Program 2(i)-
Create an Android application to design screens using different layouts and UI including Button, Edittext, Textview, Radio Button

Program 2(i)A-Program For Button

First add the following program in activity_main.xml file

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

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Hello World!"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent" />
    <Button
        android:id="@+id/button"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:background="#4CAF50"
        android:paddingStart="10dp"
        android:paddingEnd="10dp"
        android:text="@string/btn"
        android:textColor="@android:color/background_light"
        android:textSize="24sp"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent" />
</androidx.constraintlayout.widget.ConstraintLayout>
```

Now After That Add the Following Program in Main Activity File i.e Mainactivity.kt file

```
package com.example.sycspractical2ia
import android.os.Bundle
import android.widget.Button
import android.widget.Toast
import androidx.appcompat.app.AppCompatActivity
```

```
import com.example.sycspractical2a.R

class MainActivity : AppCompatActivity() {

    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_main)

        // storing ID of the button
        // in a variable
        val button = findViewById<Button>(R.id.button)

        // operations to be performed
        // when user tap on the button
        button?.setOnClickListener()
        {
            // displaying a toast message
            Toast.makeText(this@MainActivity, R.string.message, Toast.LENGTH_LONG).show() }
    }
}
```

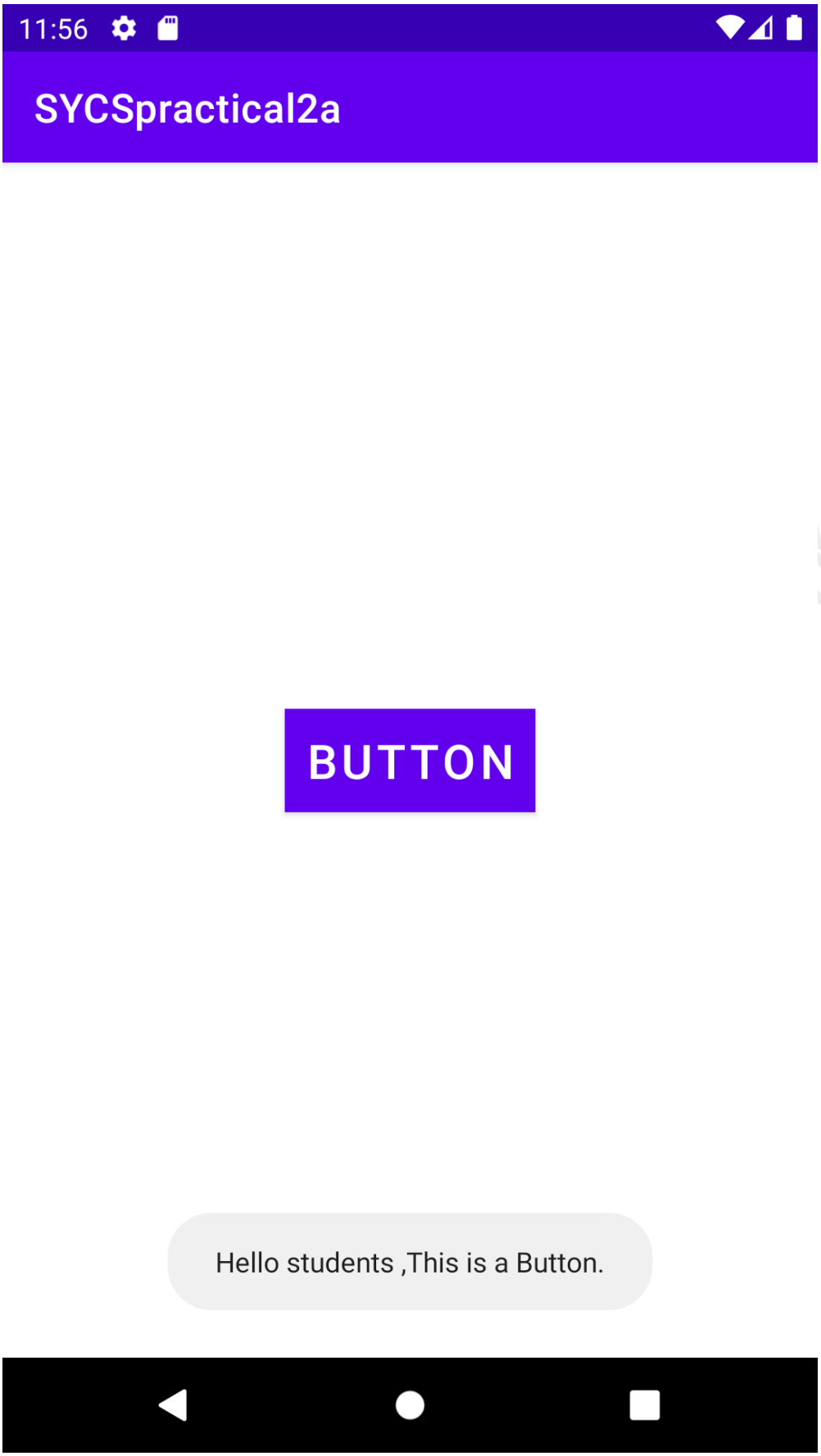
After that Add The Following Program in strings.xml file

```
<resources>
    <string name="app_name">SYCSpractical2a</string>
        <string name="btn">Button</string>
        <string name="message">Hello students ,This is a Button.</string>
</resources>
```

Now before running the Program Add The Following Code to the AndroidManifest.xml file Check before making changes if your androidmanisfest.xml looks like this don't make any unnecessary changes

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

    <application
        android:allowBackup="true"
        android:dataExtractionRules="@xml/data_extraction_rules"
        android:fullBackupContent="@xml/backup_rules"
        android:icon="@drawable/ic_launcher_background"
        android:label="@string/app_name"
        android:roundIcon="@drawable/ic_launcher_background"
        android:supportsRtl="true"
        android:theme="@style/Theme.SYCSPractical2ia"
        tools:targetApi="31">
        <activity
            android:name=".MainActivity"
            android:exported="true">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />

                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
    </application>

</manifest>
```

OUTPUT:-


11:56
SYCSpractical2a
BUTTON
Hello students ,This is a Button.

Program 2(i)B-Program For EditText
Add the following code in Activity_main.xml file -

```
<?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:orientation="vertical"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".MainActivity">

    <!--EditText with id editText-->

    <EditText
        android:id="@+id/editText"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_margin="16dp"
        android:hint="Input"
        android:inputType="text"/>

    <Button
        android:id="@+id/showInput"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_gravity="center_horizontal"
        android:text="show"
        android:backgroundTint="@color/colorPrimary"
        android:textColor="@android:color/white"
        />
</LinearLayout>
```

Now add the following Code to MainActivity.kt file:-

```
package com.example.sycspractical2ib
import android.annotation.SuppressLint
import android.os.Bundle
import android.support.v7.app.AppCompatActivity
import android.widget.Button
import android.widget.EditText
import android.widget.Toast
import com.example.sycspractical2aii.R
class MainActivity : AppCompatActivity() {

    @SuppressLint("MissingInflatedId")
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_main)

        // finding the button
        val showButton = findViewById<Button>(R.id.showInput)

        // finding the edit text
        val editText = findViewById<EditText>(R.id.editText)

        // Setting On Click Listener
        showButton.setOnClickListener {

            // Getting the user input
            val text = editText.text
```

```
            // Showing the user input
            Toast.makeText(this, text, Toast.LENGTH_SHORT).show()
        }
    }
}
```

Before Runing The Program add the following Code To AndroidManifest.xml File:-

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

    <application
        android:allowBackup="true"
        android:dataExtractionRules="@xml/data_extraction_rules"
        android:fullBackupContent="@xml/backup_rules"
        android:icon="@drawable/ic_launcher_background"
        android:label="@string/app_name"
        android:roundIcon="@drawable/ic_launcher_background"
        android:supportsRtl="true"
        android:theme="@style/Theme.SYCSPractical2ib"
        tools:targetApi="31">
        <activity
            android:name=".MainActivity"
            android:exported="true">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />

                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
    </application>

</manifest>
```

==========

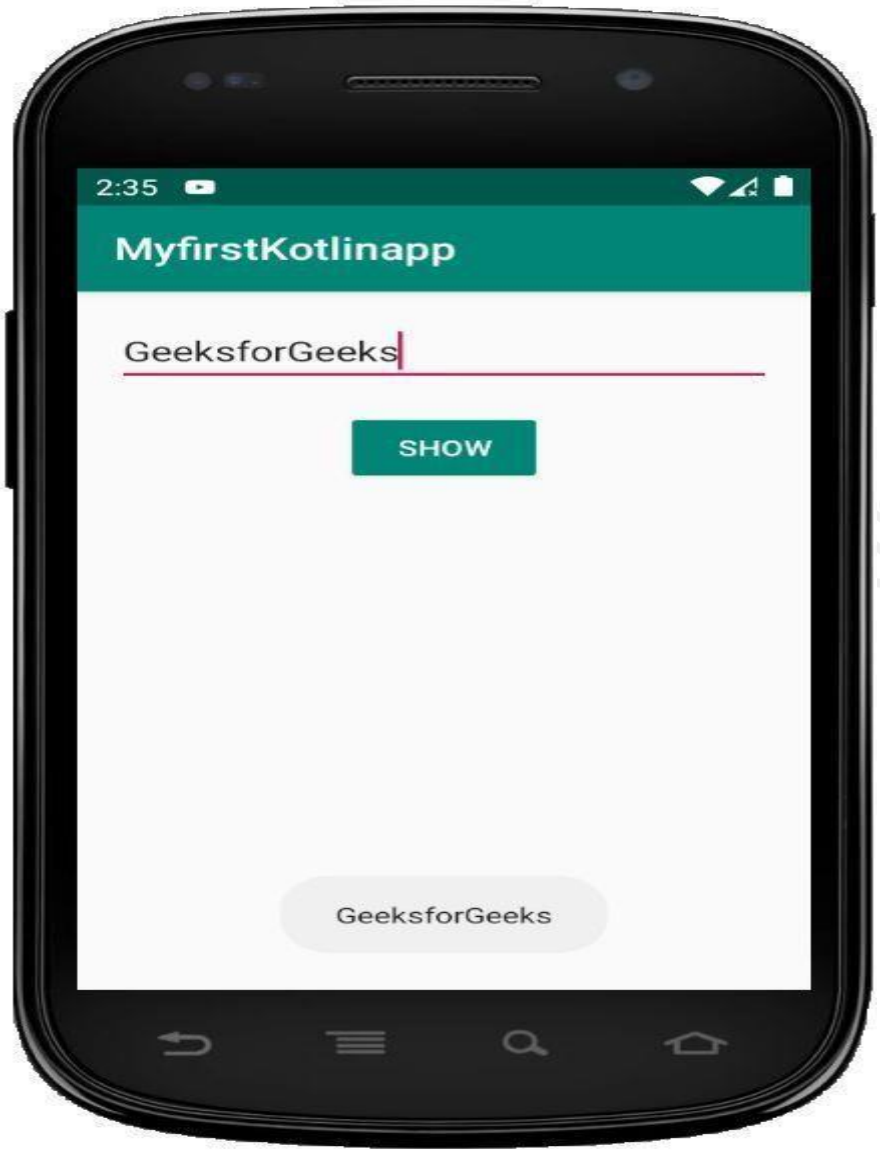
Possible error and its solution

You need to define the color resources in res/values/colors.xml to avoid the error.

Example :

- for @color/colorPrimary write the following code in res/values/colors.xml <color name="colorPrimary">#3F51B5</color>
- for @color/colorPrimaryDark write the following code in res/values/colors.xml <color name="colorPrimaryDark">#303F9F</color>
- for @color/colorAccent write the following code in res/values/colors.xml <color name="colorAccent">#FF4081</color>
- ==========

OUTPUT:-


2:35
MyfirstKotlinapp
GeeksforGeeks
SHOW
GeeksforGeeks

Program 2(i)c-Program For TextView

Add The Following Code To activity_main.xml File:-

```
<?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:orientation="vertical"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".MainActivity">

    <!—EditText with id editText→

    <TextView
        android:id="@+id/text_view_id"
        android:layout_height="wrap_content"
        android:layout_width="wrap_content"
        android:text="@string/text_view"
        android:textColor="#008000"
        android:textSize="40dp"
        android:textStyle="bold"/>
</LinearLayout>
```

Now After That Add The Following Code To MainActivity.kt File:-

```
package com.example.sycspractical2ic

import 25ndroid.appcompat.app.AppCompatActivity
import android.os.Bundle
import android.widget.TextView
import android.widget.Toast

class MainActivity : AppCompatActivity() {

    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_main)

        //accessing our 25ndroid25 from layout
        val textView = findViewById<TextView>(R.id.text_view_id)
        textView.setOnClickListener{ Toast.makeText(this@MainActivity,
            R.string.text_on_click, Toast.LENGTH_LONG).show() }
    }

}
```

Now Add The Following Code to strings.xml File:-

```
<resources>
    <string name="app_name">SYCSPractical2ic</string>
    <string name="text_view">www.profajaypashankar.com</string>
    <string name="text_on_click">COMPUTER SCIENCE PORTAL</string>
</resources>
```

Now Add The Following Code to AndroidManifest.xml File-

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

    <application
        android:allowBackup="true"
        android:dataExtractionRules="@xml/data_extraction_rules"
        android:fullBackupContent="@xml/backup_rules"
        android:icon="@mipmap/ic_launcher"
        android:label="@string/app_name"
        android:supportsRtl="true"
        android:theme="@style/Theme.SYCSPractical2ic"
        tools:targetApi="31">
        <activity
            android:name=".MainActivity"
            android:exported="true">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />

                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
    </application>

</manifest>
```

OUTPUT:-

You can run this code by adding show.button onclicklistener .

Program 2(i)d-Program For RadioButton

Add The Following Program To activity_main.xml File:-

```
<?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">

      <TextView
         android:layout_width="wrap_content"
         android:layout_height="wrap_content"
         android:text="@string/select_your_subject"
         android:textStyle="bold"
         android:layout_marginStart="10dp"
         android:textSize="20sp"/>

      <!—add RadioGroup which contain the many RadioButton-->
      <RadioGroup
         android:layout_marginTop="50dp"
         android:id="@+id/groupradio"
```

```
        android:layout_marginStart="10dp"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content">

            <!—In RadioGroup create the 1 Radio Button→
            <!—like this we will add some more Radio Button→
            <RadioButton
                android:layout_width="fill_parent"
                android:layout_height="wrap_content"
                android:id="@+id/radia_id1"
                android:text="@string/dbms"
                android:textSize="20sp"/>

            <RadioButton
                android:layout_width="fill_parent"
                android:layout_height="wrap_content"
                android:id="@+id/radia_id2"
                android:text="@string/c_c_programming"
                android:textSize="20sp"/>

            <RadioButton
                android:layout_width="fill_parent"
                android:layout_height="wrap_content"
                android:id="@+id/radia_id3"
                android:text="@string/data_structure"
                android:textSize="20sp"/>

            <RadioButton
                android:layout_width="fill_parent"
                android:layout_height="wrap_content"
                android:id="@+id/radia_id4"
                android:text="@string/algorithms"
                android:textSize="20sp"/>
    </RadioGroup>

    <!—add button For Submit the Selected item→
    <Button
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="@string/submit"
        android:id="@+id/submit"
        android:textStyle="bold"
        android:textSize="20sp"
        android:layout_marginTop="200dp"
        android:layout_marginStart="180dp"
        />

    <!—add clear button for clear the selected item→
    <Button
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="@string/clear"
        android:id="@+id/clear"
        android:textSize="20sp"
        android:textStyle="bold"
        android:layout_marginTop="200dp"
        android:layout_marginStart="20dp"
        />

</RelativeLayout>
```

Now After That Add The Following Program To MainActivity.kt File:-

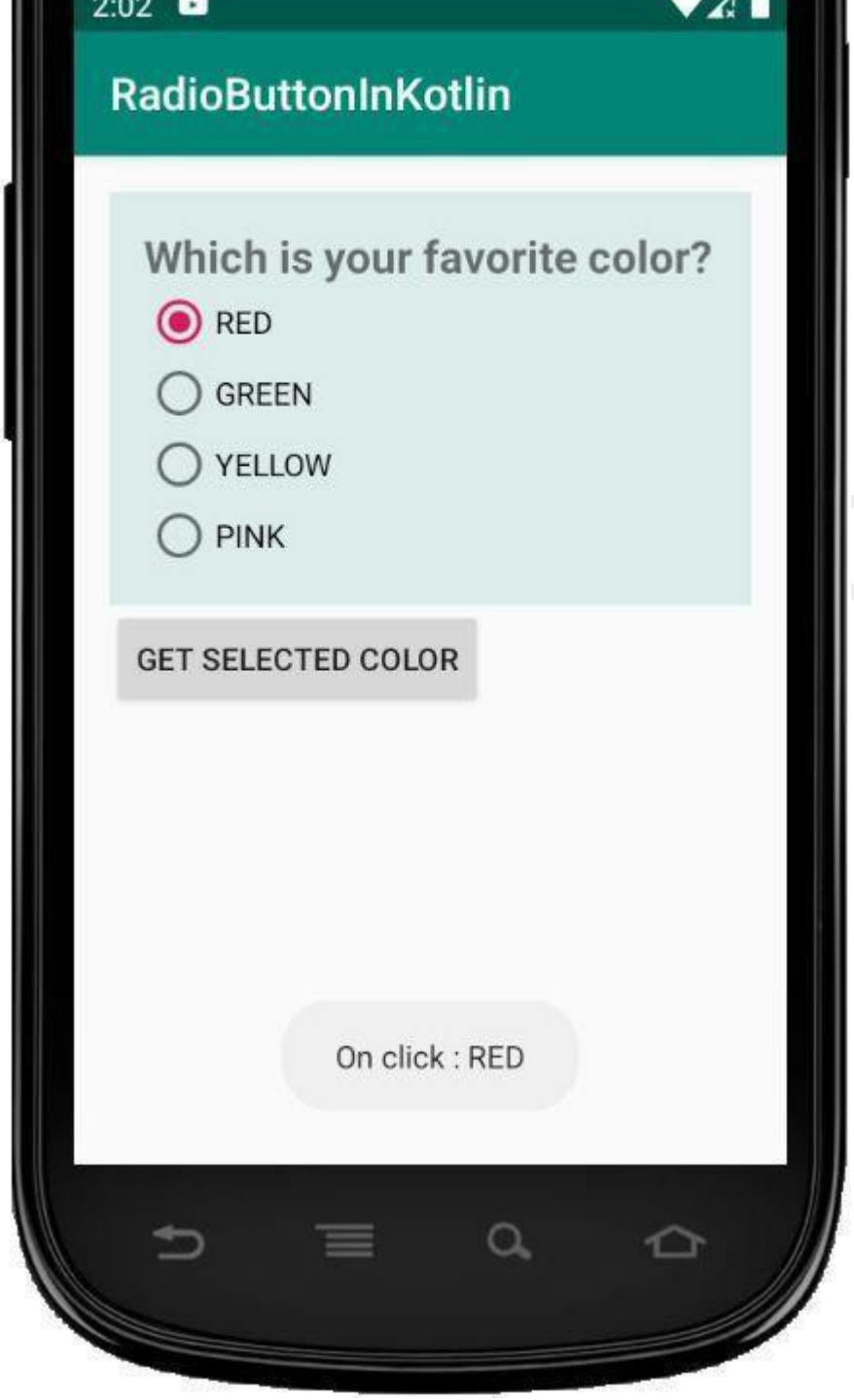
```
package com.example.sycspractical2id

import android.os.Bundle
import android.support.v7.app.AppCompatActivity
import android.view.View
import android.widget.Button
import android.widget.RadioButton
import android.widget.RadioGroup
import android.widget.Toast


class MainActivity : AppCompatActivity() {
    // Define the object for Radio Group,
    // Submit and Clear buttons
    private var radioGroup: RadioGroup? = null
    private var submit: Button? = null
    private var clear: Button? = null
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_main)

        // Bind the components to their respective objects
        // by assigning their IDs
        // with the help of findViewById() method
        this.submit = findViewById<View>(R.id.submit) as Button
        this.clear = findViewById<View>(R.id.clear) as Button
        this.radioGroup = findViewById<View>(R.id.groupradio) as RadioGroup

        // Uncheck or reset the radio buttons initially
        radioGroup!!.clearCheck()

        // Add the Listener to the RadioGroup
        radioGroup!!.setOnCheckedChangeListener { group, checkedId ->
            // The flow will come here when
            // any of the radio buttons in the radioGroup
            // has been clicked
            // Check which radio button has been clicked
            // Get the selected Radio Button
            val radioButton = group
                .findViewById<View>(checkedId) as RadioButton
        }

        // Add the Listener to the Submit Button
        submit!!.setOnClickListener {
            // When submit button is clicked,
            // Ge the Radio Button which is set
            // If no Radio Button is set, -1 will be returned
            val selectedId = radioGroup!!.checkedRadioButtonId
            if (selectedId == -1) {
                Toast.makeText(
                    this@MainActivity,
                    "No answer has been selected",
                    Toast.LENGTH_SHORT
                )
                    .show()
            } else {
                val radioButton = radioGroup!!
                    .findViewById<View>(selectedId) as RadioButton

                // Now display the value of selected item
                // by the Toast message
                Toast.makeText(
                    this@MainActivity,
```

```
                radioButton.text,
                Toast.LENGTH_SHORT
            )
                .show()
        }
    }

    // Add the Listener to the Submit Button
    clear!!.setOnClickListener { // Clear RadioGroup
        // i.e. reset all the Radio Buttons
        radioGroup!!.clearCheck()
    }
  }
}
```

Add The Following Program in strings.xml File:-

```
<resources>
    <string name="app_name">SYCSPractical2id</string>
    <string name="select_your_subject">Select your Subject ?</string>
    <string name="dbms">DBMS</string>
    <string name="c_c_programming">C/C++ Programming</string>
    <string name="data_structure">Data Structure</string>
    <string name="algorithms">Algorithms</string>
    <string name="submit">Submit</string>
    <string name="clear">Clear</string>
</resources>
```

OUTPUT:-


2:02
RadioButtonInKotlin
Which is your favorite color?
RED
GREEN
YELLOW
PINK
GET SELECTED COLOR
On click : RED

Program 2(ii)-
Write an android application demonstrating response to event/user interaction for
a. Checkbox
b. Radio button
c. Button
d. Spinner

Program 2(ii)a- Program For Checkbox
Add rboto font sketch file in asset folder in app/src/main
https://fonts.google.com/specimen/Roboto

First Of All Add The Following Program In activity_main.xml file:-

```
<?xml version="1.0" encoding="utf-8"?>
<!—suppress ALL →
<32ndroid.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"
    android:background="#168BC34A"
    tools:context=".MainActivity"
    tools:ignore="MissingClass">

    <TextView
        android:id="@+id/textView"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:fontFamily="@font/roboto"
        android:text="@string/Heading"
        android:textAlignment="center"
        android:textColor="@android:color/holo_green_dark"
        android:textSize="36sp"
        android:textStyle="bold"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        app:layout_constraintVertical_bias="0.17000002" />

    <LinearLayout
        android:id="@+id/32ndroid32_container"
        android:layout_width="0dp"
        android:layout_height="wrap_content"
        android:orientation="vertical"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/textView"
        app:layout_constraintVertical_bias="0.18">

        <CheckBox
            android:id="@+id/32ndroid32"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:fontFamily="@font/roboto"
            android:text="@string/checkBox1_text"
            android:textSize="18sp"
            android:padding="7dp"/>
```

```
        <CheckBox
            android:id="@+id/checkBox2"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:fontFamily="@font/roboto"
            android:text="@string/checkBox2_text"
            android:textSize="18sp"
            android:padding="7dp"/>

        <CheckBox
            android:id="@+id/checkBox3"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:fontFamily="@font/roboto"
            android:text="@string/checkBox3_text"
            android:textSize="18sp"
            android:padding="7dp"/>

        <CheckBox
            android:id="@+id/checkBox4"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:fontFamily="@font/roboto"
            android:text="@string/checkBox4_text"
            android:textSize="18sp"
            android:padding="7dp"/>

        <CheckBox
            android:id="@+id/checkBox5"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:fontFamily="@font/roboto"
            android:text="@string/checkBox5_text"
            android:textSize="18sp"
            android:padding="7dp"/>
    </LinearLayout>

    <Button
        android:id="@+id/submitButton"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:background="#AB4CAF50"
        android:fontFamily="@font/roboto"
        android:text="@string/submitButton"
        android:textSize="18sp"
        android:textStyle="bold"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/33ndroid33_container"
        app:layout_constraintVertical_bias="0.23000002" />
</33ndroid.constraintlayout.widget.ConstraintLayout>
```

Now Add The Following Program In MainActivity.kt File:-

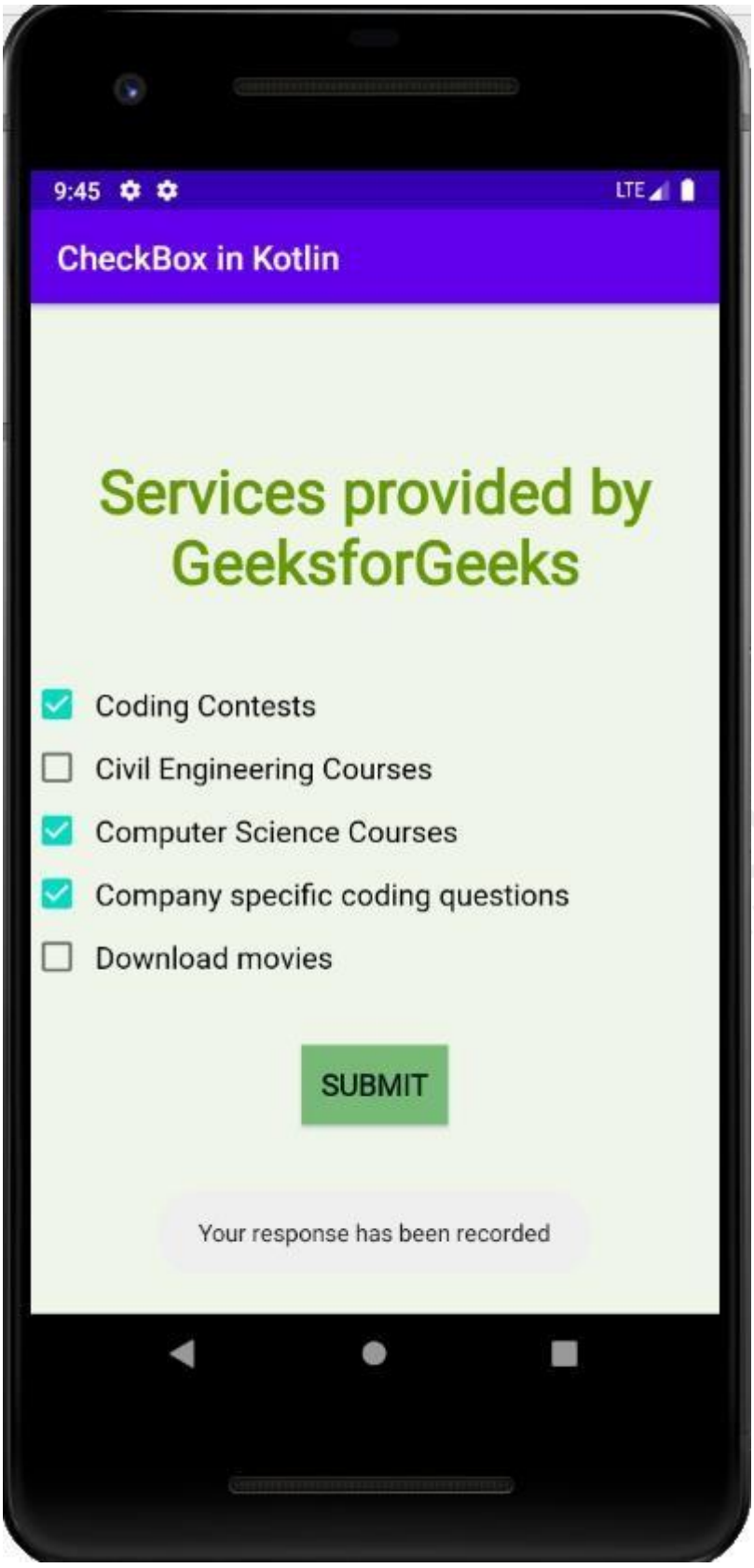
```
package com.example.sycspractical2iia

import android.os.Build.VERSION_CODES.R
import android.os.Bundle
import android.support.v7.app.AppCompatActivity
import android.widget.Button
```

```
import android.widget.Toast

class MainActivity : AppCompatActivity() {
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_main)

        // Assigning id of the submit button
        val button : Button = findViewById(R.id.submitButton)

        // Actions to be performed
        // when Submit button is clicked
        button.setOnClickListener{

            // Display toast message
            Toast.makeText(applicationContext,
                "Your response has been recorded", Toast.LENGTH_LONG).show()
        }
    }
}
```

Now Add The Following Program in strings.xml File-

```
<resources>
    <string name="app_name">SYCSPractical2iia</string>
    <string name="Heading">Services provided by GeeksforGeeks</string>
    <string name="checkBox1">Coding contests</string>
    <string name="checkBox2_text">Civil Engineering Courses</string>
    <string name="checkBox1_text">Coding Contests</string>
    <string name="checkBox3_text">Computer Science Courses</string>
    <string name="checkBox4_text">Company specific coding questions</string>
    <string name="checkBox5_text">Download movies</string>
    <string name="submitButton">SUBMIT</string>
</resources>
```

OUTPUT:-


9:45
LTE
CheckBox in Kotlin
Services provided by GeeksforGeeks
Coding Contests
Civil Engineering Courses
Computer Science Courses
Company specific coding questions
Download movies
SUBMIT
Your response has been recorded

Program 2(ii)b -Program For RadioButton

Add The Following Program in activity_main.xml File:-

```
<?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:id="@+id/root_layout"
   android:layout_width="match_parent"
   android:layout_height="match_parent"
   android:orientation="vertical"
   android:padding="16dp">
   <RadioGroup
      android:id="@+id/radio_group"
      android:layout_width="match_parent"
      android:layout_height="wrap_content"
      android:background="#dbeceb"
      android:padding="15dp">

      <TextView
         android:id="@+id/title"
         android:layout_width="match_parent"
         android:layout_height="wrap_content"
         android:text="Which is your favorite color?"
         android:textStyle="bold"
         android:textSize="20sp"/>

      <RadioButton
         android:id="@+id/red"
         android:layout_width="wrap_content"
         android:layout_height="wrap_content"
         android:text="RED"
         android:onClick="radio_button_click"/>

      <RadioButton
         android:id="@+id/green"
         android:layout_width="wrap_content"
         android:layout_height="wrap_content"
         android:text="GREEN"
         android:onClick="radio_button_click"/>

      <RadioButton
```

```
        android:id="@+id/yellow"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="YELLOW"
        android:onClick="radio_button_click"/>

    <RadioButton
        android:id="@+id/pink"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="PINK"
        android:onClick="radio_button_click"/>
  </RadioGroup>

  <Button
    android:id="@+id/button"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Get Selected Color"/>
</LinearLayout>
```

Now Add The Following Program in MainActivity.kt file:-

```
package com.example.sycspractical2iib

import 37ndroid.appcompat.app.AppCompatActivity
import android.os.Bundle
import android.view.View
import android.widget.*
import android.widget.RadioGroup


private val Nothing?.checkedRadioButtonId: Int
  get() {
    TODO("Not yet implemented")
  }

class MainActivity : AppCompatActivity() {

  override fun onCreate(savedInstanceState: Bundle?) {
    super.onCreate(savedInstanceState)
    setContentView(R.layout.activity_main)

    // Get radio group selected item using on checked change listener
    val radio_group = null
    radio_group.setOnCheckedChangeListener(
      RadioGroup.OnCheckedChangeListener { group, checkedId ->
        val radio: RadioButton = findViewById(checkedId)
        Toast.makeText(applicationContext," On checked change :"+
            " ${radio.text}",
          Toast.LENGTH_SHORT).show()
      })
    // Get radio group selected status and text using button click event
    val button = null
    button.setOnClickListener{
      // Get the checked radio button id from radio group
      var id: Int = radio_group.checkedRadioButtonId
      if (id!=-1){ // If any radio button checked from radio group
        // Get the instance of radio button using id
        val radio:RadioButton = findViewById(id)
        Toast.makeText(applicationContext,"On button click :" +
            " ${radio.text}",
          Toast.LENGTH_SHORT).show()
      }else{
```

```
            // If no radio button checked in this radio group
            Toast.makeText(applicationContext,"On button click :" +
                    " nothing selected",
                Toast.LENGTH_SHORT).show()
        }
    }
}
// Get the selected radio button text using radio button on click listener
fun radio_button_click(view: View){
    // Get the clicked radio button instance
    val radio_group = null
    val radio: RadioButton = findViewById(radio_group.checkedRadioButtonId)
    Toast.makeText(applicationContext,"On click : ${radio.text}",
        Toast.LENGTH_SHORT).show()
}
}

private fun Nothing?.setOnClickListener(function: () -> Unit) {
    TODO("Not yet implemented")
}

private fun Nothing?.setOnCheckedChangeListener(onCheckedChangeListener:
RadioGroup.OnCheckedChangeListener) {
    TODO("Not yet implemented")
}
```

Now Add The Following Program in strings.xml File:-

```
<resources>
    <string name="app_name">SYCSPractical2iib</string>
    <string name="checked">checked</string>
    <string name="unchecked">unchecked</string>
</resources>
```

OUTPUT:-

Program 2(ii)c-Program For Button

Add The Following Code To activity_main.xml File:-

```
<?xml version="1.0" encoding="utf-8"?>
<!—suppress ALL →
<39ndroid.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"
   android:background="#168BC34A"
   tools:context=".MainActivity"
   tools:ignore="MissingClass">

   <!—Button added in the activity →
   <Button
      android:id="@+id/button"
      android:layout_width="wrap_content"
      android:layout_height="wrap_content"
      android:background="#4CAF50"
      android:paddingStart="10dp"
      android:paddingEnd="10dp"
```

```
        android:text="@string/btn"
        android:textColor="@android:color/background_light"
        android:textSize="24sp"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent" />

</40ndroid.constraintlayout.widget.ConstraintLayout>
```

Now Add The Following Code To MainActivity.kt file:-

```
package com.example.sycspractical2iic

import android.os.Bundle
import android.support.v7.app.AppCompatActivity
import android.widget.Button
import android.widget.Toast

class MainActivity : AppCompatActivity() {

    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_main)

        // storing ID of the button
        // in a variable
        val button = findViewById<Button>(R.id.button)

        // operations to be performed
        // when user tap on the button
        button?.setOnClickListener()
        {
            // displaying a toast message
            Toast.makeText(this@MainActivity, R.string.message, Toast.LENGTH_LONG).show() }
    }
}
```

After That, Add The Following Code To strings.xml File:-

```
<resources>
    <string name="app_name">SYCSPractical2iic</string>
    <string name="btn">Button</string>
    <string name="message">Hello Geeks!! This is a Button.</string>
</resources>
```

OUTPUT:-


10:54
GfG | Button In Kotlin
BUTTON
GeeksforGeeks


10:54
GfG | Button In Kotlin
BUTTON
Hello Geeks!! This is a Button.

Program 2(ii)d -Program For Spinner

Firstly, Add The Following Code to activity_main.xml File:-

```
<?xml version="1.0" encoding="utf-8"?>
<android.support.constraint.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">

    <Spinner
        android:id="@+id/coursesspinner"
        android:layout_height="50dp"
        android:layout_width="160dp"
        android:layout_marginEnd="10dp"
        android:layout_marginStart="10dp"
        android:layout_marginBottom="10dp"
        android:layout_marginTop="10dp"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent"/>

</android.support.constraint.ConstraintLayout>
```

Now Add The Following Code To MainActivity.kt File:-

```
package com.example.sycspractical2iid

import android.os.Bundle
import android.support.v7.app.AppCompatActivity
import android.view.View
import android.widget.AdapterView
import android.widget.AdapterView.OnItemSelectedListener
import android.widget.ArrayAdapter
import android.widget.Spinner
import android.widget.Toast


class MainActivity : AppCompatActivity(), OnItemSelectedListener {
    // create array of Strings
    // and store name of courses
    var courses = arrayOf<String?>("C", "Data structures",
        "Interview prep", "Algorithms",
        "DSA with java", "OS")

    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_main)

        // Take the instance of Spinner and
        // apply OnItemSelectedListener on it which
        // tells which item of spinner is clicked
        val spin = findViewById<Spinner>(R.id.coursesspinner)
        spin.onItemSelectedListener = this

        // Create the instance of ArrayAdapter
```

```
        // having the list of courses
        val ad: ArrayAdapter<*> = ArrayAdapter<Any?>(
            this,
            android.R.layout.simple_spinner_item,
            courses)

        // set simple layout resource file
        // for each item of spinner
        ad.setDropDownViewResource(
            android.R.layout.simple_spinner_dropdown_item)

        // Set the ArrayAdapter (ad) data on the
        // Spinner which binds data to spinner
        spin.adapter = ad
    }

    override fun onItemSelected(parent: AdapterView<*>?,
                                view: View, position: Int,
                                id: Long) {
        // make toastof name of course
        // which is selected in spinner
        Toast.makeText(applicationContext,
            courses[position],
            Toast.LENGTH_LONG)
            .show()
    }

    override fun onNothingSelected(parent: AdapterView<*>?) {}
}
```

OUTPUT:-

Program 3:-

Program 3(i)-Create an application to create Image Flipper and Image Gallery. On click on the image display the information about the image.

Working with the activity_main.xml file
Navigate to the app > res > layout > activity_main.xml and add the below code to that file. Below is the code for the activity_main.xml file. Comments are added inside the code to understand the code in more detail.

```
<?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:orientation="vertical"
    tools:context=".MainActivity">
    <!--on below line we are adding view pager -->
    <androidx.viewpager.widget.ViewPager
        android:id="@+id/idViewPager"
        android:layout_width="300dp"
        android:layout_height="300dp"
        android:layout_centerInParent="true"
        android:layout_gravity="center"
        android:contentDescription="hello students enjoying android "
```

```
        android:layout_margin="10dp" />
</RelativeLayout>
```

Creating a layout file for ImageView in View Pager

Navigate to the app > res > layout > Right-click on it > New > Layout Resource file and specify the name as image_slider_item. Add the below code to it. Comments are added in the code to get to know in detail.

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

    <!--on below line we are creating an image view-->
    <ImageView
        android:id="@+id/idIVImage"
        android:layout_width="200dp"
        android:layout_height="200dp"
        android:layout_centerInParent="true" />

</RelativeLayout>
```

Creating a new java class for the adapter of our ViewPager

Navigate to the app > java > your app's package name > Right-click on it > New > Java/Kotlin class and name it as ViewPagerAdapter and add the below code to it. Comments are added in the code to get to know in detail.

```
package com.example.sycs3iimageflipper
import android.content.Context
import android.view.LayoutInflater
import android.view.View
import android.view.ViewGroup
import android.widget.ImageView
import android.widget.RelativeLayout
import androidx.viewpager.widget.PagerAdapter
import java.util.*

class ViewPagerAdapter(val context: Context, val imageList: List<Int>) : PagerAdapter() {
    // on below line we are creating a method
    // as get count to return the size of the list.
    override fun getCount(): Int {
        return imageList.size
    }

    // on below line we are returning the object
    override fun isViewFromObject(view: View, `object`: Any): Boolean {
        return view === `object` as RelativeLayout
    }

    // on below line we are initializing
    // our item and inflating our layout file
    override fun instantiateItem(container: ViewGroup, position: Int): Any {
        // on below line we are initializing
        // our layout inflater.
        val mLayoutInflater =
            context.getSystemService(Context.LAYOUT_INFLATER_SERVICE) as LayoutInflater

        // on below line we are inflating our custom
        // layout file which we have created.
        val itemView: View = mLayoutInflater.inflate(R.layout.image_slider_item, container, false)

        // on below line we are initializing
        // our image view with the id.
        val imageView: ImageView = itemView.findViewById<View>(R.id.idIVImage) as ImageView
```

```
        // on below line we are setting
        // image resource for image view.
        imageView.setImageResource(imageList.get(position))

        // on the below line we are adding this
        // item view to the container.
        Objects.requireNonNull(container).addView(itemView)

        // on below line we are simply
        // returning our item view.
        return itemView
    }

    // on below line we are creating a destroy item method.
    override fun destroyItem(container: ViewGroup, position: Int, `object`: Any) {
        // on below line we are removing view
        container.removeView(`object` as RelativeLayout)
    }
}
```

Adding images to the drawable folder

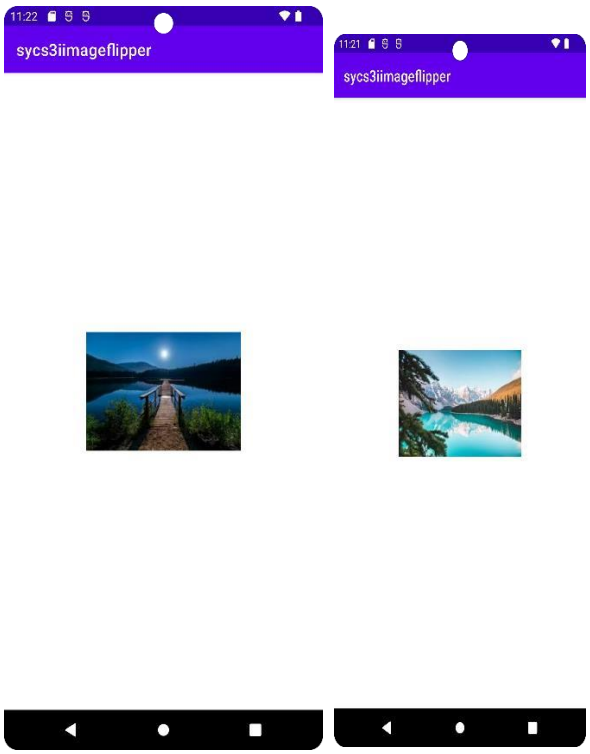
Select the images which you want to add copy them Navigate to app > res > drawable and right-click on it. Simply paste it and add all the images to the drawable folder.

Working with the MainActivity.kt file

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

```
package com.example.sycs3iimageflipper
import android.os.Bundle
import androidx.appcompat.app.AppCompatActivity
import androidx.viewpager.widget.ViewPager

class MainActivity : AppCompatActivity() {
    // on below line we are creating variable for view pager,
    // viewpager adapter and the image list.
    lateinit var viewPager: ViewPager
    lateinit var viewPagerAdapter: ViewPagerAdapter
    lateinit var imageList: List<Int>

    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_main)

        // initializing variables
        // of below line with their id.
        viewPager = findViewById(R.id.idViewPager)

        // on below line we are initializing
        // our image list and adding data to it.
        imageList = ArrayList<Int>()
        imageList = imageList + R.drawable.img1
        imageList = imageList + R.drawable.img2
        imageList = imageList + R.drawable.img3
        imageList = imageList + R.drawable.img4
        imageList = imageList + R.drawable.img5

        // on below line we are initializing our view
        // pager adapter and adding image list to it.
        viewPagerAdapter = ViewPagerAdapter(this@MainActivity, imageList)

        // on below line we are setting
        // adapter to our view pager.
        viewPager.adapter = viewPagerAdapter
```

```
    }
}
```
OUTPUT:



Just click on the screen your image changes

Second Method

Firstly Add The Following Code To activity_main.xml File-

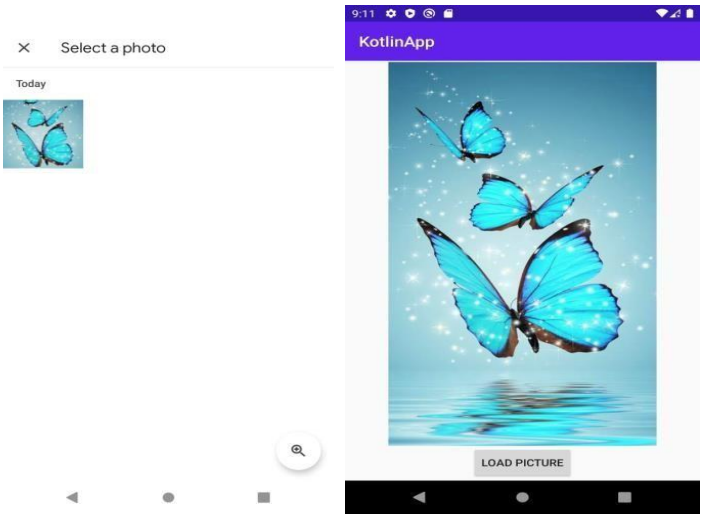
```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:orientation="vertical"
    android:padding="2dp">
    <ImageView
        android:id="@+id/imageView"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_weight="1" />
    <Button
        android:id="@+id/buttonLoadPicture"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_gravity="center"
        android:layout_weight="0"
        android:text="Load Picture" />
</LinearLayout>
```

Now Add The Following Code To MainActivity.kt File:-

```
package com.example.sycspractical3i

import android.content.Intent
import android.net.Uri
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() {
    lateinit var imageView: ImageView
    lateinit var button: Button
    private val pickImage = 100
    private var imageUri: Uri? = null
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_main)
        title = "KotlinApp"
        imageView = findViewById(R.id.imageView)
        button = findViewById(R.id.buttonLoadPicture)
        button.setOnClickListener {
            val gallery = Intent(Intent.ACTION_PICK, MediaStore.Images.Media.INTERNAL_CONTENT_URI)
            startActivityForResult(gallery, pickImage)
        }
    }
    override fun onActivityResult(requestCode: Int, resultCode: Int, data: Intent?) {
        super.onActivityResult(requestCode, resultCode, data)
        if (resultCode == RESULT_OK && requestCode == pickImage) {
            imageUri = data?.data
            imageView.setImageURI(imageUri)
        }
    }
}
```

OUTPUT:-



Program 3(ii)-
Create an application to use Gridview for shopping cart application

Add the following code to res/layout/activity_main.xml.
Example
```
<?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">
<GridView
   android:id="@+id/gridView"
   android:layout_width="match_parent"
   android:layout_height="match_parent"
   android:numColumns="2" />
</RelativeLayout>
```
Step 3 − Add the following code to src/MainActivity.kt
```
import android.os.Bundle
import android.widget.AdapterView.OnItemClickListener
import android.widget.GridView
import android.widget.Toast
import androidx.appcompat.app.AppCompatActivity
class MainActivity : AppCompatActivity() {
   lateinit var gridView: GridView
   private var playerNames = arrayOf("Cristiano Ronaldo", "Joao Felix", "Bernado Silva", "Andre    Silve",
"Bruno Fernandez", "William Carvalho", "Nelson Semedo", "Pepe", "Rui Patricio")
   private var playerImages = intArrayOf(R.drawable.ronaldo, R.drawable.felix, R.drawable.bernado,
   R.drawable.andre,
   R.drawable.bruno, R.drawable.carvalho, R.drawable.semedo, R.drawable.pepe, R.drawable.patricio)
   override fun onCreate(savedInstanceState: Bundle?) {
      super.onCreate(savedInstanceState)
```

```
    setContentView(R.layout.activity_main)
    title = "KotlinApp"
    gridView = findViewById(R.id.gridView)
    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.rowitem, 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>
```

Step 6 − Add the following code to androidManifest.xml

```
<?xml version="1.0" encoding="utf-8"?>
<manifest                                    xmlns:android="http://schemas.android.com/apk/res/android"
package="com.example.q11">
  <application
    android:allowBackup="true"
    android:icon="@mipmap/ic_launcher"
    android:label="@string/app_name"
    android:roundIcon="@mipmap/ic_launcher_round"
    android:supportsRtl="true"
    android:theme="@style/AppTheme">
    <activity android:name=".MainActivity">
      <intent-filter>
        <action android:name="android.intent.action.MAIN" />
        <category android:name="android.intent.category.LAUNCHER" />
      </intent-filter>
    </activity>
  </application>
</manifest>
```

For above code addd images to res/drawable folder make sure all images are named as refrence given in MainActivity.kt

OUTPUT:-

==========

Creating shopping cart is bit lengthy and pathetic most of the files given online are old API supported if you can run that it will generates lots of errors for simplicity in this code we have added GridView and Images into that if you can add extra methods and files like shopping cart then you can refer lin k given below but its not working as code was written in 2019 .still for more exploration refer this code

https://pusher.com/tutorials/shopping-cart-kotlin-part-1/

==========

Program 4:-

Program 4(i)a-Create an Android application to demonstrate implicit and explicit intents

1)Implicit Intent
First Add The Above Code to activity_main.xml File-

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

```
    tools:context=".MainActivity">

    <EditText
        android:id="@+id/editText"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent" />

    <Button
        android:id="@+id/btn"
        android:text="Search"
        android:onClick="search"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/editText" />


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

Now Add The Following Program to MainActivity.kt file-

```
package com.example.sycspractical4i

import android.content.Intent
import android.net.Uri
import android.os.Bundle
import android.support.v7.app.AppCompatActivity
import android.widget.EditText


class MainActivity : AppCompatActivity() {

    lateinit var editText: EditText

    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_main)

        editText = findViewById(R.id.editText)
    }

    fun search() {
        val url = editText.text.toString()
        val urlIntent = Intent(Intent.ACTION_VIEW, Uri.parse(url))
        startActivity(urlIntent)
    }
}
```

OUTPUT:-

Program 4(i)b-
EXPLICIT INTENT

Firstly Add This Program to activity_main.xml File:-

```
<?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="com.example.explicitintentexample.MainActivity">
<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    android:gravity="center">

    <Button
        android:id="@+id/button"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Launch Second Activity"/>

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

Now Add The Following Program To MainActivity.kt File:-

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

class MainActivity : AppCompatActivity() {

    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_main)

        val button = findViewById<Button>(R.id.button)
        button.setOnClickListener {
            val intent = Intent(this, SecondActivity::class.java)
            startActivity(intent)
        }
    }
}
```

Now Create a New File named activity_main2.xml and add the Following Code:-

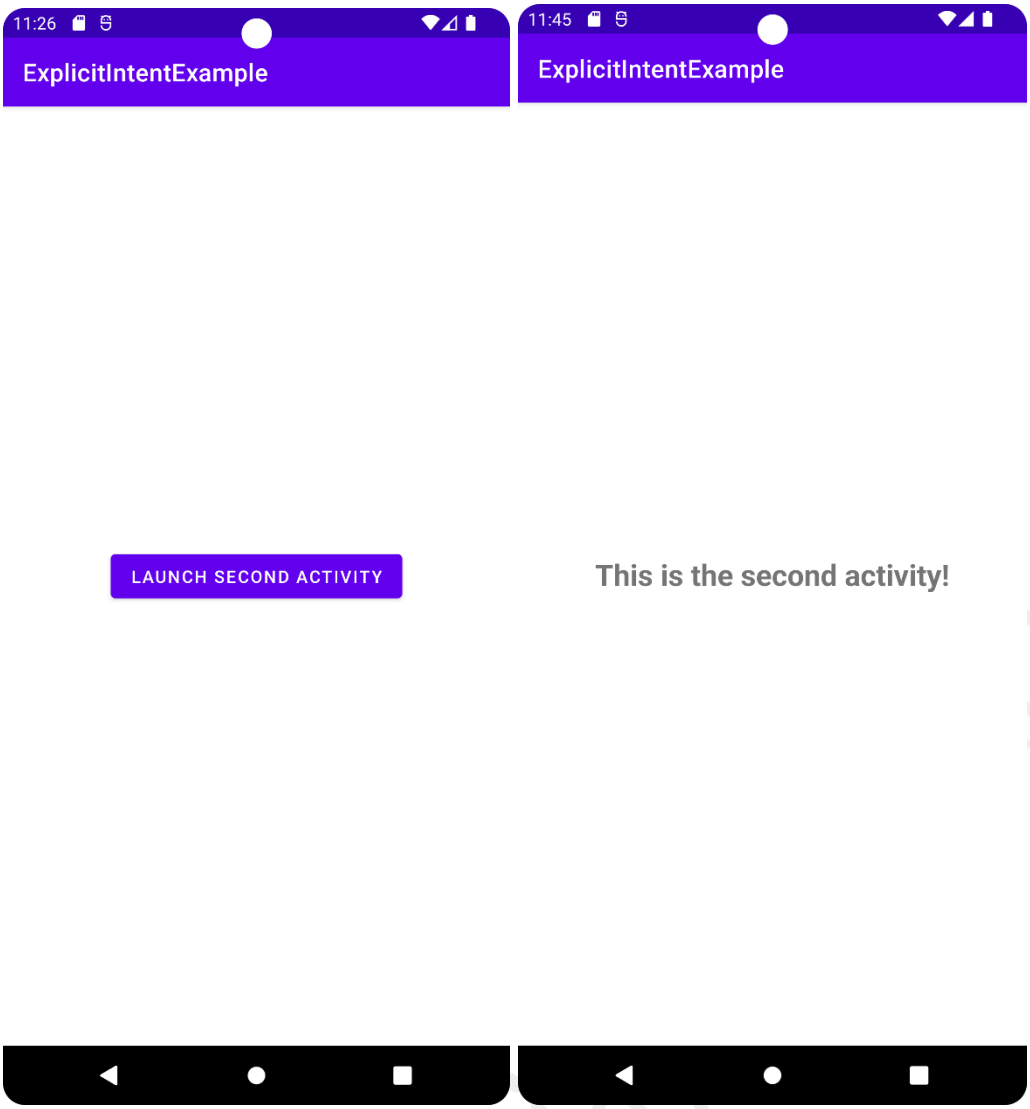
```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context="com.example.explicitintentexample.SecondActivity">
<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    android:gravity="center">

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="This is the second activity!"
        android:textSize="24sp"
        android:textStyle="bold"/>

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

Now After That Add The Following Code To MainActivity2.kt file-

```
package com.example.explicitintentexample

import androidx.appcompat.app.AppCompatActivity
import android.os.Bundle
class SecondActivity : AppCompatActivity() {

    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_main2)
    }
}
```

NOTE: class name of our second activity file is SecondActivity not MainActivity2 be careful while giving reference in Androidmanifest.xml

OUTPUT:-



Program 4(ii)-Create an application to demonstrate shared preferences

Firstly Add The Following Code to activity_main.xml File:-

```
<?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/textview"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_centerHorizontal="true"
        android:layout_marginTop="32dp"
        android:text="Shared Preferences Demo"
        android:textColor="@android:color/black"
        android:textSize="24sp" />

    <!--EditText to take the data from the user and save the data in SharedPreferences-->
    <EditText
        android:id="@+id/edit1"
        android:layout_width="match_parent"
```

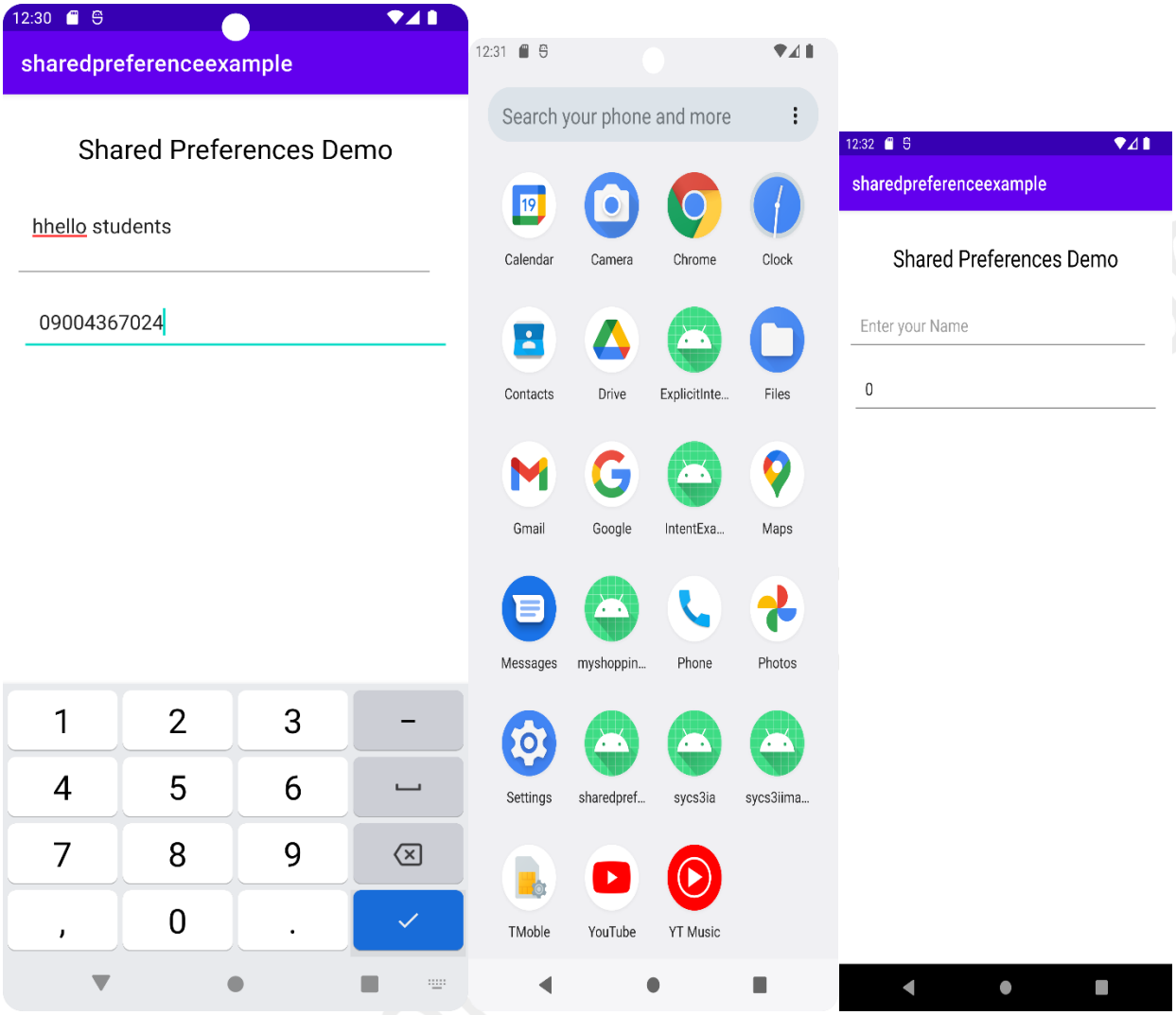
```
        android:layout_height="wrap_content"
        android:layout_below="@+id/textview"
        android:layout_marginStart="16dp"
        android:layout_marginTop="8dp"
        android:layout_marginEnd="16dp"
        android:hint="Enter your Name"
        android:padding="10dp" />

    <!--EditText to take the data from the user and save the data in SharedPreferences-->
    <EditText
        android:id="@+id/edit2"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_below="@+id/edit1"
        android:layout_marginStart="16dp"
        android:layout_marginTop="8dp"
        android:layout_marginEnd="16dp"
        android:hint="Enter your Age"
        android:inputType="number"
        android:padding="10dp" />
</RelativeLayout>
```

Now Add The Following Code To MainActivity.kt File-

```
package com.example.sycspractical4ii

import android.os.Bundle
import android.widget.EditText
import androidx.appcompat.app.AppCompatActivity

class MainActivity : AppCompatActivity() {
    private lateinit var name: EditText
    private lateinit var age: EditText

    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_main)
        name = findViewById(R.id.edit1)
        age = findViewById(R.id.edit2)
    }

    // Fetch the stored data in onResume() Because this is what will be called when the app opens again
    override fun onResume() {
        super.onResume()
        // Fetching the stored data from the SharedPreference
        val sh = getSharedPreferences("MySharedPref", MODE_PRIVATE)
        val s1 = sh.getString("name", "")
        val a = sh.getInt("age", 0)

        // Setting the fetched data in the EditTexts
        name.setText(s1)
        age.setText(a.toString())
    }

    // Store the data in the SharedPreference in the onPause() method
    // When the user closes the application onPause() will be called and data will be stored
    override fun onPause() {
        super.onPause()
        // Creating a shared pref object with a file name "MySharedPref" in private mode
        val sharedPreferences = getSharedPreferences("MySharedPref", MODE_PRIVATE)
        val myEdit = sharedPreferences.edit()

        // write all the data entered by the user in SharedPreference and apply
```

```
        myEdit.putString("name",  name.text.toString())
        myEdit.putInt("age", age.text.toString().toInt())
        myEdit.apply()
    }
}
```

OUTPUT:-



OUTPUT EXPLAINATION :
When user enter details and minimizes current activity tab then this edittext got reset automatically as core concept of shared preference works.

Program 5:-

Program 5(i)-Create an Android application to demonstrate the use of Broadcast listeners.

Firstly add the following code to activity_main.xml file-

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

```
<android.support.constraint.ConstraintLayout
xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".MainActivity">

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Hello World!"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent" />

</android.support.constraint.ConstraintLayout>
```

Now Add The Following Code To MainActivity.kt File:-

```
package com.example.sycspractical5i

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)
    }
}
```

After That Crate a File Named AirPlaneModeChangeReceiver.kt and Add The Following Code to it:-

```
package com.example.sycspractical5i

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()
        }
    }
}
```

OUTPUT:-

Program 5(ii)-Create an Android application to create and use services

activity_main.xml file-

```
<?xml version="1.0" encoding="utf-8"?>
<!--suppress ALL -->
<android.support.constraint.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"
    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:fontFamily="@font/roboto"
            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:fontFamily="@font/roboto"
            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:fontFamily="@font/roboto"
        android:text="@string/stopButtonText"
        android:textAlignment="center"
        android:textAppearance="@style/TextAppearance.AppCompat.Display1"
        android:textColor="#FFFFFF"
        android:textStyle="bold" />

    <ImageView
        android:id="@+id/imageView"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_marginTop="80dp"
        app:srcCompat="@drawable/banner" />
</LinearLayout>


</android.support.constraint.ConstraintLayout>
```

MainActivity.kt File:-

```
package com.example.sycspractical5ii

import android.content.Intent
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))
        }
    }
}
```

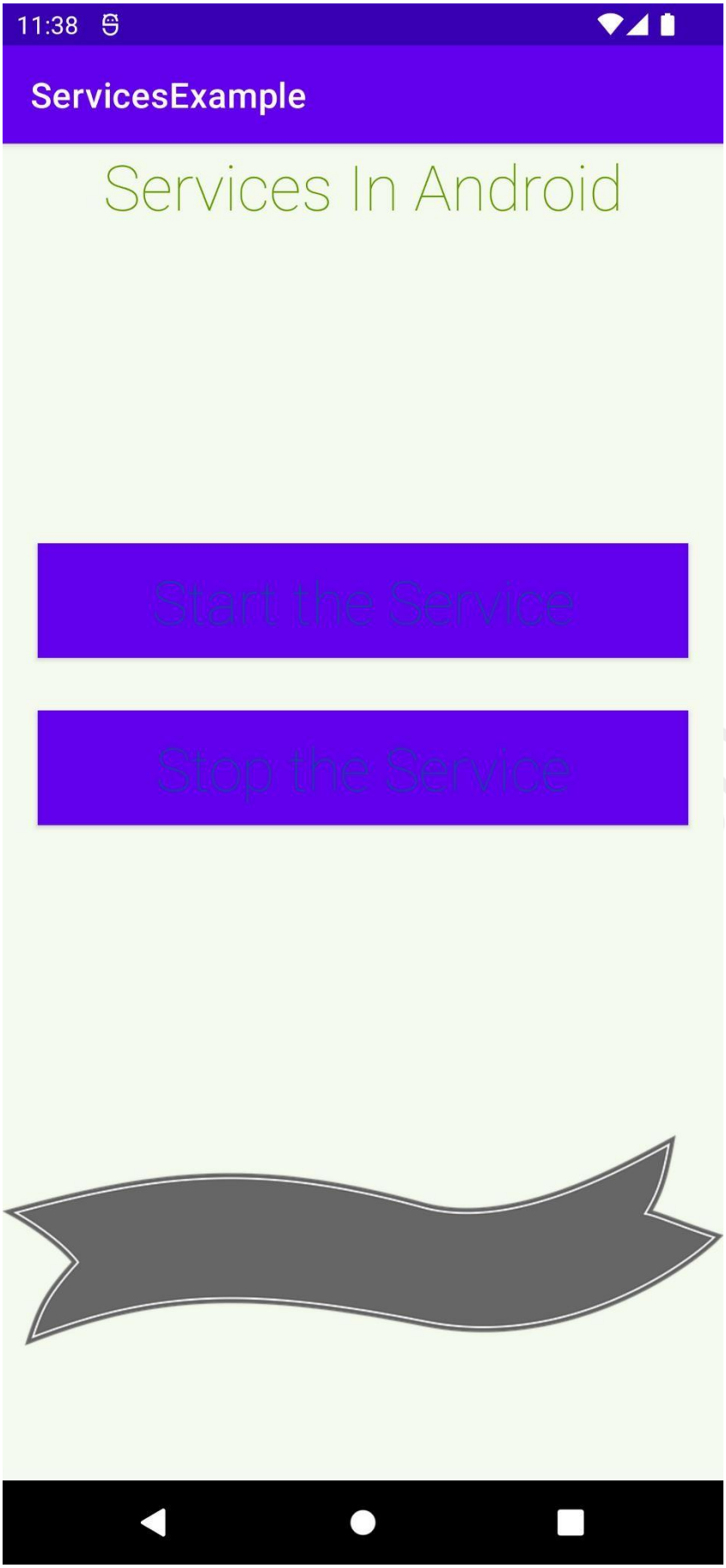
Now Create A New File named NewService.kt and Add The Code:-

```
package com.example.sycspractical5ii

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()
    }

    override fun onBind(intent: Intent): IBinder? {
        return null
    }
}
```

Strings.xml File:-

```
<resources>
    <string name="app_name">SYCSPractical5ii</string>
    <string name="heading">Services In Android</string>
    <string name="startButtonText">Start the Service</string>
    <string name="stopButtonText">Stop the Service</string>
</resources>
```

OUTPUT:-



Program 6:-

Program 6(i):-Create an Android application to demonstrate XML based animation

Activity_main.xml File:-

```
<?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:orientation="vertical">

    <Button
        android:id="@+id/button"
        android:layout_centerInParent="true"
        android:background="@color/colorPrimary"
        android:textColor="#ffffff"
        android:text="Let's Bounce"
        android:layout_width="200dp"
        android:layout_height="80dp"/>

</RelativeLayout>
```

MainActivity.kt File-

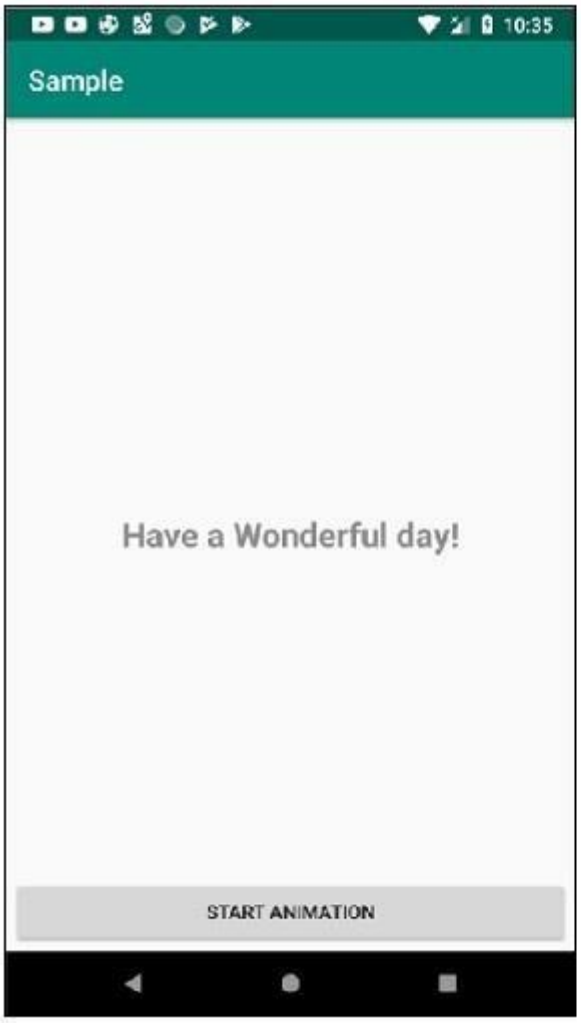
```
package com.example.sycspractical6i

import android.os.Bundle
import android.support.v7.app.AppCompatActivity
import android.view.animation.Animation
import android.view.animation.AnimationUtils
import android.widget.Button


class MainActivity : AppCompatActivity() {
    protected override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_main)

        // loading Animation from
        val animation: Animation = AnimationUtils.loadAnimation(this, R.anim.bounce)

        // getting the Button from activity_main.xml file
        val button: Button = findViewById(R.id.button)
        button.setOnClickListener { // start the animation
            button.startAnimation(animation)
        }
    }
}
```

Create a new File bounce.xml and add the following code:-

```
<?xml version="1.0" encoding="utf-8"?>
<set
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:fillAfter="true"
    android:interpolator="@android:anim/bounce_interpolator">

    <scale
        android:pivotX="50%"
        android:pivotY="50%"
        android:fromXScale="0.5"
        android:toXScale="1.0"
        android:fromYScale="0.5"
        android:toYScale="1.0"
```

```
        android:duration="500"/>
</set>
```

OUTPUT:-



Program 6(ii)-Create an Android application to display canvas and allow the user to draw on it.

activity_main.xml file -

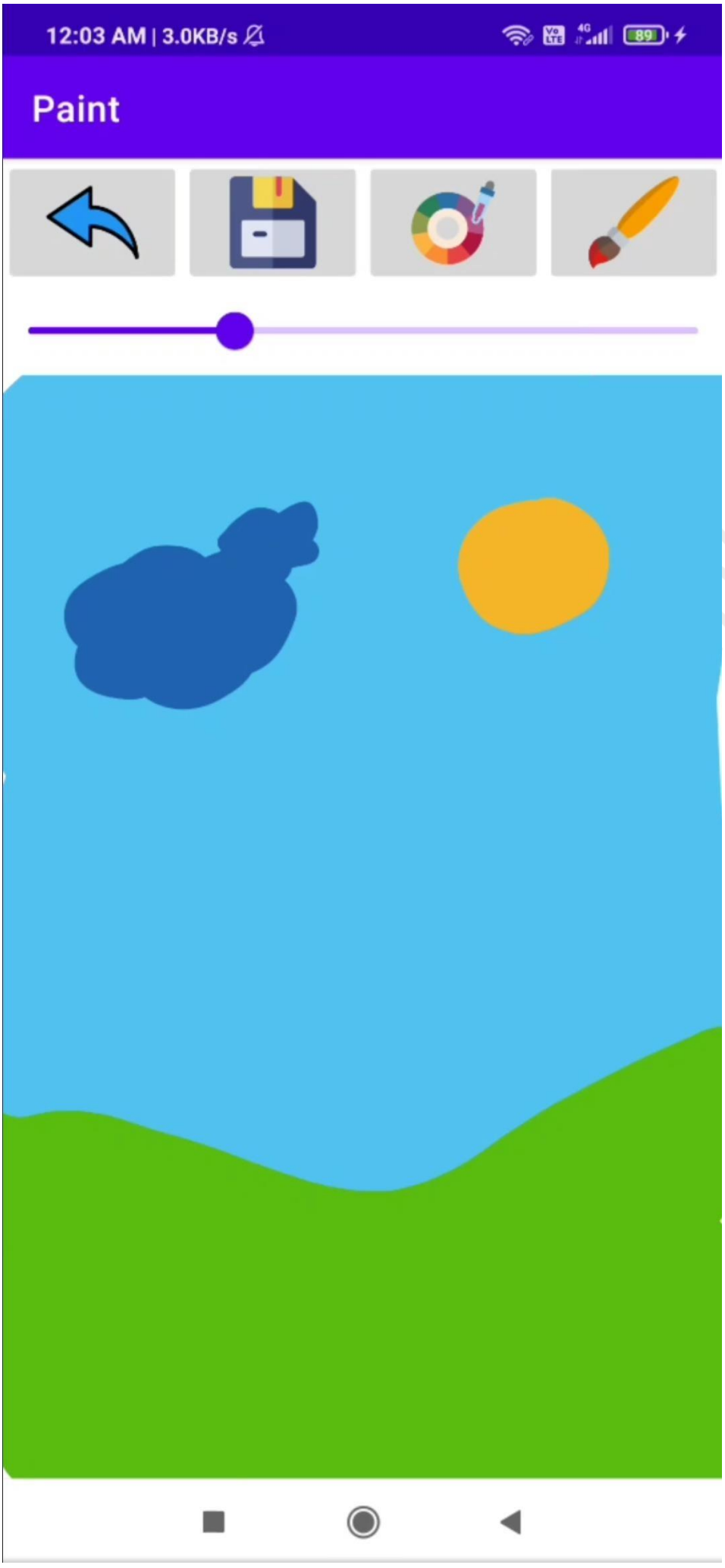
```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".MainActivity">

    <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.kt File –

```
package com.example.sycspractical6ii

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 android.os.Bundle
import android.support.annotation.RequiresApi
import android.support.v7.app.AppCompatActivity
import android.view.MotionEvent
import android.view.View
import android.widget.ImageView


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:-


12:03 AM | 3.0KB/s
Paint

Program 7:-
Program 7(i)-
Create a media player application in android that plays audio. Implement play, pause, and loop features

activity_main.xml file-

```
<?xml version="1.0" encoding="utf-8"?>
<android.support.constraint.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/pauseBtn"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginBottom="8dp"
        android:layout_marginEnd="8dp"
        android:layout_marginStart="8dp"
        android:layout_marginTop="8dp"
        android:enabled="false"
        android:text="Pause"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toStartOf="@+id/playBtn"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent" />

    <Button
        android:id="@+id/playBtn"
        android:layout_width="88dp"
        android:layout_height="wrap_content"
        android:layout_marginBottom="8dp"
        android:layout_marginEnd="8dp"
        android:layout_marginStart="8dp"
        android:layout_marginTop="8dp"
        android:text="Play"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toStartOf="@+id/stopBtn"
        app:layout_constraintStart_toEndOf="@+id/pauseBtn"
        app:layout_constraintTop_toTopOf="parent" />

    <Button
        android:id="@+id/stopBtn"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginBottom="8dp"
        android:layout_marginEnd="24dp"
        android:layout_marginRight="24dp"
        android:layout_marginTop="8dp"
        android:enabled="false"
        android:text="Stop"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintTop_toTopOf="parent" />

    <RelativeLayout
        android:layout_width="368dp"
```

```
        android:layout_height="wrap_content"
        android:layout_marginEnd="8dp"
        android:layout_marginStart="8dp"
        android:layout_marginTop="76dp"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="1.0"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent">

        <TextView
            android:id="@+id/tv_pass"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content" />

        <TextView
            android:id="@+id/tv_due"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_alignParentEnd="true"
            android:layout_alignParentRight="true" />

        <SeekBar
            android:id="@+id/seek_bar"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:layout_below="@id/tv_pass"
            android:saveEnabled="false" />
    </RelativeLayout>
</android.support.constraint.ConstraintLayout>
```

MainActivity.kt file-

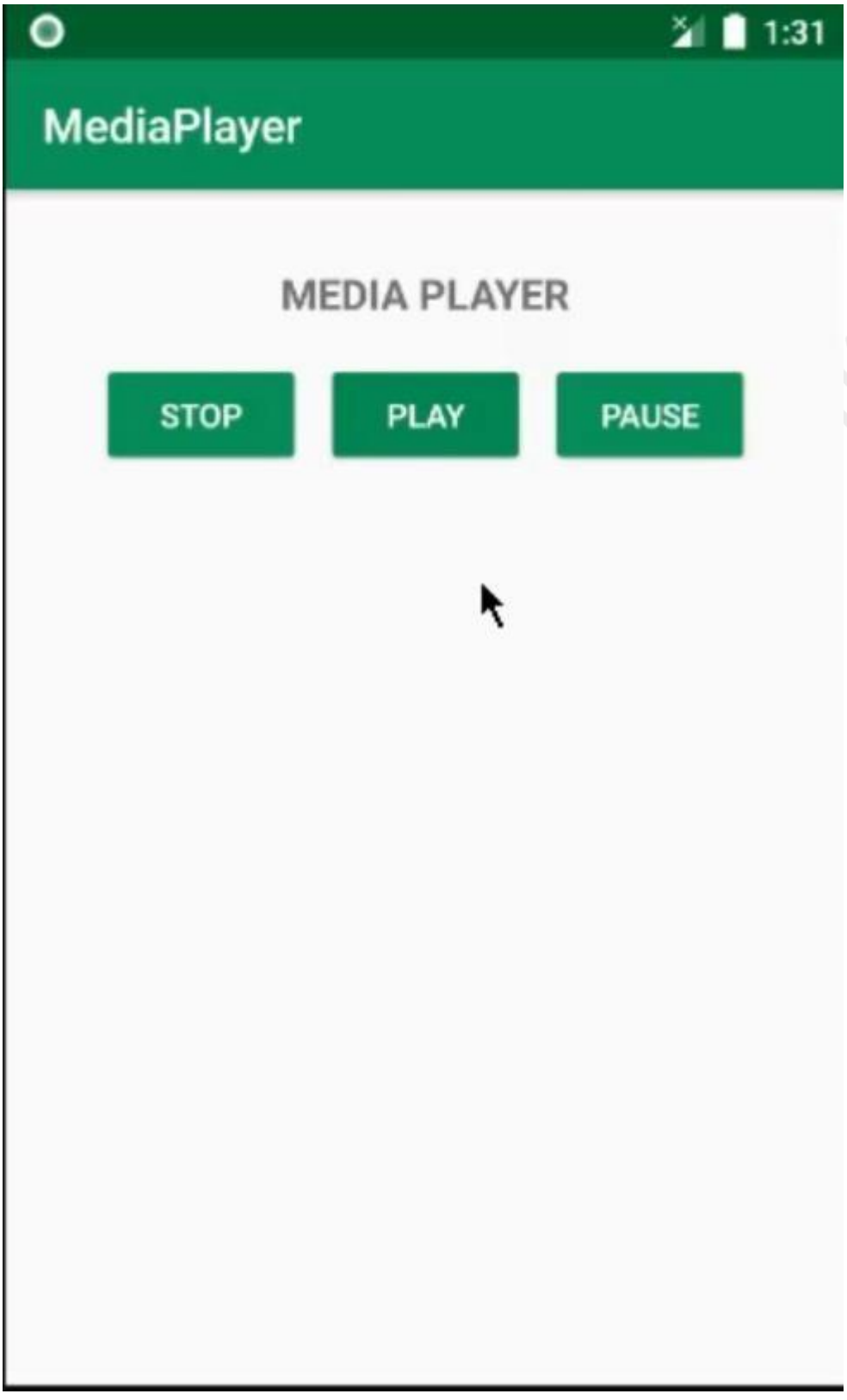
```
package com.example.sycspractical7i

import android.media.MediaPlayer
import android.support.v7.app.AppCompatActivity
import android.os.Bundle
import android.widget.Toast
import android.os.Handler
import android.widget.SeekBar

class MainActivity : AppCompatActivity() {
    private lateinit var mediaPlayer: MediaPlayer
    private lateinit var runnable:Runnable
    private var handler: Handler = Handler()
    private var pause:Boolean = false
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_main)

        // Start the media player
        val playBtn = null
        playBtn.setOnClickListener{
            if(pause){
                mediaPlayer.seekTo(mediaPlayer.currentPosition)
                mediaPlayer.start()
                pause = false
                Toast.makeText(this,"media playing",Toast.LENGTH_SHORT).show()
            }else{

                mediaPlayer = MediaPlayer.create(applicationContext,R.raw.school_bell)
                mediaPlayer.start()
                Toast.makeText(this,"media playing",Toast.LENGTH_SHORT).show()
```

```
    }
    initializeSeekBar()
    playBtn.isEnabled = false
    val pauseBtn = null
    pauseBtn.isEnabled = true
    val stopBtn = null
    stopBtn.isEnabled = true

    mediaPlayer.setOnCompletionListener {
        playBtn.isEnabled = true
        val pauseBtn = null
        pauseBtn.isEnabled = false
        val stopBtn = null
        stopBtn.isEnabled = false
        Toast.makeText(this,"end",Toast.LENGTH_SHORT).show()
    }
}
// Pause the media player
val pauseBtn = null
pauseBtn.setOnClickListener {
    if(mediaPlayer.isPlaying){
        mediaPlayer.pause()
        pause = true
        playBtn.isEnabled = true
        pauseBtn.isEnabled = false
        val stopBtn = null
        stopBtn.isEnabled = true
        Toast.makeText(this,"media pause",Toast.LENGTH_SHORT).show()
    }
}
// Stop the media player
val stopBtn = null
stopBtn.setOnClickListener{
    if(mediaPlayer.isPlaying || pause.equals(true)){
        pause = false
        val seek_bar = null
        seek_bar.setProgress(0)
        mediaPlayer.stop()
        mediaPlayer.reset()
        mediaPlayer.release()
        handler.removeCallbacks(runnable)

        playBtn.isEnabled = true
        pauseBtn.isEnabled = false
        stopBtn.isEnabled = false
        val tv_pass = null
        tv_pass.text = ""
        val tv_due = null
        tv_due.text = ""
        Toast.makeText(this,"media stop",Toast.LENGTH_SHORT).show()
    }
}
// Seek bar change listener
val seek_bar = null
seek_bar.setOnSeekBarChangeListener(object : SeekBar.OnSeekBarChangeListener {
    override fun onProgressChanged(seekBar: SeekBar, i: Int, b: Boolean) {
        if (b) {
            mediaPlayer.seekTo(i * 1000)
        }
    }

    override fun onStartTrackingTouch(seekBar: SeekBar) {
```

```
        }

        override fun onStopTrackingTouch(seekBar: SeekBar) {
        }
    })
  }
  // Method to initialize seek bar and audio stats
  private fun initializeSeekBar() {
    val seek_bar = null
    seek_bar.max() = mediaPlayer.seconds

    runnable = Runnable {
      seek_bar.progress = mediaPlayer.currentSeconds

      val tv_pass = null
      tv_pass.text = "${mediaPlayer.currentSeconds} sec"
      val diff = mediaPlayer.seconds - mediaPlayer.currentSeconds
      val tv_due = null
      tv_due.text = "$diff sec"

      handler.postDelayed(runnable, 1000)
    }
    handler.postDelayed(runnable, 1000)
  }
}

private fun Nothing?.max(): Any {
  TODO("Not yet implemented")
}

private fun Nothing?.setOnSeekBarChangeListener(onSeekBarChangeListener:
SeekBar.OnSeekBarChangeListener) {
  TODO("Not yet implemented")
}

@JvmName("setProgress")
private fun Nothing?.setProgress(i: Int) {
  TODO("Not yet implemented")
}

private fun Nothing?.setOnClickListener(function: () -> Unit) {
  TODO("Not yet implemented")
}

private var Nothing?.progress: Int
  get() {
    TODO("Not yet implemented")
  }
  set(progress:) {}
private var Nothing?.text: String
  get() {
    TODO("Not yet implemented")
  }
  set(text) {}
private var Nothing?.isEnabled: Boolean
  get() {
    TODO("Not yet implemented")
  }
  set(isEnabled) {}

// Creating an extension property to get the media player time duration in seconds
val MediaPlayer.seconds:Int
  get() {
```

```
        return this.duration / 1000
    }
// Creating an extension property to get media player current position in seconds
val MediaPlayer.currentSeconds:Int
    get() {
        return this.currentPosition/1000
    }
```

OUTPUT:-



Program 7(ii)-Create an Android application to use a camera and capture image/video and display them on the screen.

activity_main.xml File-

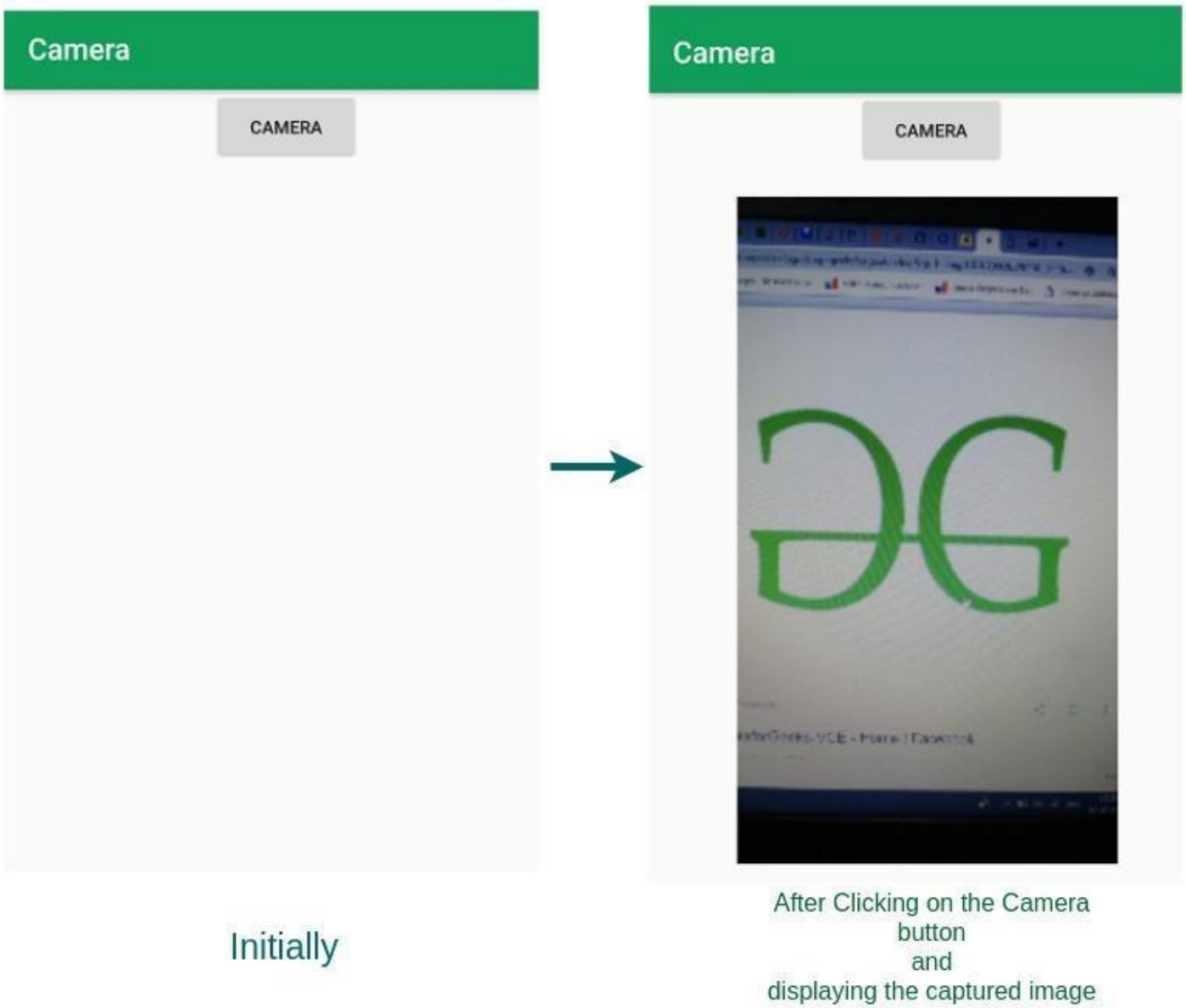
```
<RelativeLayout xmlns:androclass="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    xmlns:android="http://schemas.android.com/apk/res-auto"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".MainActivity"
    androclass:layout_height="match_parent"
    androclass:layout_width="match_parent"
    tools:ignore="NamespaceTypo">

    <Button
        android:id="@+id/button1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignParentBottom="true"
        android:layout_centerHorizontal="true"
        android:text="Take a Photo"
        androclass:layout_height="match_parent"
        androclass:layout_width="match_parent">
    </Button>

    <ImageView
        android:id="@+id/imageView1"
        android:layout_width="fill_parent"
        android:layout_height="fill_parent"
        android:layout_above="@+id/button1"
        android:layout_alignParentTop="true"
        android:src="@drawable/ic_launcher"
        androclass:layout_height="match_parent"
        androclass:layout_width="match_parent"
        tools:ignore="NotSibling"
        androclass:contentDescription="TODO">
    </ImageView>
</RelativeLayout>
```

MainActivity.kt file:-

```
package com.example.sycspractical7ii

import android.annotation.SuppressLint
import android.app.Activity
import android.content.Intent
import android.graphics.Bitmap
import android.os.Bundle
import android.provider.MediaStore
import android.view.Menu
import android.view.View
import android.widget.Button
import android.widget.ImageView


class MainActivity : Activity() {
    var imageView: ImageView? = null
    @SuppressLint("MissingInflatedId")
    public override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_main)
        imageView = findViewById<View>(R.id.imageView1) as ImageView
        val photoButton = findViewById<View>(R.id.button1) as Button
        photoButton.setOnClickListener {
            val cameraIntent = Intent(MediaStore.ACTION_IMAGE_CAPTURE)
            startActivityForResult(cameraIntent, CAMERA_REQUEST)
```

```
        }
    }

    override fun onActivityResult(requestCode: Int, resultCode: Int, data: Intent) {
        if (requestCode == CAMERA_REQUEST) {
            val photo = data.extras!!["data"] as Bitmap?
            imageView!!.setImageBitmap(photo)
        }
    }

    override fun onCreateOptionsMenu(menu: Menu): Boolean {
         getMenuInflater().inflate(R.menu.activity_main, menu);
        return true
    }

    companion object {
        private const val CAMERA_REQUEST = 1888
    }
}
```

OUTPUT:-



Program 8:-

Program 8(i)-Create an android application to implement Asynctask and threading concepts

activity_main.xml File-

```
<?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:id = "@+id/rootview"
   android:layout_width = "match_parent"
   android:layout_height = "match_parent"
   android:orientation = "vertical"
   android:background = "#c1c1c1"
   android:gravity = "center_horizontal"
   tools:context = ".MainActivity">
   <Button
      android:id = "@+id/asyncTask"
      android:text = "Download"
      android:layout_width = "wrap_content"
      android:layout_height = "wrap_content" />
   <ImageView
      android:id = "@+id/image"
      android:layout_width = "300dp"
      android:layout_height = "300dp" />
</LinearLayout>
```

MainActivity.kt file-

```
package com.example.myapplication

import android.app.ProgressDialog
import android.graphics.Bitmap
import android.graphics.BitmapFactory
import android.net.wifi.WifiConfiguration.AuthAlgorithm.strings
import android.os.AsyncTask
import android.os.Bundle
import android.support.v7.app.AppCompatActivity
import android.widget.Button
import android.widget.ImageView
import  java.io.IOException
import java.io.InputStream
import java.net.HttpURLConnection
import java.net.URL


enum class AsyncTaskExample {

}

class MainActivity : AppCompatActivity() {
   var ImageUrl: URL? = null
   var `is`: InputStream? = null
   var bmImg: Bitmap? = null
   var imageView: ImageView? = null
   var p: ProgressDialog? = null
   override fun onCreate(savedInstanceState: Bundle?) {
      super.onCreate(savedInstanceState)
      setContentView(R.layout.activity_main)
      val button = findViewById<Button>(R.id.asyncTask)
      imageView = findViewById(R.id.image)
      button.setOnClickListener {
         val asyncTask: AsyncTaskExample = this.AsyncTaskExample() {
```

```
        }
    }

    abstract class AsyncTaskExample :
        AsyncTask<String?, String?, Bitmap?>() {
        override fun onPreExecute() {
            super.onPreExecute()
            p = ProgressDialog(this@MainActivity)
            p!!.setMessage("Please wait...It is downloading")
            p!!.isIndeterminate = false
            p!!.setCancelable(false)
            p!!.show()
        }

        protected override fun doInBackground(vararg p0: String?): Bitmap? {
            try {
                ImageUrl = URL(strings[0])
                val conn = ImageUrl!!.openConnection() as HttpURLConnection
                conn.doInput = true
                conn.connect()
                `is` = conn.inputStream
                val options = BitmapFactory.Options()
                options.inPreferredConfig = Bitmap.Config.RGB_565
                bmImg = BitmapFactory.decodeStream(`is`, null, options)
            } catch (e: IOException) {
                e.printStackTrace()
            }
            return bmImg
        }

        override fun onPostExecute(bitmap: Bitmap?) {
            super.onPostExecute(bitmap)
            if (imageView != null) {
                p!!.hide()
                imageView!!.setImageBitmap(bitmap)
            } else {
                p!!.show()
            }
        }
    }
}

    private fun AsyncTaskExample(function: () -> Unit) {
        "TODO(\"Not yet implemented\")"}
```

OUTPUT:-

Program 8(ii)-
Create an Android application to demonstrate the different types of menus.
a. Pop-up Menu
b. Context Menu
c. Option Menu

Program 8(ii)a-Program For Pop-up Menu

Activity_main.xml File-

```
<?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/clickBtn"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:background="#0F9D58"
        android:text="Click Me"
        android:textColor="#ffffff"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintLeft_toLeftOf="parent"
        app:layout_constraintRight_toRightOf="parent"
        app:layout_constraintTop_toTopOf="parent" />
```

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

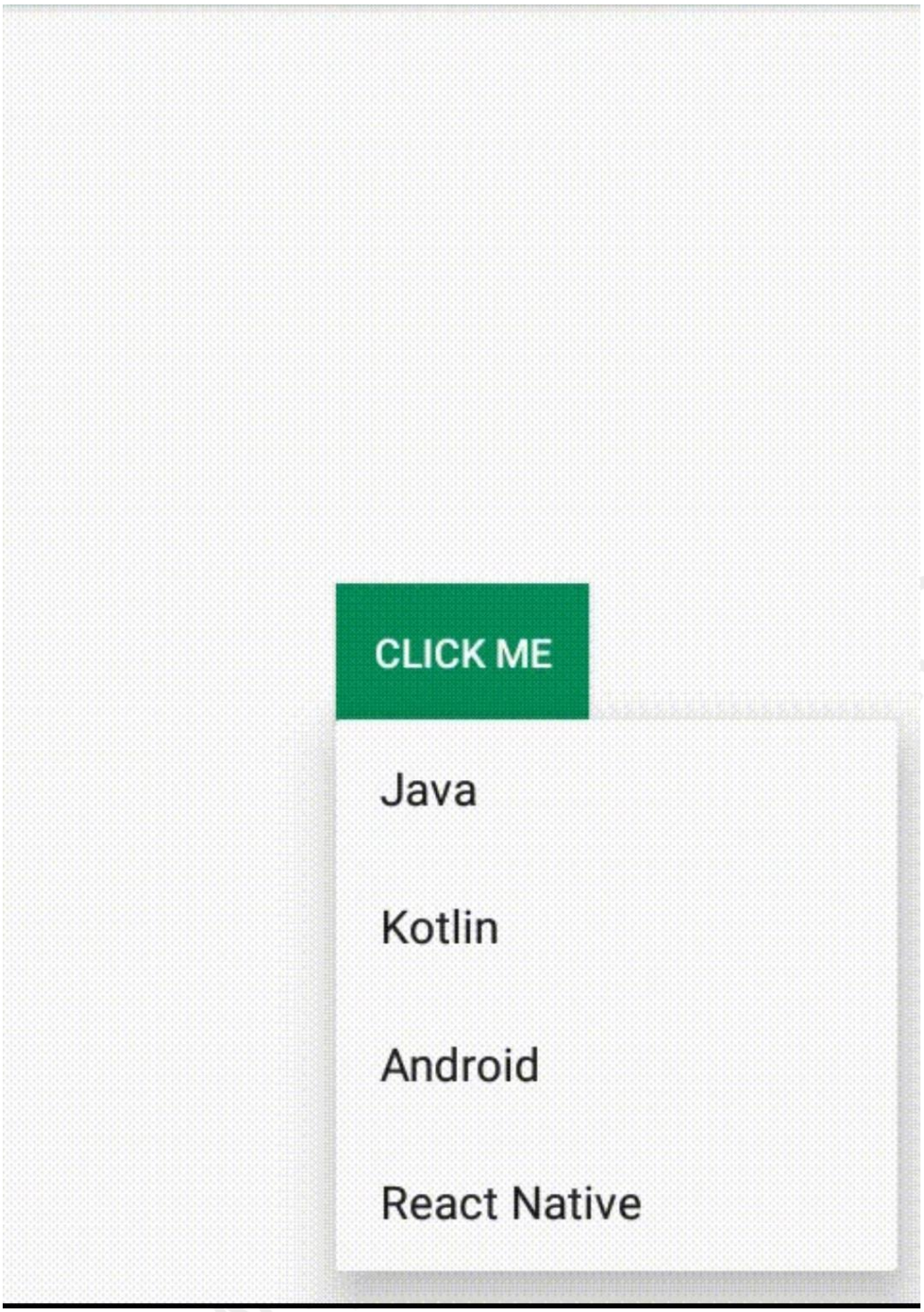
MainActivity.kt File:-

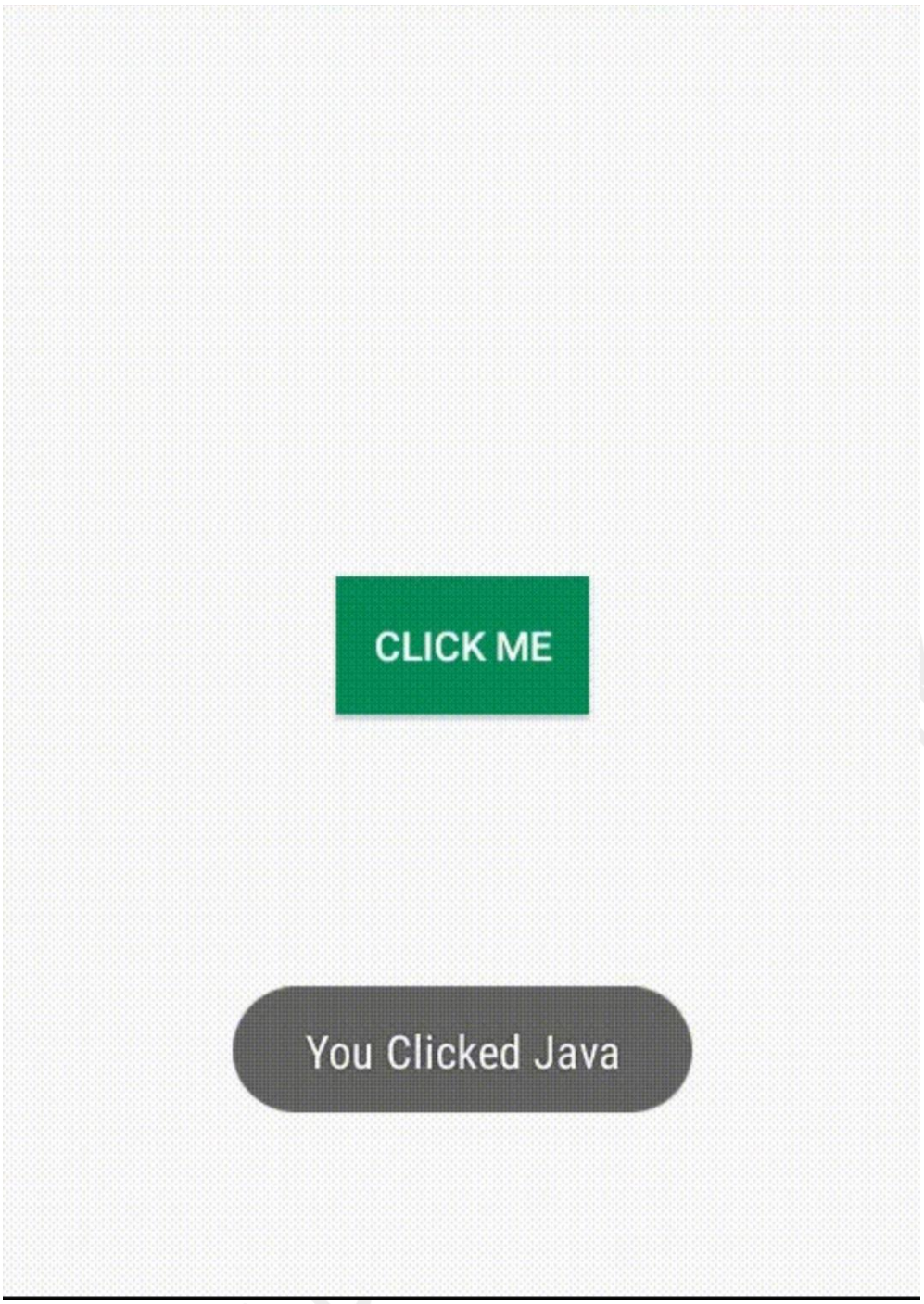
```
package com.example.sycspractical8iia

import android.os.Bundle
import android.widget.Button
import android.widget.PopupMenu
import android.widget.Toast
import androidx.appcompat.app.AppCompatActivity

class MainActivity : AppCompatActivity() {
    lateinit var button: Button

    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_main)

        // Referencing and Initializing the button
        button = findViewById(R.id.clickBtn)

        // Setting onClick behavior to the button
        button.setOnClickListener {
            // Initializing the popup menu and giving the reference as current context
            val popupMenu = PopupMenu(this@MainActivity, button)

            // Inflating popup menu from popup_menu.xml file
            popupMenu.menuInflater.inflate(R.menu.popup_menu, popupMenu.menu)
            popupMenu.setOnMenuItemClickListener { menuItem ->
                // Toast message on menu item clicked
                Toast.makeText(this@MainActivity, "You Clicked " + menuItem.title,
Toast.LENGTH_SHORT).show()
                true
            }
            // Showing the popup menu
            popupMenu.show()
        }
    }
}
```

Now Crate a New Directory 'Menu' And Then Create a File named popup_menu.xml and add the code:-

```
<?xml version="1.0" encoding="utf-8"?>
<menu xmlns:android="http://schemas.android.com/apk/res/android">
    <item
        android:id="@+id/java"
        android:title="Java" />

    <item
        android:id="@+id/kotlin"
        android:title="Kotlin" />

    <item
        android:id="@+id/android"
        android:title="Android" />

    <item
        android:id="@+id/react_native"
        android:title="React Native" />
</menu>
```

OUTPUT:-





Program 8(ii)b-Program For ContextMenu

activity_main.xml File-

```
<?xml version="1.0" encoding="utf-8"?>
<!-- Relative Layout to display all the details -->
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:id="@+id/relLayout"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="#fff"
    android:padding="16dp"
    tools:context=".MainActivity">

    <TextView
        android:id="@+id/textView"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_centerHorizontal="true"
        android:layout_marginTop="20dp"
        android:text="Long press me!"
        android:textColor="#000"
        android:textSize="20sp"
        android:textStyle="bold" />
</RelativeLayout>
```

MainActivity.kt fille-

```
package com.example.sycspractical8iib

import android.graphics.Color
import android.os.Bundle
import android.view.ContextMenu
import android.view.ContextMenu.ContextMenuInfo
import android.view.MenuItem
import android.view.View
import android.widget.RelativeLayout
import android.widget.TextView
import androidx.appcompat.app.AppCompatActivity

class MainActivity : AppCompatActivity() {
    lateinit var textView: TextView
    lateinit var relativeLayout: RelativeLayout

    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_main)

        // Link those objects with their respective id's that we have given in .XML file
        textView = findViewById(R.id.textView)
        relativeLayout = findViewById(R.id.relLayout)

        // here you have to register a view for context menu you can register any view
        // like listview, image view, textview, button etc
        registerForContextMenu(textView)
    }

    override fun onCreateContextMenu(menu: ContextMenu, v: View, menuInfo: ContextMenuInfo) {
        super.onCreateContextMenu(menu, v, menuInfo)
        // you can set menu header with title icon etc
        menu.setHeaderTitle("Choose a color")
```

```
        // add menu items
        menu.add(0, v.id, 0, "Yellow")
        menu.add(0, v.id, 0, "Gray")
        menu.add(0, v.id, 0, "Cyan")
    }

    // menu item select listener
    override fun onContextItemSelected(item: MenuItem): Boolean {
        if (item.title === "Yellow") {
            relativeLayout.setBackgroundColor(Color.YELLOW)
        } else if (item.title === "Gray") {
            relativeLayout.setBackgroundColor(Color.GRAY)
        } else if (item.title === "Cyan") {
            relativeLayout.setBackgroundColor(Color.CYAN)
        }
        return true
    }
}
```

OUTPUT:-

Program 8(ii)c-Program For Option Menu

activity_main.xml file-

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

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Hello World!"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent" />

</android.support.constraint.ConstraintLayout>
```

MainActivity.kt file-

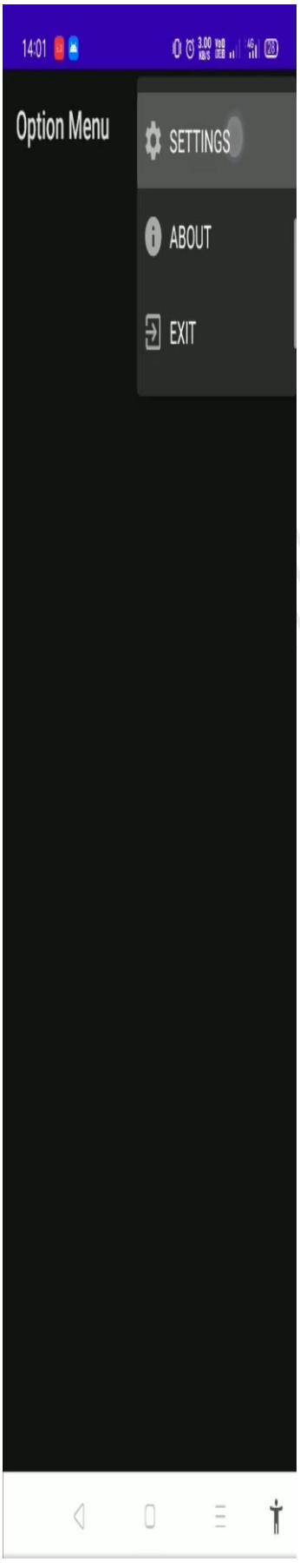
```
package com.example.sycspractical8iic
import android.os.Bundle
import android.support.v7.app.AppCompatActivity
import android.view.Menu
import android.view.MenuItem
import android.widget.Toast
class MainActivity : AppCompatActivity() {
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_main)
    }
    override fun onCreateOptionsMenu(menu: Menu?): Boolean {
```

```
        menuInflater.inflate(R.menu.menu,menu)
        return super.onCreateOptionsMenu(menu)
    }
    override fun onOptionsItemSelected(item: MenuItem): Boolean {
        when (item.itemId){
            R.id.about -> Toast.makeText(this,"About Selected",Toast.LENGTH_SHORT).show()
            R.id.settings -> Toast.makeText(this,"Settings Selected",Toast.LENGTH_SHORT).show()
            R.id.exit -> Toast.makeText(this,"Exit Selected",Toast.LENGTH_SHORT).show()
        }
        return super.onOptionsItemSelected(item)
    }
}
```

Now Create a New Directory "Menu" in res Folder and Create file Named menu.xml and Add The Code-

```
<?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">
    <item
        android:id="@+id/overflowMenu"
        android:icon="@drawable/ic_3_dots"
        android:title=""
        app:showAsAction="always">
        <menu>
            <item
                android:id="@+id/settings"
                android:icon="@drawable/ic_settings"
                android:title="SETTINGS"
                app:showAsAction="never" />
            <item
                android:id="@+id/about"
                android:icon="@drawable/ic_about"
                android:title="ABOUT"
                app:showAsAction="never" />
            <item
                android:id="@+id/exit"
                android:icon="@drawable/ic_exit"
                android:title="EXIT"
                app:showAsAction="never" />
        </menu>
    </item>
</menu>
```

OUTPUT:-


14:01
Option Menu
SETTINGS
ABOUT
EXIT


14:01
Option Menu
Settings Selected

Program 9- Create an Android application to record the current location. Based on the current location allow the user to use some useful services/applications

Activity_main.xml File-

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

   <LinearLayout
      android:layout_width="match_parent"
      android:layout_height="wrap_content"
      android:orientation="vertical"
      android:padding="30dp">

      <TextView
         android:layout_width="match_parent"
         android:layout_height="wrap_content"
         android:textAppearance="@style/TextAppearance.AppCompat.Title"
         android:text="Current Location:"
         />

      <TextView
         android:id="@+id/tvLatitude"
         android:layout_width="match_parent"
         android:layout_height="wrap_content"
         android:textAppearance="@style/TextAppearance.AppCompat.Body1"
         android:layout_marginTop="20dp"
         android:text="Latitude: -"
         />

      <TextView
         android:id="@+id/tvLongitude"
         android:layout_width="match_parent"
         android:layout_height="wrap_content"
         android:textAppearance="@style/TextAppearance.AppCompat.Body1"
         android:layout_marginTop="10dp"
         android:text="Longitude: -"
         />

      <TextView
         android:id="@+id/tvProvider"
         android:layout_width="match_parent"
         android:layout_height="wrap_content"
         android:textAppearance="@style/TextAppearance.AppCompat.Body1"
         android:layout_marginTop="10dp"
         android:text="Provider: -"
         />

      <Button
         android:id="@+id/btOpenMap"
         android:layout_width="150dp"
         android:layout_height="wrap_content"
         android:background="@color/colorAccent"
         android:text="Open Map"
         android:textColor="@android:color/white"
         android:layout_marginTop="30dp"
         android:visibility="gone"
         />

   </LinearLayout>
```

```
    <Button
        android:id="@+id/btGetLocation"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:background="@color/colorPrimary"
        android:layout_margin="30dp"
        android:text="Get Current Location"
        android:textColor="@android:color/white"
        android:layout_alignParentBottom="true"
        />

</RelativeLayout>
```

MainActivity.kt File-

```
package com.example.sycspractical9

import android.Manifest
import android.content.Intent
import android.content.pm.PackageManager
import android.net.Uri
import android.os.Bundle
import android.support.v4.app.ActivityCompat
import android.support.v7.app.AppCompatActivity
import android.view.View
import android.widget.Toast
import com.google.android.gms.location.FusedLocationProviderClient
import kotlinx.android.synthetic.main.activity_main.*

private var Nothing?.visibility: Int
    get() {
        TODO("Not yet implemented")
    }
    set() {}
private var Nothing?.text: String
    get() {
        TODO("Not yet implemented")
    }
    set() {}

class MainActivity : AppCompatActivity() {
    private val LOCATION_PERMISSION_REQ_CODE = 1000;
    private lateinit var fusedLocationClient: FusedLocationProviderClient
    private var latitude: Double = 0.0
    private var longitude: Double = 0.0
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_main)
        // initialize fused location client
        val LocationServices = null
        fusedLocationClient = LocationServices.getFusedLocationProviderClient(this)
        val btGetLocation = null
        btGetLocation.setOnClickListener {
            getCurrentLocation()
        }
        val btOpenMap = null
        btOpenMap.setOnClickListener {
            openMap()
        }
    }
    private fun getCurrentLocation() {
        // checking location permission
```

```
        if (ActivityCompat.checkSelfPermission(this,
                Manifest.permission.ACCESS_FINE_LOCATION)                                    !=
PackageManager.PERMISSION_GRANTED) {
            // request permission
            ActivityCompat.requestPermissions(this,
                arrayOf(Manifest.permission.ACCESS_FINE_LOCATION),
LOCATION_PERMISSION_REQ_CODE);
            return
        }
        val addOnFailureListener = fusedLocationClient.lastLocation
            .addOnSuccessListener { location ->
                // getting the last known or current location
                latitude = location.latitude
                longitude = location.longitude
                val tvLatitude = null
                tvLatitude.text = "Latitude: ${location.latitude}"
                val tvLongitude = null
                tvLongitude.text = "Longitude: ${location.longitude}"
                val tvProvider = null
                tvProvider.text = "Provider: ${location.provider}"
                val btOpenMap = null
                btOpenMap.visibility = View.VISIBLE
            }
            .addOnFailureListener {
                Toast.makeText(
                    this, "Failed on getting current location",
                    Toast.LENGTH_SHORT
                ).show()
            }
    }
    override fun onRequestPermissionsResult(
        requestCode: Int, permissions: Array<out String>, grantResults: IntArray
    ) {
        when (requestCode) {
            LOCATION_PERMISSION_REQ_CODE -> {
                if (grantResults.isNotEmpty() &&
                    grantResults[0] == PackageManager.PERMISSION_GRANTED) {
                    // permission granted
                } else {
                    // permission denied
                    Toast.makeText(this, "You need to grant permission to access location",
                        Toast.LENGTH_SHORT).show()
                }
            }
        }
    }
    private fun openMap() {
        val uri = Uri.parse("geo:${latitude},${longitude}")
        val mapIntent = Intent(Intent.ACTION_VIEW, uri)
        mapIntent.setPackage("com.google.android.apps.maps")
        startActivity(mapIntent)
    }
}

private fun Nothing?.getFusedLocationProviderClient(mainActivity: MainActivity): Any {
    TODO("Not yet implemented")
}

private fun Nothing?.setOnClickListener(function: () -> Unit) {
    TODO("Not yet implemented")
}
```

//difficult method
Other way
MainActivity.kt

```
package com.example.locationactivity
import com.example.locationactivity.LocationService
import android.Manifest
import android.content.pm.PackageManager
import android.os.Bundle
import androidx.appcompat.app.AppCompatActivity
import androidx.core.app.ActivityCompat
import androidx.core.content.ContextCompat

class MainActivity : AppCompatActivity() {
    private lateinit var locationService: LocationService

    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_main)

        locationService = LocationService(this)
        requestPermissions()
    }

    private fun requestPermissions() {
        if (ContextCompat.checkSelfPermission(this, Manifest.permission.ACCESS_FINE_LOCATION) !=
PackageManager.PERMISSION_GRANTED) {
            ActivityCompat.requestPermissions(this,
arrayOf(Manifest.permission.ACCESS_FINE_LOCATION), REQUEST_LOCATION_PERMISSION)
        } else {
            startLocationUpdates()
        }
    }

    private fun startLocationUpdates() {
        locationService.startLocationUpdates()
    }

    override fun onRequestPermissionsResult(requestCode: Int, permissions: Array<out String>,
grantResults: IntArray) {
        super.onRequestPermissionsResult(requestCode, permissions, grantResults)
        if (requestCode == REQUEST_LOCATION_PERMISSION) {
            if (grantResults.isNotEmpty() && grantResults[0] ==
PackageManager.PERMISSION_GRANTED) {
                startLocationUpdates()
            }
        }
    }

    override fun onDestroy() {
        super.onDestroy()
        locationService.stopLocationUpdates()
    }

    companion object {
        private const val REQUEST_LOCATION_PERMISSION = 1
    }
}
```

androidmanifest.xml

```
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    package="com.example.locationactivity">
```

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

    <application
        android:allowBackup="true"
        android:icon="@mipmap/ic_launcher"
        android:label="@string/app_name"
        android:roundIcon="@mipmap/ic_launcher_round"
        android:supportsRtl="true"
        android:theme="@style/AppTheme">

        <activity android:name=".MainActivity"
            android:exported="true"
            tools:ignore="MissingClass">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />

                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>

        <service android:name=".LocationService"
            tools:ignore="MissingClass" />

    </application>

</manifest>
```

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">

    <Button
        android:id="@+id/stop_button"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Stop Location Updates"
        android:layout_centerInParent="true" />

</RelativeLayout>
```

Style.xml

```
<resources>
    <!-- Base application theme. -->
    <style name="AppTheme" parent="Theme.AppCompat.Light.DarkActionBar">
        <!-- Customize your theme here. -->
        <item name="colorPrimary">@color/colorPrimary</item>
        <item name="colorPrimaryDark">@color/colorPrimaryDark</item>
        <item name="colorAccent">@color/colorAccent</item>
    </style>
</resources>
```

Add class LocationService.kt

```
package com.example.locationactivity

import android.Manifest
import android.app.Service
import android.content.Context
import android.content.Intent
```

```
import android.content.pm.PackageManager
import android.location.Location
import android.location.LocationListener
import android.location.LocationManager
import android.os.Bundle
import android.os.IBinder
import android.util.Log
import androidx.core.app.ActivityCompat
import java.io.File



class LocationService(private val context: Context) : LocationListener {
    private var locationManager: LocationManager? = null

    fun startLocationUpdates() {
        locationManager = context.getSystemService(Context.LOCATION_SERVICE) as LocationManager
        if (ActivityCompat.checkSelfPermission(context, Manifest.permission.ACCESS_FINE_LOCATION)
== PackageManager.PERMISSION_GRANTED
            && ActivityCompat.checkSelfPermission(context,
Manifest.permission.ACCESS_COARSE_LOCATION) == PackageManager.PERMISSION_GRANTED) {
            locationManager?.requestLocationUpdates(LocationManager.NETWORK_PROVIDER, 0L, 0f,
this)
        }
    }

    fun stopLocationUpdates() {
        locationManager?.removeUpdates(this)
    }

    override fun onLocationChanged(location: Location) {
        // You can do something with the current location here
    }

    override fun onStatusChanged(provider: String, status: Int, extras: Bundle) {}

    override fun onProviderEnabled(provider: String) {}

    override fun onProviderDisabled(provider: String) {}
}
```

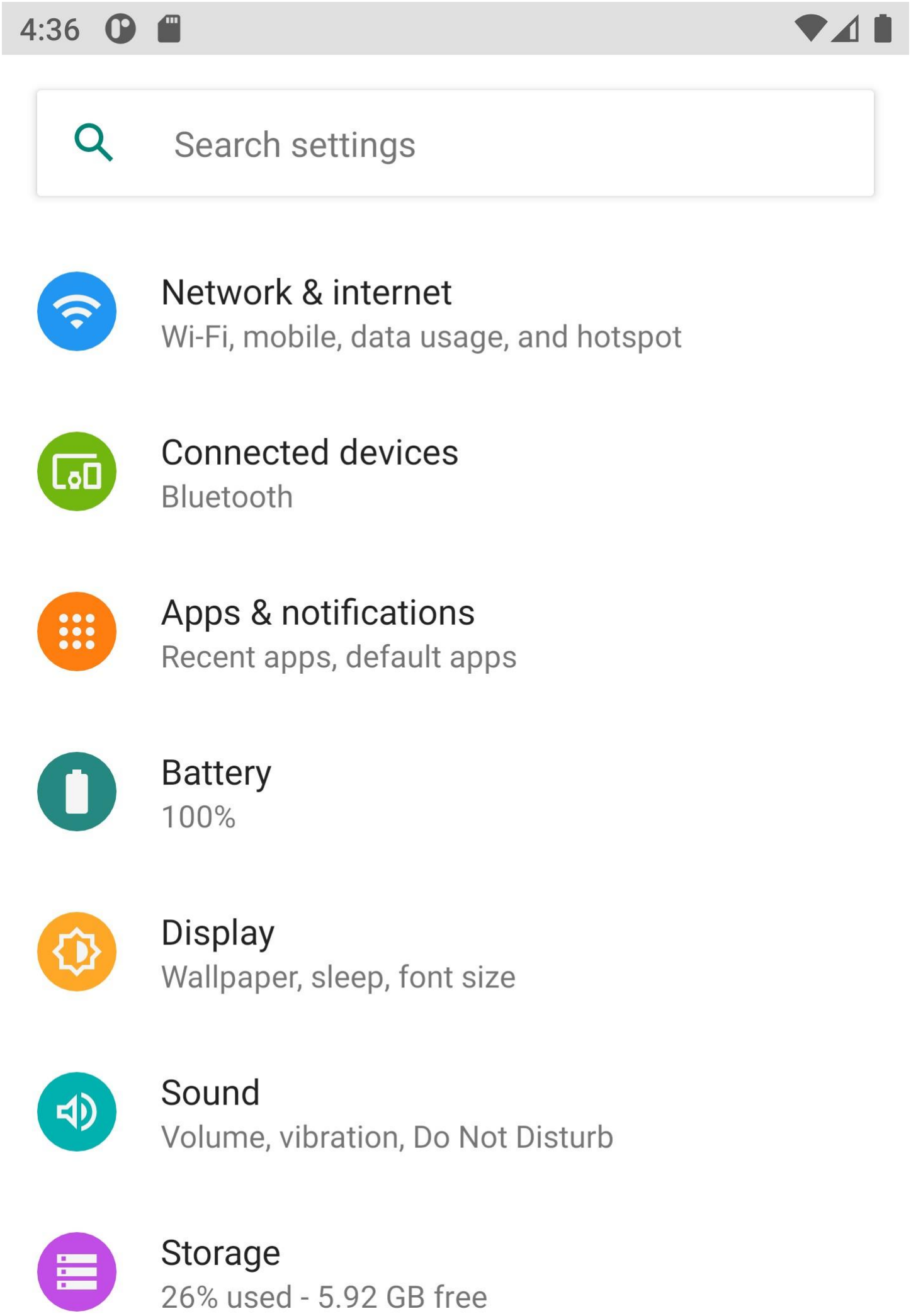
you can explore more try this code:

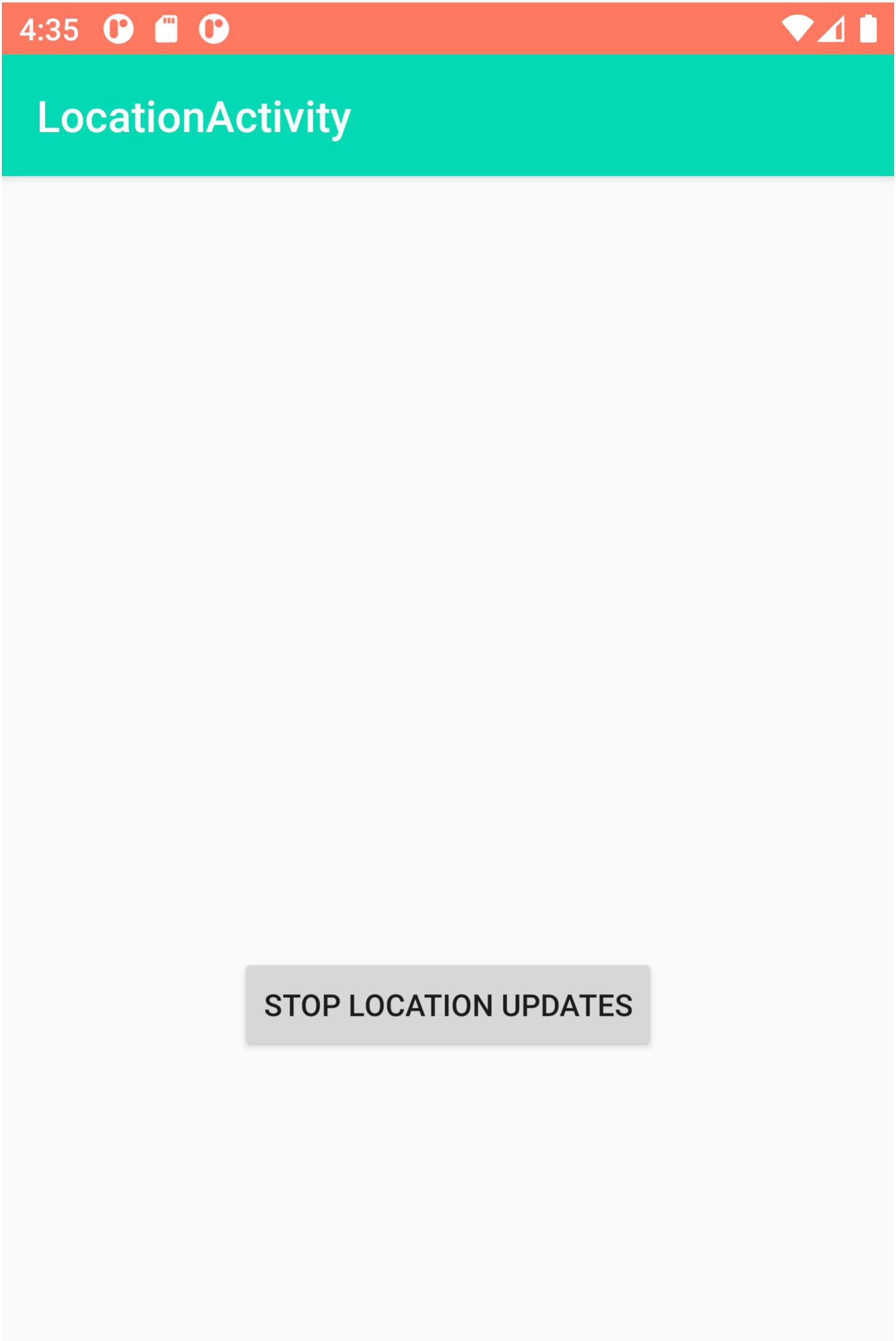
LocationService.kt

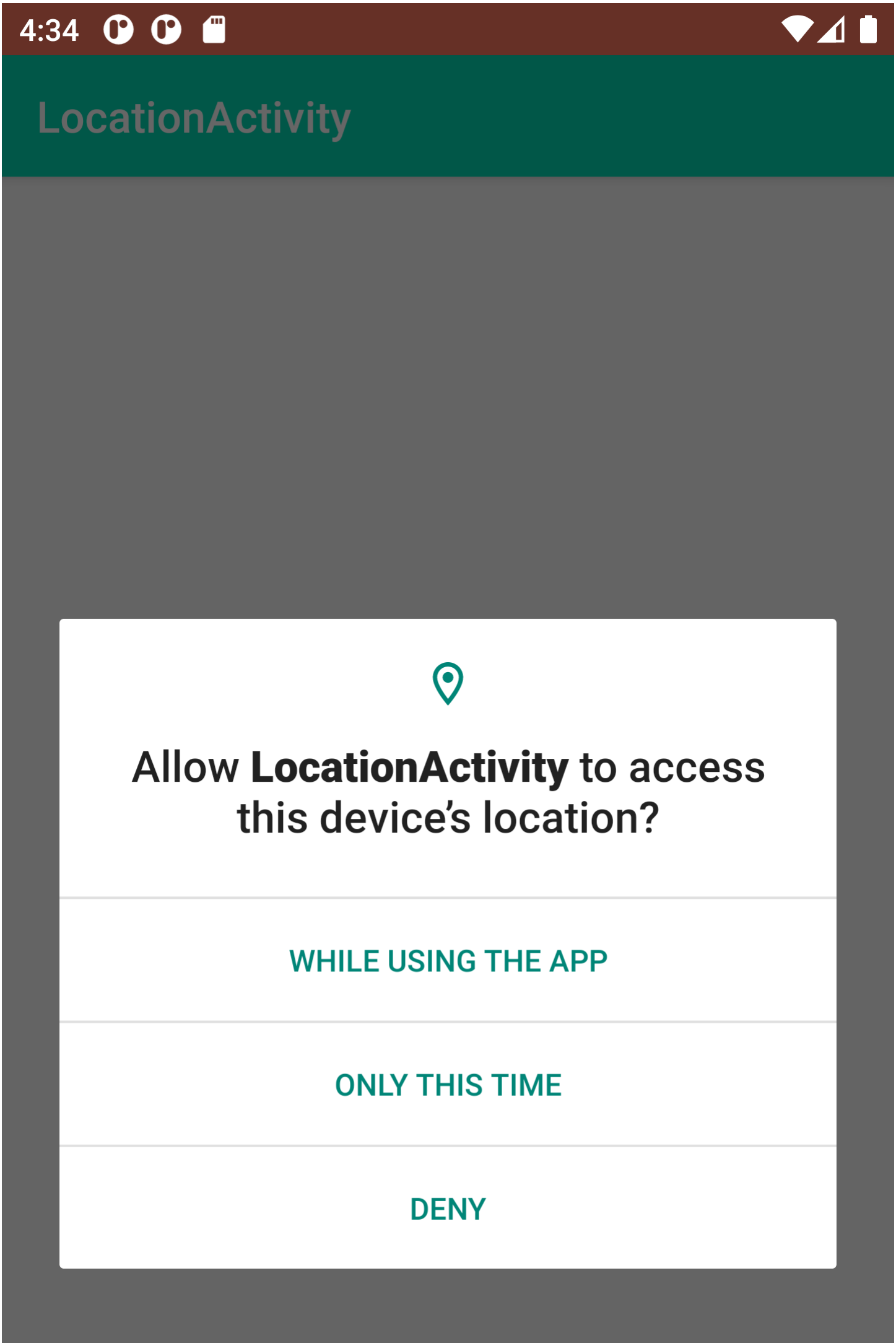
```
 class LocationService : Service() {

    private lateinit var locationManager: LocationManager
    private lateinit var locationListener: LocationListener

    override fun onCreate() {
        super.onCreate()

        // Create the file to store the location data
        val file = File(applicationContext.filesDir, "location_data.txt")

        // Create a new instance of the location manager
        locationManager = getSystemService(Context.LOCATION_SERVICE) as LocationManager

        // Set up the location listener to receive location updates
        locationListener = object : LocationListener {
            override fun onLocationChanged(location: Location) {
                val latitude = location.latitude
```

```
            val longitude = location.longitude
            val timestamp = System.currentTimeMillis()

            val data = "Latitude: $latitude, Longitude: $longitude, Timestamp: $timestamp\n"

            file.appendText(data)
        }

        override fun onStatusChanged(provider: String, status: Int, extras: Bundle) {
            // Do nothing
        }

        override fun onProviderEnabled(provider: String) {
            // Do nothing
        }

        override fun onProviderDisabled(provider: String) {
            // Do nothing
        }
    }
}

// Start location updates
fun startLocationUpdates() {
    try {
        locationManager.requestLocationUpdates(
            LocationManager.GPS_PROVIDER,
            MIN_TIME_BW_UPDATES,
            MIN_DISTANCE_CHANGE_FOR_UPDATES,
            locationListener
        )
    } catch (e: SecurityException) {
        Log.e(TAG, "Error requesting location updates: ${e.message}")
    }
}

// Stop location updates
fun stopLocationUpdates() {
    locationManager.removeUpdates(locationListener)
}

companion object {
    private const val TAG = "LocationService"
    private const val MIN_TIME_BW_UPDATES = 1000L // 1 second
    private const val MIN_DISTANCE_CHANGE_FOR_UPDATES = 0F // 0 meters
}
}
```

OUTPUT:-


4:36
Search settings
Network & internet
Wi-Fi, mobile, data usage, and hotspot
Connected devices
Bluetooth
Apps & notifications
Recent apps, default apps
Battery
100%
Display
Wallpaper, sleep, font size
Sound
Volume, vibration, Do Not Disturb
Storage
26% used - 5.92 GB free


4:35
LocationActivity
STOP LOCATION UPDATES


4:34
LocationActivity
Allow LocationActivity to access this device's location?
WHILE USING THE APP
ONLY THIS TIME
DENY

Program 10- Create a suitable Android application to store and retrieve data in the SQLite database.

Activity_main.xml File-

```
<?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">
    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:orientation="vertical">
        <EditText
            android:id="@+id/editTextName"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:layout_margin="10dp"
            android:padding="8dp" />
        <EditText
            android:id="@+id/editTextAge"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:layout_margin="10dp"
            android:autofillHints="Age"
            android:inputType="number"
            android:padding="8dp"
            android:textColor="@android:color/background_dark" />
        <Button
            android:id="@+id/btnInsert"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:layout_margin="10dp"
            android:padding="8dp"
            android:text="Add data" />
    </LinearLayout>
    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:gravity="center"
        android:orientation="horizontal"
        android:weightSum="3">
        <Button
            android:id="@+id/btnRead"
            android:layout_width="0dp"
            android:layout_height="wrap_content"
            android:layout_margin="10dp"
            android:layout_weight="1"
            android:padding="8dp"
            android:text="Read" />
    </LinearLayout>
    <ScrollView
        android:layout_width="match_parent"
        android:layout_height="match_parent">
    <TextView
      android:id="@+id/tvResult"
      android:layout_width="match_parent"
      android:layout_height="wrap_content"
      android:padding="8dp"
```

```
    android:textSize="16sp"
    android:textStyle="bold" />
  </ScrollView>
</LinearLayout>
```

MainActivity.kt-

```
package com.example.sycspractical10

import android.os.Bundle
import android.support.v7.app.AppCompatActivity
import android.widget.Toast


private val Nothing?.text: Any
    get() {
        TODO("Not yet implemented")
    }

private fun Nothing?.setOnClickListener(function: () -> Unit) {
    TODO("Not yet implemented")
}

private fun Any.clear() {
    TODO("Not yet implemented")
}

class MainActivity : AppCompatActivity() {
    private fun User(toString: String, toInt: Int): Any {
        TODO("Not yet implemented")
    }

    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_main)
        title = "KotlinApp"
        val context = this
        val db = DataBaseHandler(context)
        val btnInsert = null
        btnInsert.setOnClickListener {
            val editTextName = null
            val editTextAge = null
            if (editTextName.text.toString().isNotEmpty() &&
                editTextAge.text.toString().isNotEmpty()
            ) {
                val user = User(editTextName.text.toString(), editTextAge.text.toString().toInt())
                db.insertData(user)
                clearField()
            }
            else {
                Toast.makeText(context, "Please Fill All Data's", Toast.LENGTH_SHORT).show()
            }
        }
        val btnRead = null
        btnRead.setOnClickListener {
            val data = db.readData()
            val tvResult = null
            tvResult.text = ""
            for (i in 0 until data.size) {
                tvResult?.append(
                    data[i].id.toString() + " " + data[i].name + " " + data[i].age + "
                    "
                )
            }
```

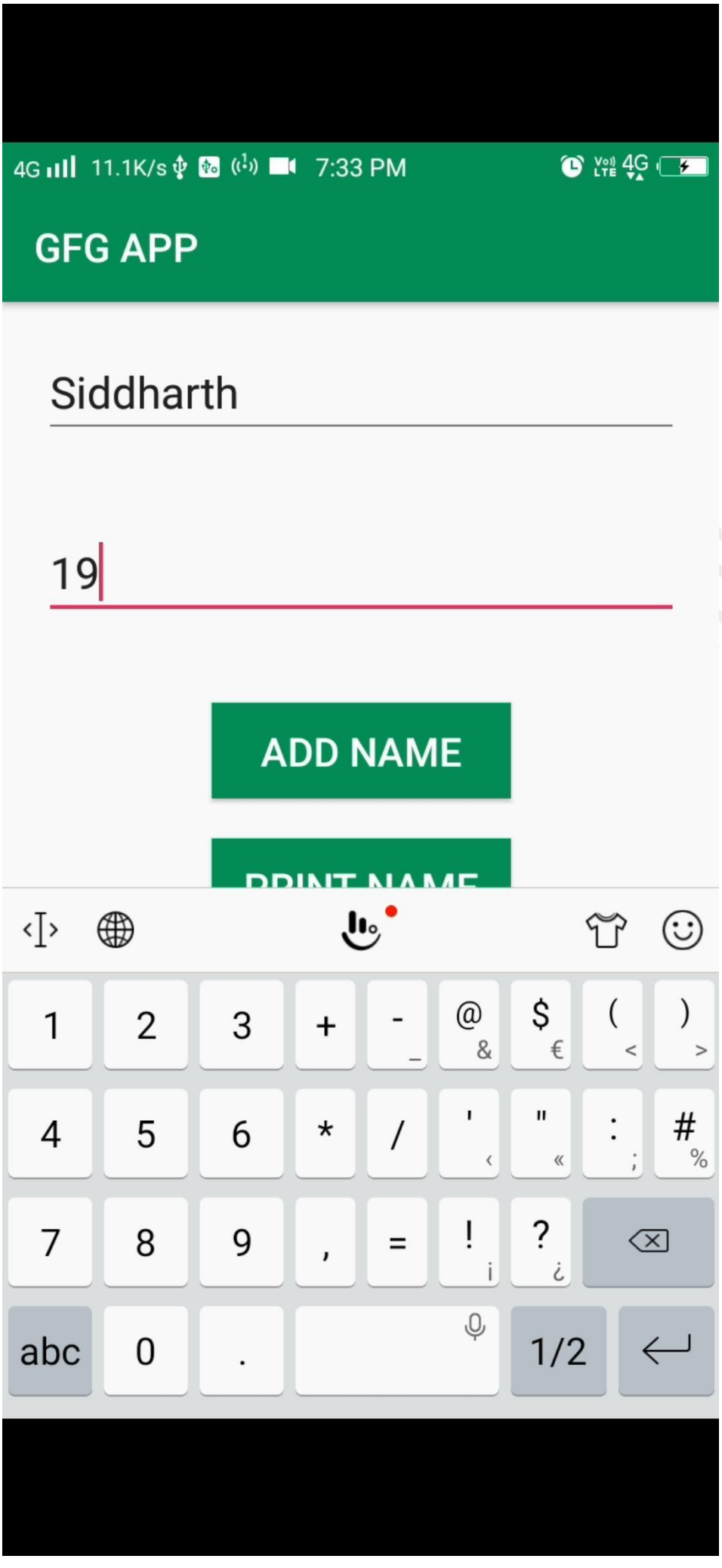
```
        }
    }
    private fun clearField() {
        val editTextName = null
        editTextName.text.clear()
        val editTextAge = null
        editTextAge.text.clear()
    }
}
```

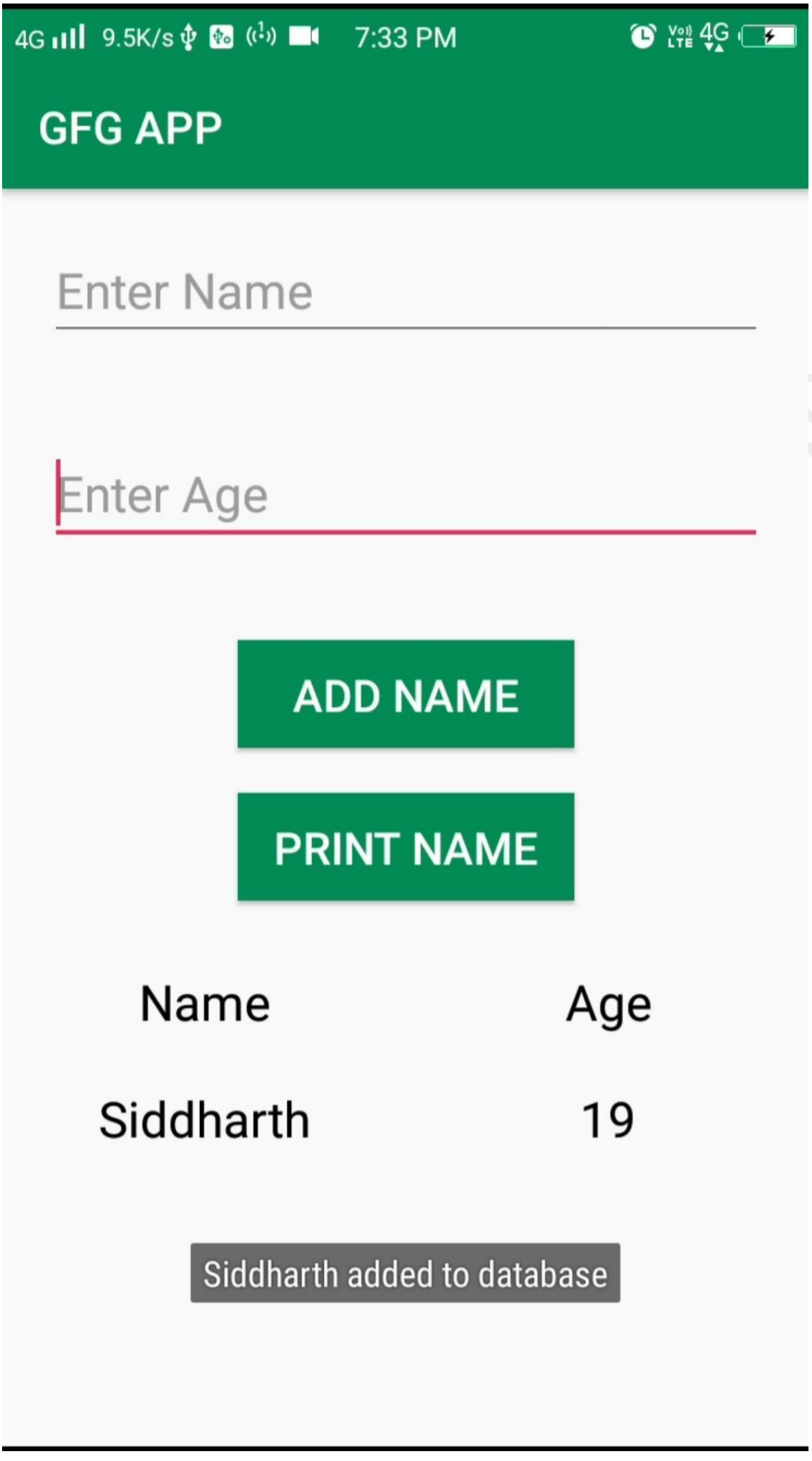
Now Create a New File Named DataBaseHandler.kt and Add The Folllowing code –

```
package com.example.sycspractical10

import android.annotation.SuppressLint
import android.content.ContentValues
import android.content.Context
import android.database.sqlite.SQLiteDatabase
import android.database.sqlite.SQLiteOpenHelper
import android.widget.Toast

private val <User> User.age: String?
    get() {
        TODO("Not yet implemented")
    }
private val <User> User.name: String?
    get() {
        TODO("Not yet implemented")
    }
val DATABASENAME = "MY DATABASE"
val TABLENAME = "Users"
val COL_NAME = "name"
val COL_AGE = "age"
val COL_ID = "id"
class DataBaseHandler<User>(var context: Context) : SQLiteOpenHelper(context, DATABASENAME,
null,
    1) {
    override fun onCreate(db: SQLiteDatabase?) {
        val createTable = "CREATE TABLE " + TABLENAME + " (" + COL_ID + " INTEGER PRIMARY KEY
AUTOINCREMENT," + COL_NAME + " VARCHAR(256)," + COL_AGE + " INTEGER)"
        db?.execSQL(createTable)
    }
    override fun onUpgrade(db: SQLiteDatabase?, oldVersion: Int, newVersion: Int) {
        //onCreate(db);
    }
    fun insertData(user: User) {
        val database = this.writableDatabase
        val contentValues = ContentValues()
        contentValues.put(COL_NAME, user.name)
        contentValues.put(COL_AGE, user.age)
        val result = database.insert(TABLENAME, null, contentValues)
        if (result == (0).toLong()) {
            Toast.makeText(context, "Failed", Toast.LENGTH_SHORT).show()
        }
        else {
            Toast.makeText(context, "Success", Toast.LENGTH_SHORT).show()
        }
    }
    @SuppressLint("Range")
    fun readData(): MutableList<User> {
        val list: MutableList<User> = ArrayList()
        val db = this.readableDatabase
        val query = "Select * from $TABLENAME"
```

```
        val result = db.rawQuery(query, null)
        if (result.moveToFirst()) {
            do {
                val user = User()
                user.id = result.getString(result.getColumnIndex(COL_ID)).toInt()
                user.name = result.getString(result.getColumnIndex(COL_NAME))
                user.age = result.getString(result.getColumnIndex(COL_AGE)).toInt()
                list.add(user)
            }
            while (result.moveToNext())
        }
        return list
    }
}
```

OUTPUT-


4G 11.1K/s 7:33 PM
GFG APP
Siddharth
19
ADD NAME
PRINT NAME
1 2 3 + - @ $ ( )
4 5 6 * / ' " : #
7 8 9 , = ! ?
abc 0 . 1/2


4G 9.5K/s 7:33 PM
GFG APP
Enter Name
Enter Age
ADD NAME
PRINT NAME
Name
Age
Siddharth
19
Siddharth added to database

Program 11- Create a suitable Android application to work with Firebase for storing and manipulating data

First of all add the the Following lines In AndroidManifest.xml File-

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

Now After That Add The Following Code in activity_main.xml file-

```
<?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">

   <!--EditText for adding employee name-->
   <EditText
      android:id="@+id/idEdtEmployeeName"
      android:layout_width="match_parent"
      android:layout_height="wrap_content"
      android:layout_centerHorizontal="true"
      android:layout_margin="10dp"
      android:hint="@string/enter_employee_name"
      android:importantForAutofill="no"
      android:inputType="textPersonName" />

   <!--EditText for adding employee phone-->
   <EditText
      android:id="@+id/idEdtEmployeePhoneNumber"
      android:layout_width="match_parent"
      android:layout_height="wrap_content"
      android:layout_below="@id/idEdtEmployeeName"
      android:layout_margin="10dp"
      android:hint="@string/enter_employee_phone_number"
      android:importantForAutofill="no"
      android:inputType="phone" />

   <!--EditText for adding employee address-->
   <EditText
      android:id="@+id/idEdtEmployeeAddress"
      android:layout_width="match_parent"
      android:layout_height="wrap_content"
      android:layout_below="@id/idEdtEmployeePhoneNumber"
      android:layout_margin="10dp"
      android:hint="@string/enter_employee_address"
      android:inputType="textPostalAddress" />

   <!--Button for adding data to Firebase-->
   <Button
      android:id="@+id/idBtnSendData"
      android:layout_width="match_parent"
      android:layout_height="wrap_content"
      android:layout_below="@id/idEdtEmployeeAddress"
      android:layout_margin="10dp"
      android:text="@string/add_employee_details"
      android:textAllCaps="false" />

</RelativeLayout>
```

Now Create a Class File Named "EmployeeInfo.kt" And Add The Following Code to it-

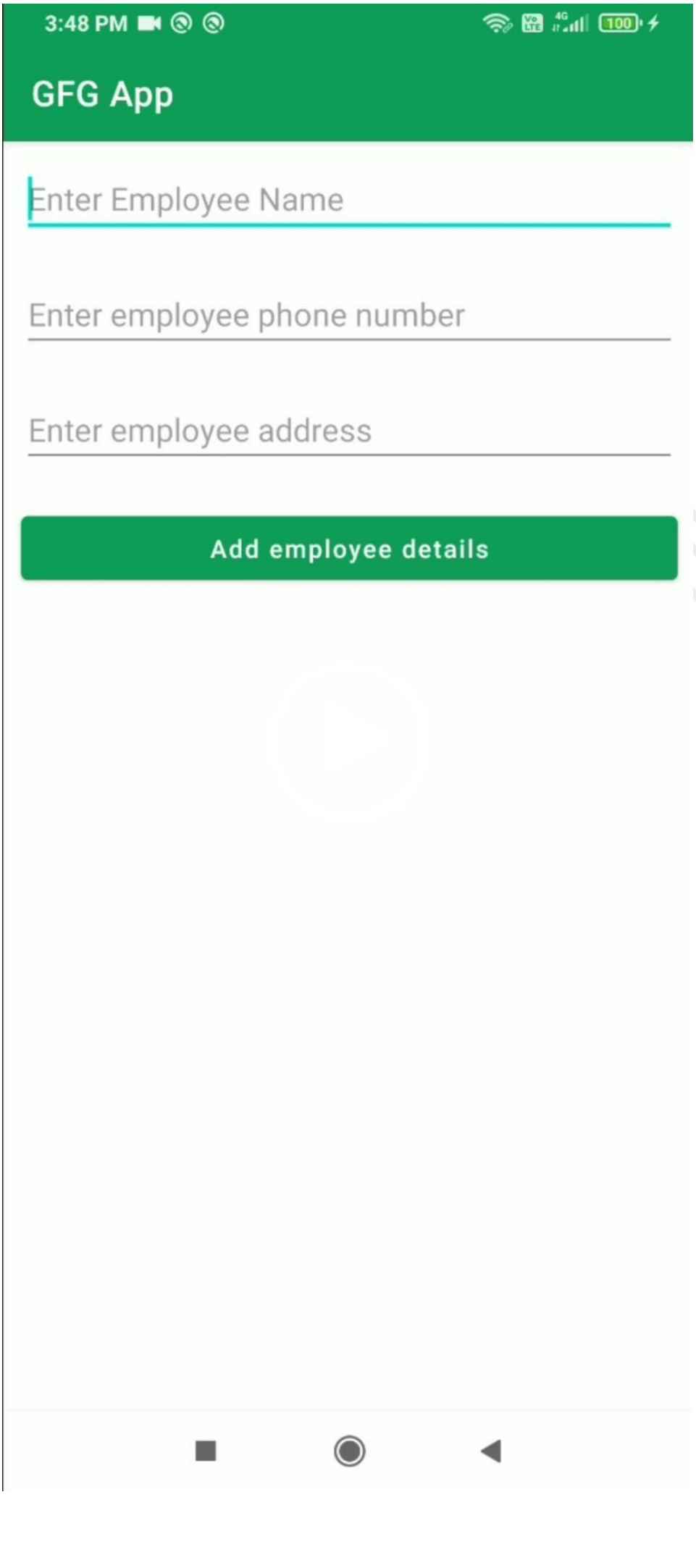
```
package com.example.firebaseapplication

class EmployeeInfo
{

    var employeeName: String? = null


    var employeeContactNumber: String? = null


    var employeeAddress: String? = null
}
```

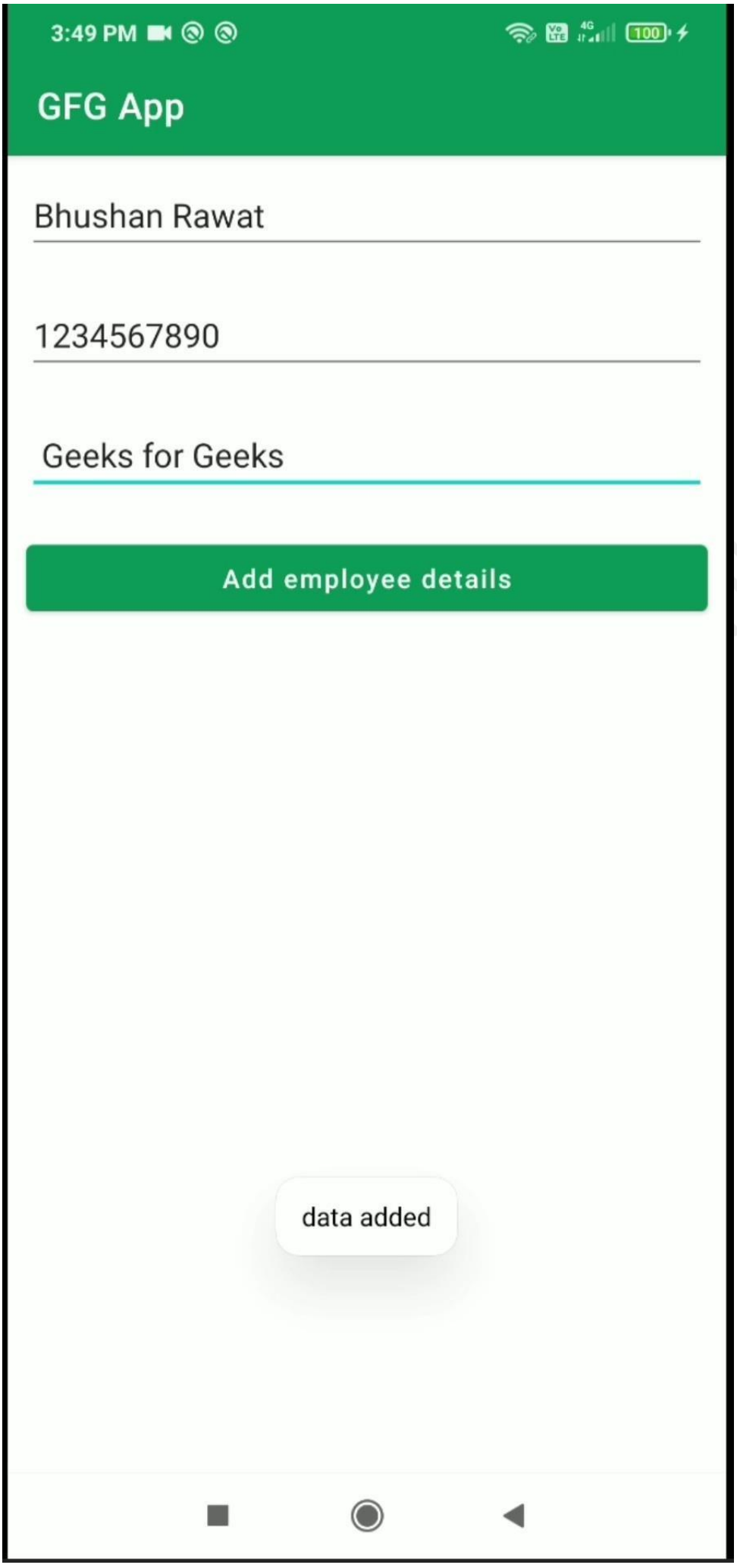
So Now add the Following Code to MainActivity.kt File-

```
package com.example.firebaseapplication

import android.os.Bundle
import android.support.v7.app.AppCompatActivity
import android.text.TextUtils
import android.view.View
import android.widget.Button
import android.widget.EditText
import android.widget.Toast
import com.google.firebase.database.DataSnapshot
import com.google.firebase.database.DatabaseError
import com.google.firebase.database.DatabaseReference
import com.google.firebase.database.FirebaseDatabase
import com.google.firebase.database.ValueEventListener


class MainActivity : AppCompatActivity() {
    // creating variables for
    // EditText and buttons.
    private var employeeNameEdt: EditText? = null
    private var employeePhoneEdt: EditText? = null
    private var employeeAddressEdt: EditText? = null
    private var sendDatabtn: Button? = null

    // creating a variable for our
    // Firebase Database.
    var firebaseDatabase: FirebaseDatabase? = null

    // creating a variable for our Database
    // Reference for Firebase.
    var databaseReference: DatabaseReference? = null

    // creating a variable for
    // our object class
    var employeeInfo: EmployeeInfo? = null
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_main)

        // initializing our edittext and button
        employeeNameEdt = findViewById(R.id.idEdtEmployeeName)
        employeePhoneEdt = findViewById(R.id.idEdtEmployeePhoneNumber)
        employeeAddressEdt = findViewById(R.id.idEdtEmployeeAddress)

        // below line is used to get the
```

```
    // instance of our FIrebase database.
    firebaseDatabase = FirebaseDatabase.getInstance()

    // below line is used to get reference for our database.
    databaseReference = firebaseDatabase.getReference("EmployeeInfo")

    // initializing our object
    // class variable.
    employeeInfo = EmployeeInfo()
    sendDatabtn = findViewById(R.id.idBtnSendData)

    // adding on click listener for our button.
    this.sendDatabtn.setOnClickListener(View.OnClickListener {
        // getting text from our edittext fields.
        val name = this.employeeNameEdt.getText().toString()
        val phone = this.employeePhoneEdt.getText().toString()
        val address = this.employeeAddressEdt.getText().toString()

        // below line is for checking whether the
        // edittext fields are empty or not.
        if (TextUtils.isEmpty(name) && TextUtils.isEmpty(phone) && TextUtils.isEmpty(address)) {
            // if the text fields are empty
            // then show the below message.
            Toast.makeText(this@MainActivity, "Please add some data.", Toast.LENGTH_SHORT)
                .show()
        } else {
            // else call the method to add
            // data to our database.
            addDatatoFirebase(name, phone, address)
        }
    })
}

private fun addDatatoFirebase(name: String, phone: String, address: String) {
    // below 3 lines of code is used to set
    // data in our object class.
    employeeInfo!!.employeeName = name
    employeeInfo!!.employeeContactNumber = phone
    employeeInfo!!.employeeAddress = address

    // we are use add value event listener method
    // which is called with database reference.
    databaseReference.addValueEventListener(object : ValueEventListener() {
        fun onDataChange(snapshot: DataSnapshot) {
            // inside the method of on Data change we are setting
            // our object class to our database reference.
            // data base reference will sends data to firebase.
            databaseReference.setValue(employeeInfo!!)

            // after adding this data we are showing toast message.
            Toast.makeText(this@MainActivity, "data added", Toast.LENGTH_SHORT).show()
        }

        fun onCancelled(error: DatabaseError) {
            // if the data is not added or it is cancelled then
            // we are displaying a failure toast message.
            Toast.makeText(this@MainActivity, "Fail to add data $error", Toast.LENGTH_SHORT)
                .show()
        }
    })
}
```

OUTPUT-


3:48 PM
GFG App
Enter Employee Name
Enter employee phone number
Enter employee address
Add employee details


3:49 PM
GFG App
Bhushan Rawat
1234567890
Geeks for Geeks
Add employee details
data added