXT,COURSE TEXT)")
  }
override fun onUpgrade(db: SQLiteDatabase, oldVersion: Int,newVersion: Int)
{
      db.execSQL("DROP TABLE IF EXISTS " + TABLE_NAME)
      onCreate(db) }
fun insertData(name: String, email: String, course: String) {
val db = this.writableDatabase
val contentValues = ContentValues()
      contentValues.put(COL_1, name)
      contentValues.put(COL_2, email)
      contentValues.put(COL_3, course)
      db.insert(TABLE_NAME, null, contentValues) }
fun updateData(id: String, name: String, email: String, course:String):
        Boolean {
val db = this.writableDatabase
val contentValues = ContentValues()
      contentValues.put(COL_0, id)
      contentValues.put(COL_1, name)
      contentValues.put(COL_2, email)
      contentValues.put(COL_3, course)
      db.update(TABLE_NAME, contentValues, "ID = ?", arrayOf(id))
return true }
```

```
fun deleteData(id: String): Int {
val db = this.writableDatabase
return db.delete(TABLE_NAME, "ID = ?", arrayOf(id)) }
val allData: Cursor
get() {
val db = this.writableDatabase
val res = db.rawQuery("SELECT * FROM " + TABLE_NAME, null)
return res }
companion object {
val DATABASE_NAME = "student.db"
val TABLE_NAME = "student_table"
val COL_0 = "ID"
val COL_1 = "NAME"
val COL_2 = "EMAIL"
val COL_3 = "COURSE"
}
}
```

☞ **MainActivity.kt**

```
package com.example.popupmenu

import android.os.Bundle
import android.widget.Toast
import android.view.View
import android.widget.Button
import android.widget.EditText
import androidx.appcompat.app.AlertDialog
import androidx.appcompat.app.AppCompatActivity

class MainActivity : AppCompatActivity() {
var dbHelper = DatabaseHelper(this)
fun showToast(text: String) {
    Toast.makeText(this, text, Toast.LENGTH_LONG).show()

  }
fun showDialog(title: String, Message: String) {
val builder = AlertDialog.Builder(this)
    builder.setCancelable(true)
    builder.setTitle(title)
    builder.setMessage(Message)
```

```
        builder.show()
    }
fun clearEditTexts() {
val editID = findViewById<EditText>(R.id.editID)
val editName = findViewById<EditText>(R.id.editName)
val editEmail = findViewById<EditText>(R.id.editEmail)
val editCourse = findViewById<EditText>(R.id.editCourse)
        editID.setText("")
        editName.setText("")
        editEmail.setText("")
        editCourse.setText("")
    }
override fun onCreate(savedInstanceState: Bundle?) {
super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_main)
        handleInserts()
        handleUpdates()
        handleDeletes()
        handleViewing()
    }
fun handleInserts() {
val btnInsert = findViewById<Button>(R.id.btnInsert)
        btnInsert.setOnClickListener {
val editName = findViewById<EditText>(R.id.editName)
val editEmail = findViewById<EditText>(R.id.editEmail)
val editCourse = findViewById<EditText>(R.id.editCourse)
try {
dbHelper.insertData(
                editName.text.toString(),
editEmail.text.toString(),
editCourse.text.toString() )
                clearEditTexts()
                showToast("Data Inserted Successfully")
            } catch (e: Exception) {
                e.printStackTrace()
                showToast(e.message.toString()) } } }
fun handleUpdates() {
val btnUpdate = findViewById<Button>(R.id.btnUpdate)
        btnUpdate.setOnClickListener {
val editID = findViewById<EditText>(R.id.editID)
```

```
val editName = findViewById<EditText>(R.id.editName)
val editEmail = findViewById<EditText>(R.id.editEmail)
val editCourse = findViewById<EditText>(R.id.editCourse)
try {
dbHelper.updateData(
            editID.text.toString(),
editName.text.toString(),
editEmail.text.toString(),
editCourse.text.toString() )
            clearEditTexts()
            showToast("Data Updated Successfully")
        } catch (e: Exception) {
            e.printStackTrace()
            showToast(e.message.toString()) } } }
fun handleDeletes() {
val btnDelete = findViewById<Button>(R.id.btnDelete)
    btnDelete.setOnClickListener {
val editID = findViewById<EditText>(R.id.editID)
try {
dbHelper.deleteData(editID.text.toString())
            clearEditTexts()
            showToast("Data Deleted Successfully")
        } catch (e: Exception) {
            e.printStackTrace()
            showToast(e.message.toString()) } } }
fun handleViewing() {
val btnViewAll = findViewById<Button>(R.id.btnViewAll)
    btnViewAll.setOnClickListener(
        View.OnClickListener {
val res = dbHelper.allData
if (res.count == 0) {
            showDialog("Error", "No Data Found")
return@OnClickListener }
val buffer = StringBuffer()
while (res.moveToNext()) {

            buffer.append("ID :" + res.getString(0) + "\n")
            buffer.append("NAME :" + res.getString(1) + "\n")
            buffer.append("EMAIL :" + res.getString(2) + "\n")
            buffer.append("COURSE :" + res.getString(3) + "\n\n")
```

```
            }
            showDialog("Data", buffer.toString())
        } )
    }
}
```

☞ **Output**

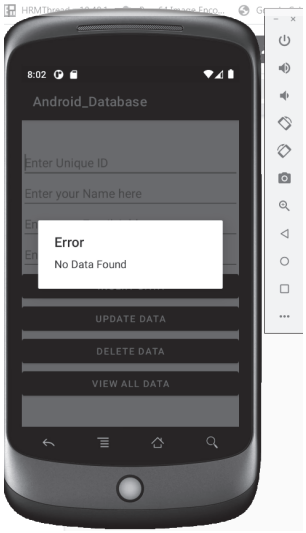
- **Screen 1 :** Don't enter Unique ID while inserting data (It's Auto-incremented)
- **Screen 2 :** Toast Message is displayed after clicking on Insert Data Button
- **Screen 3 :** Data viewed in a dialog box after clicking View All Data Button

- **Screen 1 :** Fill details in all fields while updating data (Also enter Unique ID)
- **Screen 2 :** Toast Message is displayed after clicking on Update Data Button
- **Screen 3 :** Updated data viewed in dialog box after clicking View All Data Button


8:15
Android_Database
1
Enter your Name here
Enter your Email Address
Enter your Course Name
INSERT DATA
UPDATE DATA
DELETE DATA
VIEW ALL DATA
8:21
Android_Database
1
Janhavi Raut
jau04@gmail.com
Bsc.IT
INSERT DATA
UPDATE DATA
DELETE DATA
VIEW ALL DATA
8:19
Android_Database
Enter Unique ID
Enter your Name here
Enter your Email Address
Enter your Course Name
INSERT DATA
UPDATE DATA
DELETE DATA
VIEW ALL DATA
Data Updated Successfully
8:22
Android_Database
Data
ID :1
NAME :Janhavi Raut
EMAIL :jau04@gmail.com
COURSE :Bsc.IT
DELETE DATA
VIEW ALL DATA

- **Screen 1 :** Enter only Unique ID while deleting data
- **Screen 2 :** Toast Message is displayed after clicking on Delete Data Button
- **Screen 3 :** Error Message viewed in dialog box after clicking View All Data Button

  (Custom Error Message is displayed because we deleted all the data)



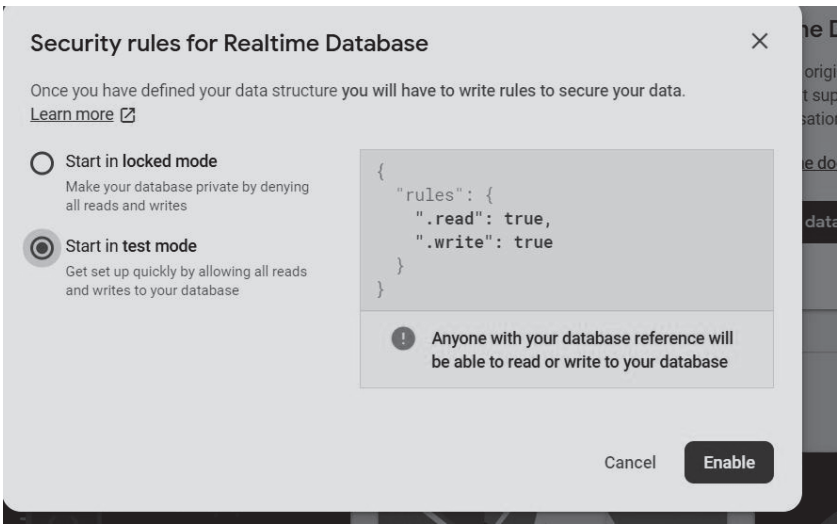
Practical 11

Aim : Create a suitable Android application to work with Firebase for storing and manipulating data.

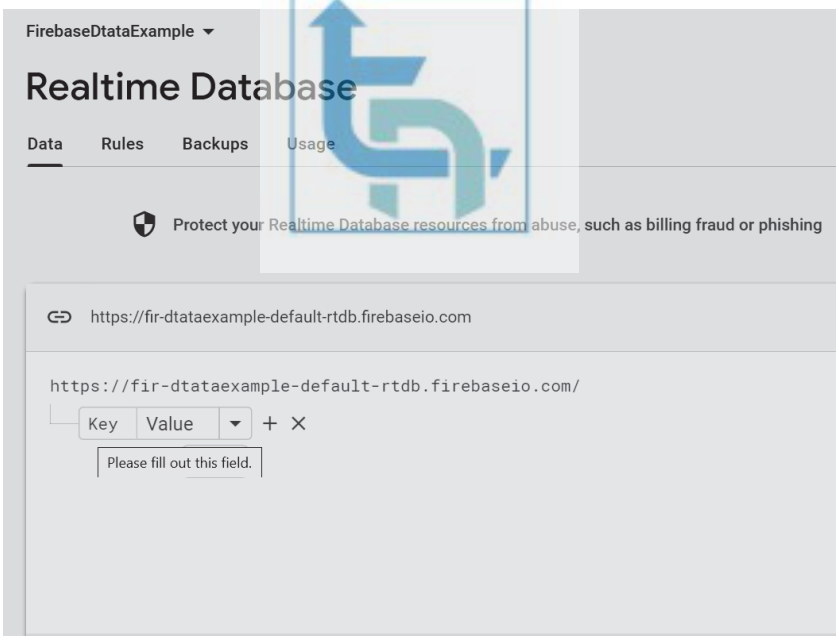
- **Step 1 : Create Realtime Database and data node in Firebase.**

  First go to Firebase and open your Firebase Project and click on Database and then click on Create Database under Realtime Database Heading.

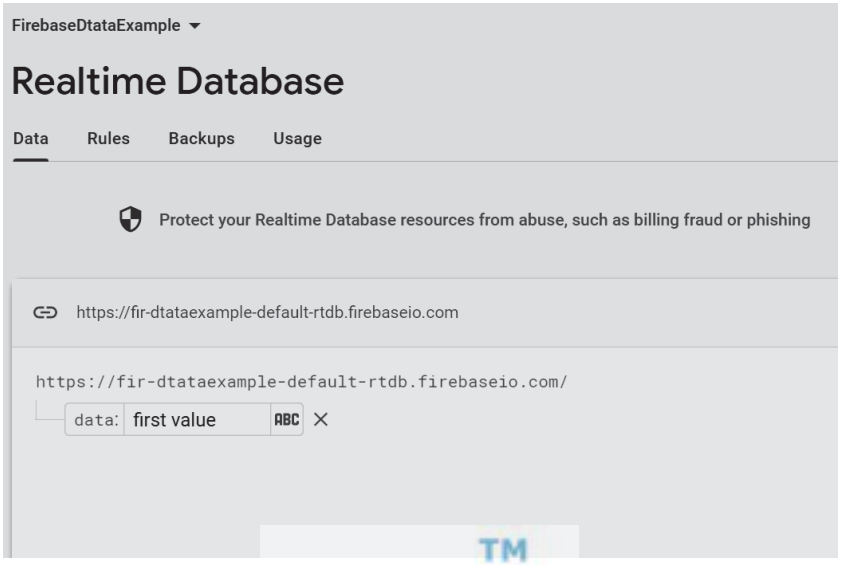
- Start in test mode during testing, which allows anyone who have your db reference to read and write data to your database. During real time development or production impose security rules by using firebase authentication according to your requirements.



- After creating Database, create a node named "data" and assign its value "first value".



- Copy the URL which we can see above key-value pair. We will need it in MainActivity.



- **Step 2 : Integrate and Code Firebase Realtime Database in your app**

- First add firebase realtime database library at app level build.gradle file implementation 'com.google.firebase:firebase-database:19.3.0'

- **Step 3: Add below line in "gradle.properties"**

```
android.useAndroidX=true
android.enableJetifier=true
```

- **Step 4 : Edit activity_main.xml as below**

```
<?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:orientation="vertical"
>
<TextView
android:id="@+id/data"
android:layout_width="match_parent"
android:layout_height="0dp"
android:layout_weight="1"
android:text="Value is : "
android:gravity="center"
/>
<EditText
```

```
android:id="@+id/changeData"
android:layout_width="match_parent"
android:layout_height="0dp"
android:layout_weight="1"
android:text=""
android:hint="Change Data"
android:gravity="center"
/>
<Button
android:layout_width="match_parent"
android:layout_height="0dp"
android:layout_weight="0.5"
android:text="change"
android:onClick="change"
/>
</LinearLayout>
```

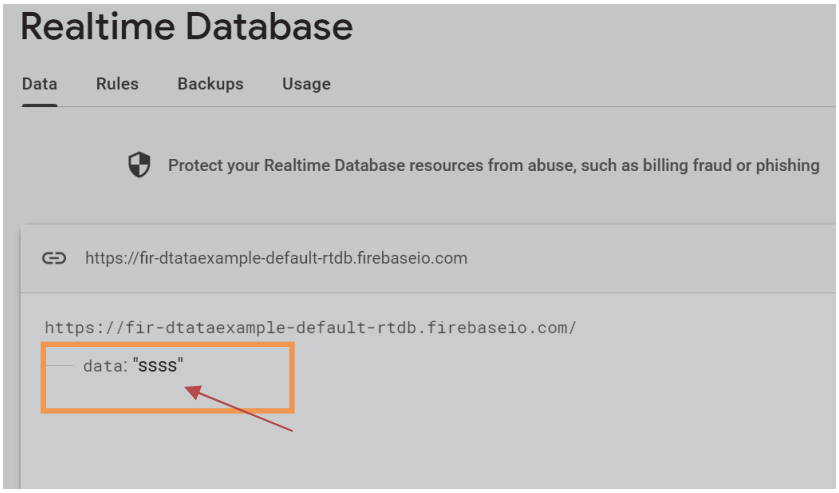
- **Step 5 : Edit MainActivity.kt as below**

```
package com.example.firebasedtataexample
import android.os.Bundle
import android.view.View
import android.widget.Button
import android.widget.EditText
import android.widget.TextView
import androidx.appcompat.app.AppCompatActivity
import com.google.firebase.database.*
class MainActivity : AppCompatActivity()
{
var db: FirebaseDatabase? = null
   var dataNodeRef: DatabaseReference? = null
   override fun onCreate(savedInstanceState: Bundle?)
   {
super.onCreate(savedInstanceState)
      setContentView(R.layout.activity_main)

val data = findViewById<TextView>(R.id.data)

db = FirebaseDatabase.getInstance("https://fir-dtataexample-default-
rtdb.firebaseio.com/")
dataNodeRef = db!!.getReference("data")
dataNodeRef!!.addValueEventListener(object : ValueEventListener
      {
override fun onDataChange(dataSnapshot: DataSnapshot)
         {
```

```
if (dataSnapshot.exists()) data.text = "Value is : " + dataSnapshot.value
}
override fun onCancelled(databaseError: DatabaseError) {}
      })
   }
fun change(view: View?)
   {
val changeData = findViewById<EditText>(R.id.changeData)
dataNodeRef!!.setValue(changeData.text.toString())
      changeData.setText("")
   }
}
```

☞ **Output**

When We run the app for first time.



☞ **After changing the data**

Lab Practical Ends...

❑❑❑