Practical no 2

i.Create an Android application to design screens using different layouts and UI including Button, Edittext, Textview, Radio Button etc.

activity_main.xml

```
<RelativeLayout
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    xmlns:android="http://schemas.android.com/apk/res/android">

<Button
    android:id="@+id/button1"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Top Left Button"
    android:layout_alignParentLeft="true"
    android:layout_alignParentTop="true"/>
<Button
    android:id="@+id/button2"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Top Right Button"
    android:layout_alignParentTop="true"
    android:layout_alignParentRight="true"/>
<Button
    android:id="@+id/button3"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Bottom Left Button"
    android:layout_alignParentLeft="true"
    android:layout_alignParentBottom="true"/>
<Button
    android:id="@+id/button4"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Bottom Right Button"
    android:layout_alignParentRight="true"
    android:layout_alignParentBottom="true"/>
<Button
    android:id="@+id/button5"
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:text="Middle Button"
    android:layout_centerVertical="true"
    android:layout_centerHorizontal="true"/>

</RelativeLayout>
```

Output:


TOP LEFT BUTTON
TOP RIGHT BUTTON
MIDDLE BUTTON
BOTTOM LEFT BUTTON
BOTTOM RIGHT BUTTON

ii. Write an android application demonstrating response to event/user interaction for

a. Checkbox b. Radio button c. Button d. Spinner

activity_main.xml

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

    <TableRow>
        <EditText android:id="@+id/Hobby"
            android:layout_width="100dp"
            android:layout_height="100dp"
            android:text="Hobby"/>
        <CheckBox
            android:id="@+id/Read"
            android:layout_width="150dp"
            android:layout_height="50dp"
            android:text="Read" />
        <CheckBox
            android:id="@+id/Play"
            android:layout_width="150dp"
            android:layout_height="50dp"
            android:text="Play" />
    </TableRow>
    <TableRow>
        <EditText android:id="@+id/Gender"
            android:layout_width="100dp"
            android:layout_height="100dp"
            android:text="Gender"/>

        <RadioGroup
            android:layout_width="wrap_content"
            android:layout_height="wrap_content">

            <RadioButton
```

```
                <RadioButton
                    android:id="@+id/Male"
                    android:layout_width="150dp"
                    android:layout_height="50dp"
                    android:text="Male" />

                <RadioButton
                    android:id="@+id/Female"
                    android:layout_width="150dp"
                    android:layout_height="50dp"
                    android:text="Female" />
            </RadioGroup>
        </TableRow>
        <TableRow>
            <EditText android:id="@+id/State"
                android:layout_width="100dp"
                android:layout_height="100dp"
                android:text="State"/>
            <Spinner
                android:id="@+id/spinner"
                android:layout_width="150dp"
                android:layout_height="50dp"
                android:spinnerMode="dialog"
                android:layout_marginBottom="8dp"
                android:layout_marginEnd="8dp"
                android:layout_marginStart="8dp"
                android:layout_marginTop="8dp"   />
        </TableRow>
        <TableRow>
            <Button
                android:id="@+id/Submit"
                android:layout_width="150dp"
                android:layout_height="50dp"
                android:text="Submit"/>
        </TableRow>
    </TableLayout>
```

MainActivity.kt

MainActivity.kt × activity_main.xml

```
class MainActivity : ComponentActivity(), AdapterView.OnItemSelectedListener {
    lateinit var submit:Button
    lateinit var spinner:Spinner
    lateinit var read:CheckBox
    lateinit var play:CheckBox
    lateinit var male:RadioButton
    lateinit var female:RadioButton
    var states = arrayOf("Gujarat", "Goa", "Punjab")

    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_main)
        submit=findViewById(R.id.Submit)
        spinner=findViewById(R.id.spinner)
        read=findViewById(R.id.Read)
        play=findViewById(R.id.Play)
        male=findViewById(R.id.Male)
        female=findViewById(R.id.Female)

        submit.setOnClickListener{ it: View!
            Toast.makeText( context: this@MainActivity,  text: "Submitted Date", Toast.LENGTH_LONG).show()
        }

        read.setOnClickListener{ it: View!
            Toast.makeText( context: this@MainActivity,  text: "Read Selected", Toast.LENGTH_LONG).show()
        }

        play.setOnClickListener{ it: View!
            Toast.makeText( context: this@MainActivity,  text: "Play Selected", Toast.LENGTH_LONG).show()
        }
```

MainActivity.kt × activity_main.xml

```
        }

        play.setOnClickListener{ it: View!
            Toast.makeText( context: this@MainActivity,  text: "Play Selected", Toast.LENGTH_LONG).show()
        }

        male.setOnClickListener{ it: View!
            Toast.makeText( context: this@MainActivity,  text: "Male Selected", Toast.LENGTH_LONG).show()
        }

        female.setOnClickListener{ it: View!
            Toast.makeText( context: this@MainActivity,  text: "Female Selected", Toast.LENGTH_LONG).show(
        }

        spinner!!.setOnItemSelectedListener(this)
        val aa = ArrayAdapter( context: this, android.R.layout.simple_spinner_item, states)
        aa.setDropDownViewResource(android.R.layout.simple_spinner_dropdown_item)
        spinner!!.setAdapter(aa)


    }

    override fun onItemSelected(parent: AdapterView<*>?, view: View?, position: Int, id: Long) {
        Toast.makeText( context: this@MainActivity,  text: "Selected"+states[position], Toast.LENGTH_LONG)
    }

    override fun onNothingSelected(parent: AdapterView<*>?) {
        Toast.makeText( context: this@MainActivity,  text: "Nothing Selected", Toast.LENGTH_LONG).show()
    }
}
```

Practical no 4

i. Create an Android application to demonstrate implicit and explicit intents

activity_main.xml

MainActivity.kt | AndroidManifest.xml | activity_main.xml | activity_h

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

    <EditText
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Enter URL Here"
        android:id="@+id/url"/>

    <Button
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:id="@+id/search"
        android:text="Search"/>

    <Button
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:id="@+id/hello"
        android:text="Click for Hello"/>

</LinearLayout>
```

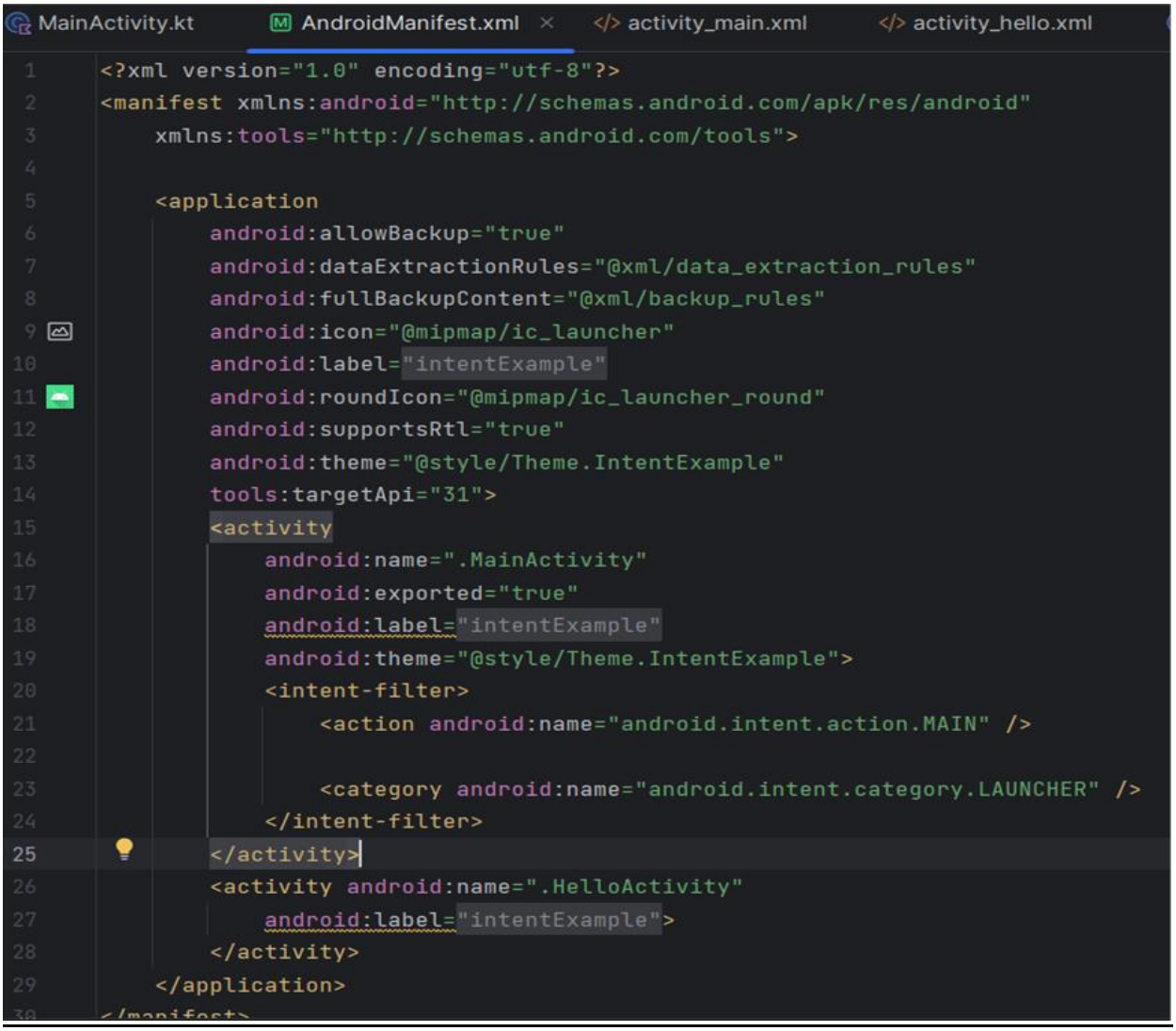
MainActivity.kt

MainActivity.kt | AndroidManifest.xml | activity_main.xml | activity_h

```
import ...

class MainActivity : ComponentActivity() {
    lateinit var url:EditText
    lateinit var search:Button
    lateinit var hello:Button
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_main)

        url=findViewById(R.id.url)
        search=findViewById(R.id.search)
        hello=findViewById(R.id.hello)

        search.setOnClickListener{ it: View!
            val url = url.text.toString()
            val urlIntent = Intent(Intent.ACTION_VIEW, Uri.parse(url))
            startActivity(urlIntent)
        }

        hello.setOnClickListener{
            val i = Intent(applicationContext,HelloActivity::class.java)
            startActivity(i)
        }

    }
}
```

val urlIntent: Intent
intentExample.app.main

AndroidManifest.xml



ii. Create an application to demonstrate shared preferences

activity_main.xml

```
<?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">

  <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>
```

MainActivity.kt

```
package com.example.spexample

import android.os.Bundle
import android.widget.EditText
import androidx.activity.ComponentActivity
import androidx.activity.compose.setContent
import androidx.appcompat.app.AppCompatActivity
import androidx.compose.foundation.layout.fillMaxSize
import androidx.compose.material3.MaterialTheme
import androidx.compose.material3.Surface
import androidx.compose.material3.Text
import androidx.compose.runtime.Composable
import androidx.compose.ui.Modifier
import androidx.compose.ui.tooling.preview.Preview
import com.example.spexample.ui.theme.SPExampleTheme

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)
    }

    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:

Practical no 5

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

AirplaneModeChaneReceiver.kt

MainActivity.kt | AirplaneModeChangeReceiver.kt | AndroidManifest.xml | activity_main.xml

```kotlin
package com.example.broadcasttest

import android.content.BroadcastReceiver
import android.content.Context
import android.content.Intent
import android.widget.Toast

class AirplaneModeChangeReceiver : BroadcastReceiver() {

    override fun onReceive(context: Context?, intent: Intent?) {

        val isAirplaneModeEnabled = intent?.getBooleanExtra( name: "state",  defaultValue: false) ?: return

        if (isAirplaneModeEnabled) {

            Toast.makeText(context,  text: "Airplane Mode Enabled", Toast.LENGTH_LONG).show()
        } else {

            Toast.makeText(context,  text: "Airplane Mode Disabled", Toast.LENGTH_LONG).show()
        }
    }

}
```

MainActivity.kt

MainActivity.kt | AirplaneModeChangeReceiver.kt | AndroidManifest.xml | activity_n

```kotlin
package com.example.broadcasttest

import android.annotation.SuppressLint
import android.os.Bundle
import android.content.Intent
import android.content.IntentFilter
import androidx.appcompat.app.AppCompatActivity
import com.example.locationtest.R

class MainActivity : AppCompatActivity() {
    lateinit var receiver: AirplaneModeChangeReceiver
    @SuppressLint("SuspiciousIndentation")
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_main)
        receiver = AirplaneModeChangeReceiver()


        IntentFilter(Intent.ACTION_AIRPLANE_MODE_CHANGED).also { it: IntentFilter

            registerReceiver(receiver, it)
        }
    }


    override fun onStop() {
        super.onStop()
        unregisterReceiver(receiver)
    }
}
```

Output:

ii. Create an Android application to create and use services.

activity_main.xml

MainActivity.kt | AndroidManifest.xml | MyService.kt | activity_main.xml

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

    <Button
   android:id="@+id/button"
   android:layout_width="wrap_content"
   android:layout_height="wrap_content"
   android:text="Click here to start background Service"
   android:textSize="16sp"
   android:textStyle="bold" />

</LinearLayout>
```

MyService.kt

MainActivity.kt | AndroidManifest.xml | MyService.kt | activity_main.xml

```
package com.example.servicetest

import android.app.Service
import android.content.Intent
import android.os.IBinder
import android.widget.Toast

class MyService : Service() {
    override fun onStartCommand(intent: Intent, flags: Int, startId: Int): Int {
        onTaskRemoved(intent)
        Toast.makeText(
        applicationContext, text: "This is a Service running in Background",
        Toast.LENGTH_SHORT
        ).show()
        return START_STICKY
    }
    override fun onBind(intent: Intent): IBinder? {
        // TODO: Return the communication channel to the service.
        throw UnsupportedOperationException("Not yet implemented")
    }
    override fun onTaskRemoved(rootIntent: Intent) {
        val restartServiceIntent = Intent(applicationContext, this.javaClass)
        restartServiceIntent.setPackage(packageName)
        startService(restartServiceIntent)
        super.onTaskRemoved(rootIntent)
    }
}
```

MainActivity.kt

MainActivity.kt | AndroidManifest.xml | MyService.kt | activity_main.xml

```kotlin
package com.example.servicetest

import android.content.Intent
import android.os.Bundle
import android.widget.Button
import androidx.activity.ComponentActivity


class MainActivity : ComponentActivity() {
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_main)
         val button: Button = findViewById(R.id.button)
      button.setOnClickListener { startService(Intent(applicationContext, MyService::class.java)) }
    }
}
```

Android.Manifest.xml

MainActivity.kt | AndroidManifest.xml | MyService.kt | activity_main.xml

```xml
<?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="servicetest"
        android:roundIcon="@mipmap/ic_launcher_round"
        android:supportsRtl="true"
        android:theme="@style/Theme.Servicetest"
        tools:targetApi="31">
        <activity
            android:name=".MainActivity"
            android:exported="true"
            android:label="servicetest"
            android:theme="@style/Theme.Servicetest">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />

                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
        <service
      android:name=".MyService"
      android:enabled="true"
      android:exported="true" />
    </application>
</manifest>
```

Output:


CLICK HERE TO START BACKGROUND SERVICE


Play Store
Gmail
Photos
YouTube

Practical no 6

i.Create an Android application to demonstrate XML based animation .

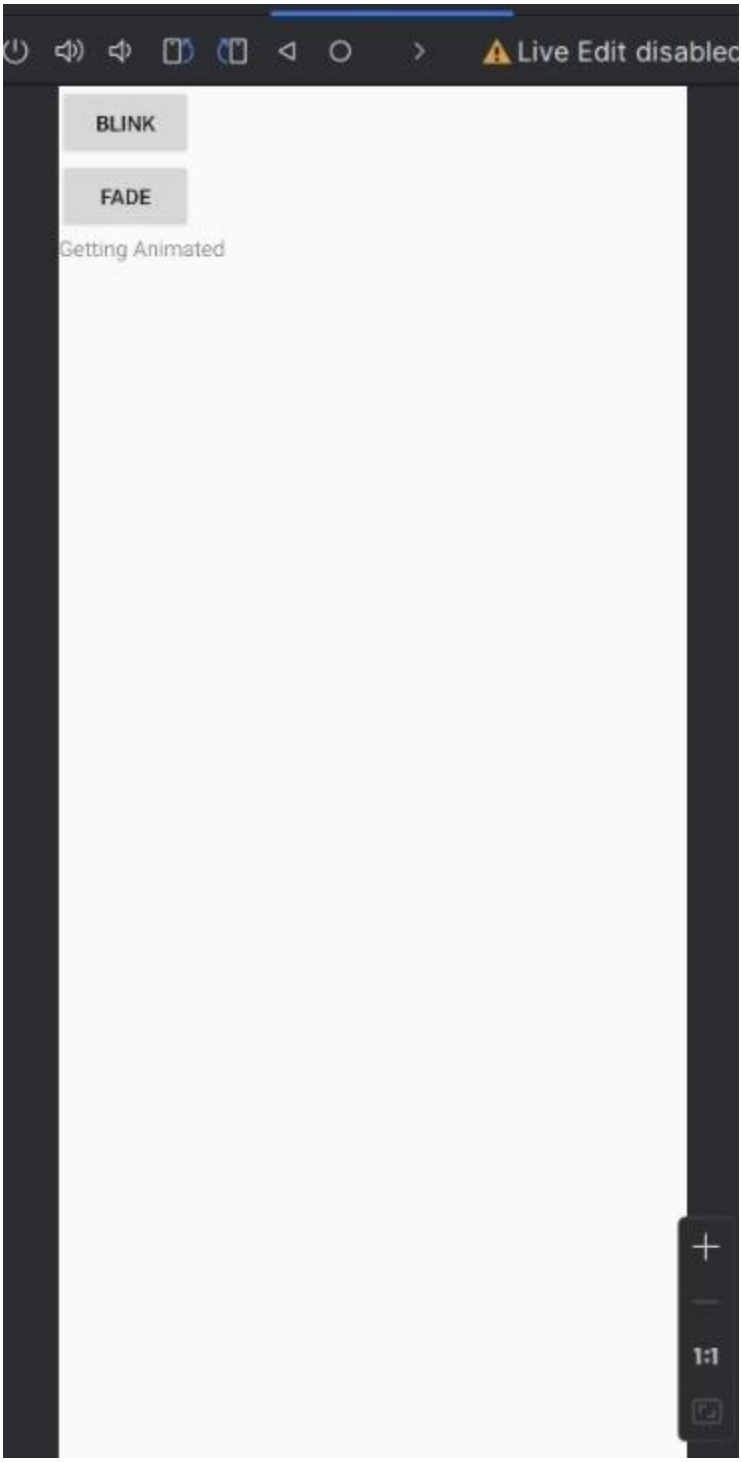
activity_main.xml

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

   <Button
      android:id="@+id/BTNblink"
      android:layout_width="wrap_content"
      android:layout_height="wrap_content"
       android:text="Blink"
 />

   <!--To start the fading animation of the image-->
   <Button
      android:id="@+id/BTNfade"
      android:layout_width="wrap_content"
      android:layout_height="wrap_content"

      android:text="Fade"
      />

   <TextView android:id="@+id/txt"
      android:layout_width="wrap_content"
      android:layout_height="wrap_content"
      android:text="Getting Animated"/>

</LinearLayout>
```

MainActivity.kt

```
package com.example.animatetest

import android.annotation.SuppressLint
import android.os.Bundle
import android.widget.Button
import androidx.activity.ComponentActivity
import android.view.animation.AnimationUtils;
import android.widget.TextView


class MainActivity : ComponentActivity() {
   @SuppressLint("MissingInflatedId")
   override fun onCreate(savedInstanceState: Bundle?) {
      super.onCreate(savedInstanceState)
      setContentView(R.layout.activity_main)
      val blinkBTN = findViewById<Button>(R.id.BTNblink);
      val fadeBTN = findViewById<Button>(R.id.BTNfade);
      val txt=findViewById<TextView>(R.id.txt)

      blinkBTN.setOnClickListener {
         val a=AnimationUtils.loadAnimation(applicationContext,R.anim.blink_animation)
         txt.startAnimation(a)
      }

      fadeBTN.setOnClickListener {
         val a=AnimationUtils.loadAnimation(applicationContext,R.anim.fade_animation)
         txt.startAnimation(a)
      }
   }
}
```

Output:



ii. Create an Android application to display canvas and allow the user to draw on it.

activity_main.xml

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

   <ImageView
      android:id="@+id/image_view_1"
      android:layout_width="match_parent"
      android:layout_height="match_parent"
     tools:ignore="ContentDescription"
      android:background="@color/black"/>


</LinearLayout>
```

MainActivity.kt

```
package com.example.canvastest


import android.os.Bundle
import android.view.View

import androidx.activity.ComponentActivity


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.view.MotionEvent
```

```
import android.widget.ImageView
import androidx.activity.compose.setContent
import androidx.annotation.RequiresApi
import androidx.compose.foundation.layout.fillMaxSize
import androidx.compose.material3.MaterialTheme
import androidx.compose.material3.Surface
import androidx.compose.material3.Text
import androidx.compose.runtime.Composable
import androidx.compose.ui.Modifier
import androidx.compose.ui.tooling.preview.Preview
import com.example.canvastest.ui.theme.CanvastestTheme

class MainActivity : ComponentActivity() ,View.OnTouchListener {
    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)


    }

    @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
    }

}
```

MainActivity.kt

```
package com.example.audiotest

import android.media.MediaPlayer
import android.os.Bundle
import android.widget.Button
import androidx.activity.ComponentActivity
```

```
class MainActivity : ComponentActivity() {
   override fun onCreate(savedInstanceState: Bundle?) {
      super.onCreate(savedInstanceState)
      setContentView(R.layout.activity_main)
      val mediaPlayer: MediaPlayer = MediaPlayer.create(applicationContext, R.raw.music1)
      val bPlay: Button = findViewById(R.id.playButton)
      val bPause: Button = findViewById(R.id.pauseButton)
      val bStop: Button = findViewById(R.id.stopButton)

      bPlay.setOnClickListener {
                mediaPlayer.start()
      }

      bPause.setOnClickListener {

         mediaPlayer.pause()
      }

      bStop.setOnClickListener {

         mediaPlayer.stop()
         mediaPlayer.prepare()
      }
   }
}
```

Output:


Running Devices
Medium P...
Live Edit disabled
1:1
4:25
LF
UTF-8
4 spaces

Practical no 7

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

activity_main.xml

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

   <TextView
      android:id="@+id/headingText"
      android:layout_width="wrap_content"
      android:layout_height="wrap_content"
      android:layout_centerHorizontal="true"
      android:layout_marginTop="32dp"
      android:text="MEDIA PLAYER"
      android:textSize="18sp"
      android:textStyle="bold" />

   <LinearLayout
      android:layout_width="match_parent"
      android:layout_height="wrap_content"
      android:layout_below="@id/headingText"
      android:layout_marginTop="16dp"
      android:gravity="center_horizontal">

      <Button
         android:id="@+id/stopButton"
         android:layout_width="wrap_content"
         android:layout_height="wrap_content"
         android:layout_marginEnd="8dp"

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

```
    <Button
        android:id="@+id/playButton"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginEnd="8dp"

        android:text="PLAY"
         />

    <Button
        android:id="@+id/pauseButton"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"

        android:text="PAUSE"
       />

  </LinearLayout>

</LinearLayout>
```

MainActivity.kt

```
package com.example.audiotest

import android.media.MediaPlayer
import android.os.Bundle
import android.widget.Button
import androidx.activity.ComponentActivity

class MainActivity : ComponentActivity() {
  override fun onCreate(savedInstanceState: Bundle?) {
    super.onCreate(savedInstanceState)
    setContentView(R.layout.activity_main)
    val mediaPlayer: MediaPlayer = MediaPlayer.create(applicationContext, R.raw.music1)
    val bPlay: Button = findViewById(R.id.playButton)
    val bPause: Button = findViewById(R.id.pauseButton)
    val bStop: Button = findViewById(R.id.stopButton)
```

```
        bPlay.setOnClickListener {
                mediaPlayer.start()
        }

        bPause.setOnClickListener {

            mediaPlayer.pause()
        }

        bStop.setOnClickListener {

            mediaPlayer.stop()
            mediaPlayer.prepare()
        }
    }
}
```

Output:

ii. Create an Android application to display canvas and allow the user to draw on it.

activity_main.xml

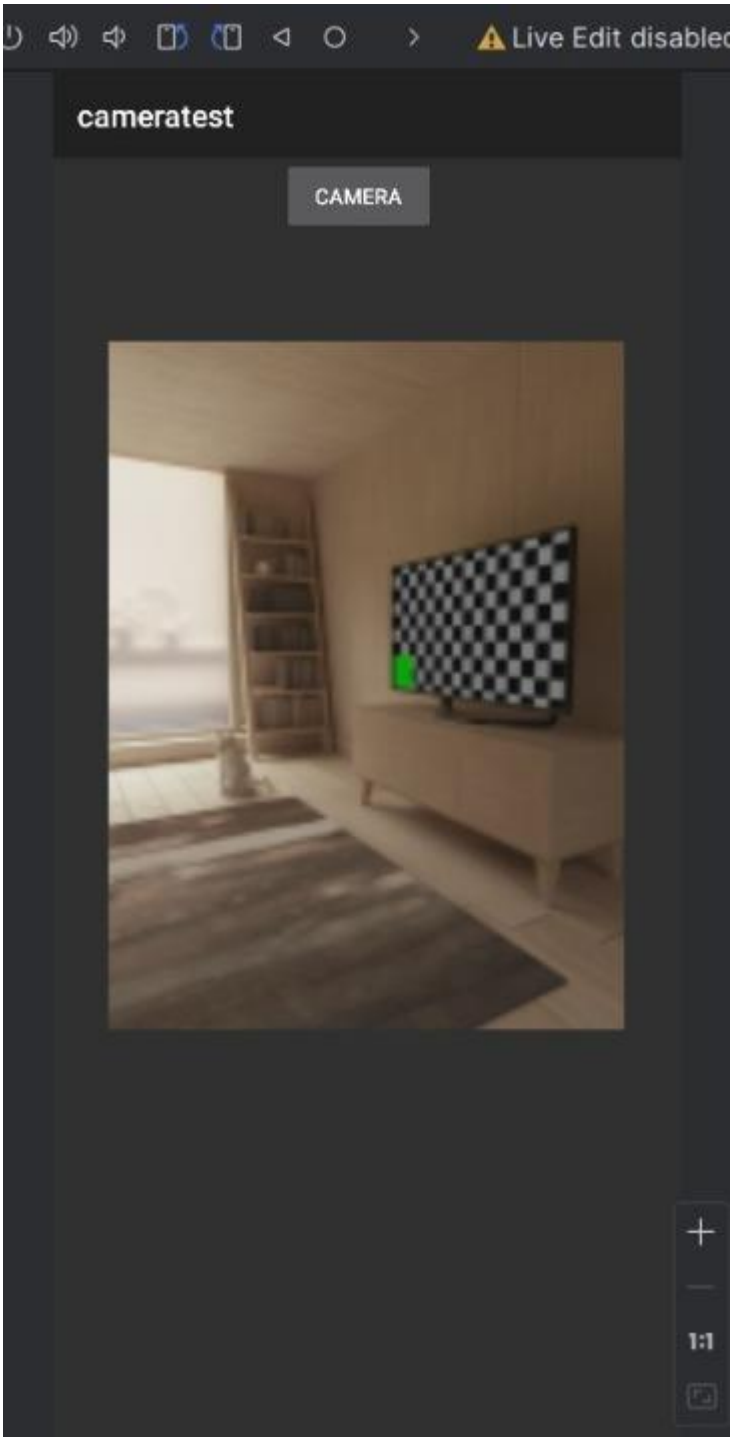
```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
   android:orientation="vertical"
   android:layout_width="match_parent"
   android:layout_height="match_parent">

   <Button
      android:id="@+id/camera_button"
      android:layout_width="100dp"
      android:layout_height="50dp"
      android:layout_marginStart="150dp"
      android:text="Camera" />

   <!-- add ImageView to display the captured image -->
   <ImageView
      android:id="@+id/click_image"
      android:layout_width="350dp"
      android:layout_height="450dp"
      android:layout_marginStart="30dp"
      android:layout_marginTop="70dp"
      android:layout_marginBottom="10dp" />

</LinearLayout>
```

MainActivity.kt

```
package com.example.cameratest

import android.content.Intent
import android.graphics.Bitmap
import android.os.Bundle
import android.provider.MediaStore
import android.view.View
import android.widget.Button
import android.widget.ImageView
import androidx.appcompat.app.AppCompatActivity
```

```
class MainActivity : AppCompatActivity() {

    private lateinit var cameraOpenId: Button
    lateinit var clickImageId: ImageView

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

        cameraOpenId = findViewById(R.id.camera_button)
        clickImageId = findViewById(R.id.click_image)

        cameraOpenId.setOnClickListener(View.OnClickListener { v: View? ->
            // Create the camera_intent ACTION_IMAGE_CAPTURE it will open the camera for
capture the image
            val cameraIntent = Intent(MediaStore.ACTION_IMAGE_CAPTURE)
            // Start the activity with camera_intent, and request pic id
            startActivityForResult(cameraIntent, pic_id)
        })

    }

    // This method will help to retrieve the image
    override fun onActivityResult(requestCode: Int, resultCode: Int, data: Intent?) {
        super.onActivityResult(requestCode, resultCode, data)
        // Match the request 'pic id with requestCode
        if (requestCode == pic_id) {
            // BitMap is data structure of image file which store the image in memory
            val photo = data!!.extras!!["data"] as Bitmap?
            // Set the image in imageview for display
            clickImageId.setImageBitmap(photo)
        }
    }

    companion object {
        // Define the pic id
        private const val pic_id = 123
    }
}
```

Output:



Practical no 8

ii. Create an Android application to demonstrate the different types of menus.

a. Pop-up Menu b. Context Menu c. Option Menu

activity_main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
   android:orientation="vertical"
   android:layout_width="match_parent"
   android:layout_height="match_parent">
  <androidx.appcompat.widget.Toolbar
     android:layout_width="match_parent"
     android:layout_height="wrap_content"
     android:id="@+id/t1"/>


</LinearLayout>
```

opitions_menu1.xml

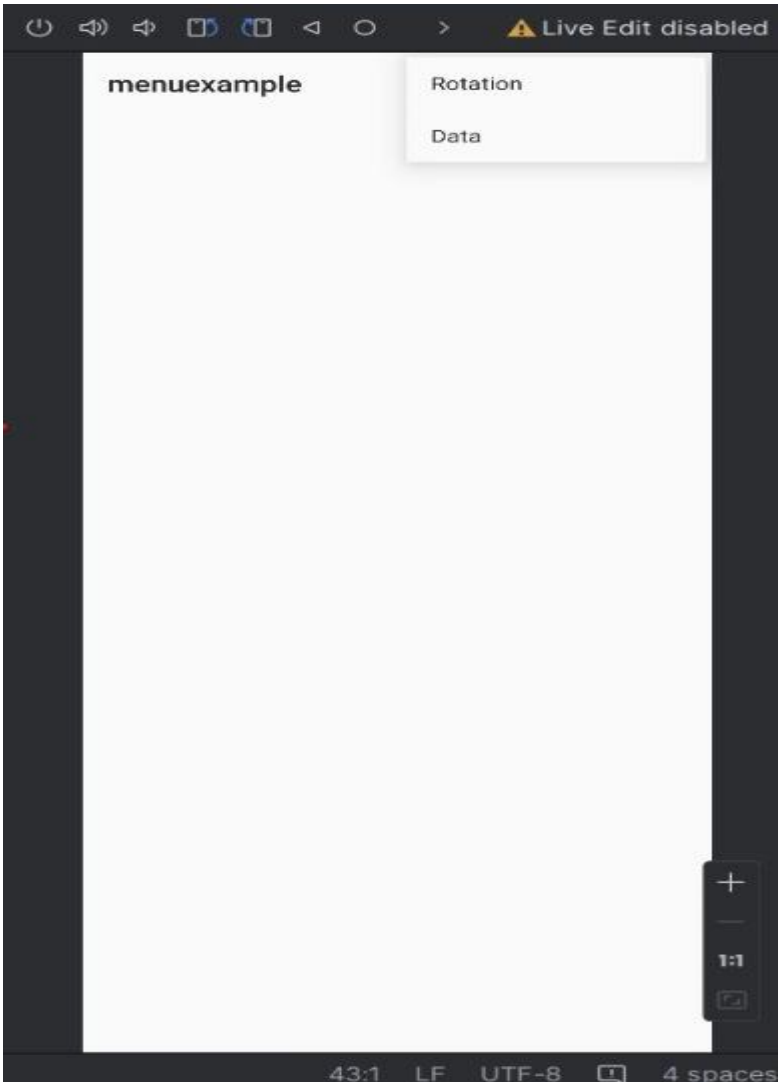
```
<?xml version="1.0" encoding="utf-8"?>
<menu xmlns:android="http://schemas.android.com/apk/res/android">
<item android:icon="@drawable/baseline_3d_rotation_24"
   android:title="Rotation"
   android:id="@+id/item1"
   android:showAsAction="always"/>


   <item android:id="@+id/item2"
      android:icon="@drawable/baseline_4g_mobiledata_24"
      android:title="Data"
      android:showAsAction="ifRoom"/>
</menu>
```

MainActivity.kt

```
package com.example.menuexample

import android.annotation.SuppressLint
import android.os.Bundle
import android.view.Menu
import android.view.MenuItem
import android.widget.Toast

import androidx.activity.ComponentActivity
import androidx.activity.compose.setContent
import androidx.appcompat.app.AppCompatActivity
import androidx.compose.foundation.layout.fillMaxSize
import androidx.compose.material3.MaterialTheme
import androidx.compose.material3.Surface
import androidx.compose.material3.Text
import androidx.compose.runtime.Composable
import androidx.compose.ui.Modifier
import androidx.compose.ui.tooling.preview.Preview
import com.example.menuexample.ui.theme.MenuexampleTheme

class MainActivity : AppCompatActivity() {
    @SuppressLint("MissingInflatedId")
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_main)
        setSupportActionBar(findViewById(R.id.t1))

    }

    override fun onCreateOptionsMenu(menu: Menu?): Boolean {
        menuInflater.inflate(R.menu.options_menu1,menu)
        return true
    }

    override fun onOptionsItemSelected(item: MenuItem): Boolean {
        when (item.itemId){
            R.id.item1 -> Toast.makeText(this,"About Selected",Toast.LENGTH_SHORT).show()
            R.id.item2 -> Toast.makeText(this,"Settings Selected",Toast.LENGTH_SHORT).show()
```

```
        }
        return super.onOptionsItemSelected(item)
    }
}
```

Output:



Practical no 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

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
   android:orientation="vertical"
   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:text="Current Location:"
        />
     <TextView
        android:id="@+id/tvLatitude"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_marginTop="20dp"
        android:text="Latitude: -"
        />
     <TextView
        android:id="@+id/tvLongitude"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_marginTop="10dp"
        android:text="Longitude: -"
        />
```

```
    <TextView
        android:id="@+id/tvProvider"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"

        android:layout_marginTop="10dp"
        android:text="Provider: -"
        />
    <Button
        android:id="@+id/btOpenMap"
        android:layout_width="150dp"
        android:layout_height="wrap_content"

        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:layout_margin="30dp"
    android:text="Get Current Location"
    android:textColor="@android:color/white"
    android:layout_alignParentBottom="true"
    />
</LinearLayout>
```

MainActivity.kt

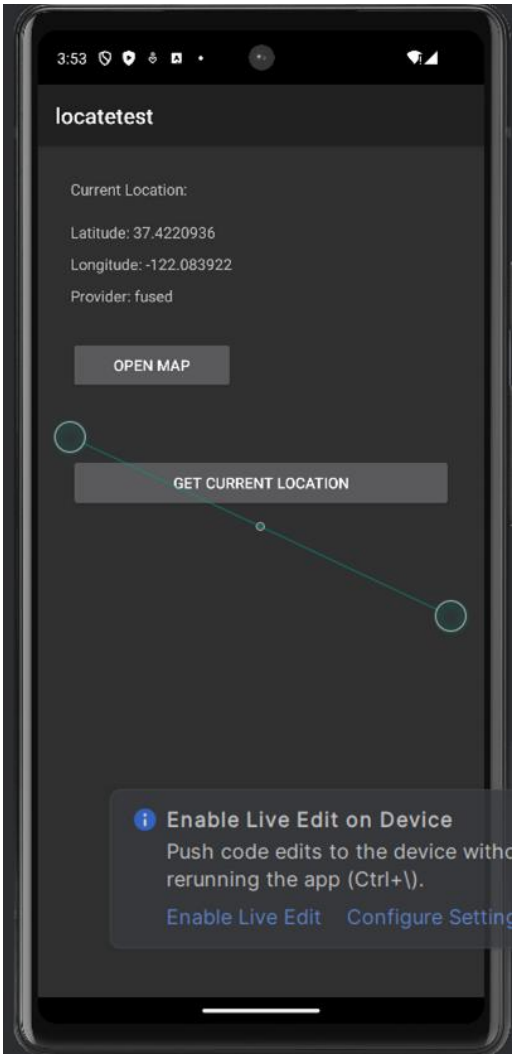
```
package com.example.locatetest

import android.os.Bundle
import android.Manifest
import android.content.Intent
import android.content.pm.PackageManager
import android.net.Uri
```

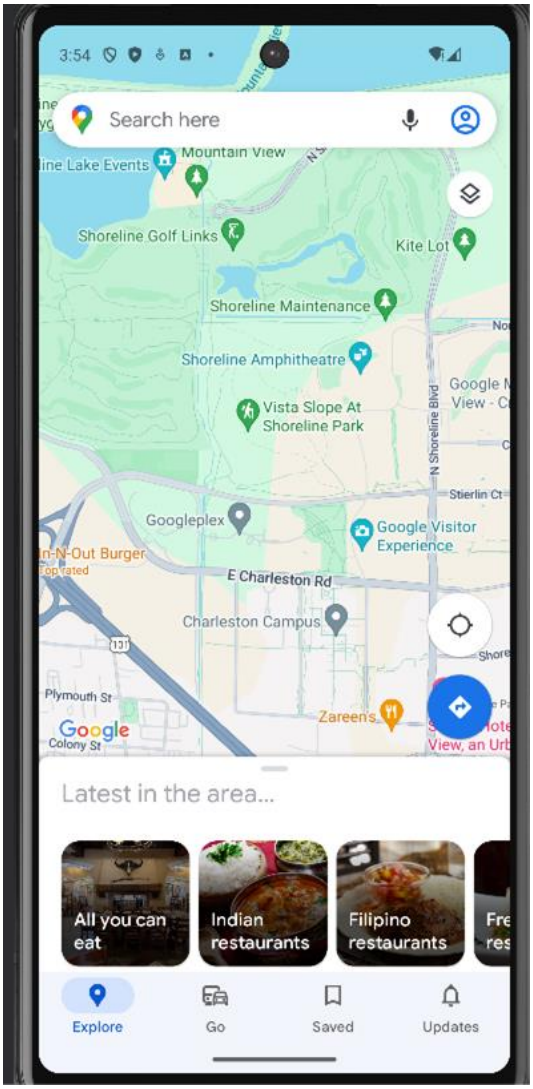
```
import android.view.View
import android.widget.Button
import android.widget.TextView
import android.widget.Toast
import androidx.appcompat.app.AppCompatActivity
import androidx.core.app.ActivityCompat
import com.google.android.gms.location.FusedLocationProviderClient
import com.google.android.gms.location.LocationServices

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)
        val btGetLocation=findViewById<Button>(R.id.btGetLocation)
        val btOpenMap=findViewById<Button>(R.id.btOpenMap)

        fusedLocationClient = LocationServices.getFusedLocationProviderClient(this)
       btGetLocation.setOnClickListener {
          getCurrentLocation()
       }
       btOpenMap.setOnClickListener {
          openMap()
       }
    }

    private fun getCurrentLocation() {
        val tvLatitude=findViewById<TextView>(R.id.tvLatitude)
        val tvLongitude=findViewById<TextView>(R.id.tvLongitude)
        val tvProvider=findViewById<TextView>(R.id.tvProvider)
        val btOpenMap=findViewById<Button>(R.id.btOpenMap)

        // 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
        }
        fusedLocationClient.lastLocation
            .addOnSuccessListener { location ->
                // getting the last known or current location
                latitude = location.latitude
                longitude = location.longitude
                tvLatitude.text = "Latitude: ${location.latitude}"
                tvLongitude.text = "Longitude: ${location.longitude}"
                tvProvider.text = "Provider: ${location.provider}"
                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
    ) {
        super.onRequestPermissionsResult(requestCode, permissions, grantResults)
        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)
    }


}
```

Output:





Practical no 10

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

activity_main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
   android:orientation="vertical"
   android:layout_width="match_parent"
   android:layout_height="match_parent">
   <!-- Edit text to enter name -->
   <EditText
      android:id="@+id/enterName"
      android:layout_width="match_parent"
      android:layout_height="wrap_content"
      android:hint="Enter Name"
      android:textSize="22sp"
      android:layout_margin="20sp"/>

   <!-- Edit text to enter age -->
   <EditText
      android:id="@+id/enterAge"
      android:layout_width="match_parent"
      android:layout_height="wrap_content"
      android:layout_margin="20sp"
      android:textSize="22sp"
      android:hint="Enter Age"/>

   <!-- Button to add Name -->
   <Button
      android:id="@+id/addName"
      android:layout_width="150sp"
      android:layout_gravity="center"
      android:background="@color/purple_200"
      android:text="Add Name"
      android:textColor="#ffffff"
      android:textSize="20sp"
      android:layout_height="wrap_content"
      android:layout_marginTop="20sp"/>
```

```
<!-- Button to print Name -->
<Button
   android:id="@+id/printName"
   android:layout_width="150sp"
   android:layout_gravity="center"
   android:background="@color/purple_200"
   android:text="Print Name"
   android:textColor="#ffffff"
   android:textSize="20sp"
   android:layout_height="wrap_content"
   android:layout_marginTop="20sp"/>

<LinearLayout
   android:layout_width="match_parent"
   android:layout_height="wrap_content">

   <!-- Text view to get all name -->
   <TextView
      android:id="@+id/Name"
      android:textAlignment="center"
      android:layout_weight="1"
      android:layout_width="match_parent"
      android:layout_height="wrap_content"
      android:layout_margin="20sp"
      android:text="Name\n\n"
      android:textSize="22sp"
      android:padding="10sp"
      android:textColor="#000000"/>

   <!-- Text view to get all ages -->
   <TextView
      android:layout_weight="1"
      android:id="@+id/Age"
      android:textAlignment="center"
      android:layout_width="match_parent"
      android:layout_height="wrap_content"
      android:layout_margin="20sp"
      android:text="Age\n\n"
      android:textSize="22sp"
      android:padding="10sp"
```

```
        android:textColor="#000000"/>

    </LinearLayout>
</LinearLayout>
```

DBHelper.kt

```
package com.example.sqldatabaseexample

import android.content.ContentValues
import android.content.Context
import android.database.Cursor
import android.database.sqlite.SQLiteDatabase
import android.database.sqlite.SQLiteOpenHelper

class DBHelper(context: Context, factory: SQLiteDatabase.CursorFactory?) :
    SQLiteOpenHelper(context, DATABASE_NAME, factory, DATABASE_VERSION){

    override fun onCreate(db: SQLiteDatabase) {
        // below is a sqlite query, where column names along with their data types is given
        val query = ("CREATE TABLE " + TABLE_NAME + " ("
                + ID_COL + " INTEGER PRIMARY KEY, " +
                NAME_COl + " TEXT," +
                AGE_COL + " TEXT" + ")")

        // we are calling sqlite Method for executing our query
        db.execSQL(query)
    }

    override fun onUpgrade(db: SQLiteDatabase, p1: Int, p2: Int) {
        // this method is to check if table already exists
        db.execSQL("DROP TABLE IF EXISTS " + TABLE_NAME)
        onCreate(db)
    }

    // This method is for adding data in our database
    fun addName(name : String, age : String ){

        // below we are creating a content values variable
        val values = ContentValues()
```

```
        // we are inserting our values in the form of key-value pair
        values.put(NAME_COl, name)
        values.put(AGE_COL, age)

        // here we are creating a writable variable of our database as we want to insert value in our
database
        val db = this.writableDatabase

        // all values are inserted into database
        db.insert(TABLE_NAME, null, values)

        // at last we are closing our database
        db.close()
    }

    // below method is to get all data from our database
    fun getName(): Cursor? {

        // here we are creating a readable variable of our database as we want to read value from it
        val db = this.readableDatabase

        // below code returns a cursor to read data from the database
        return db.rawQuery("SELECT * FROM " + TABLE_NAME, null)

    }

    companion object{
        // here we have defined variables for our database

        // below is variable for database name
        private val DATABASE_NAME = "MY_DATABASE"

        // below is the variable for database version
        private val DATABASE_VERSION = 1

        // below is the variable for table name
        val TABLE_NAME = "Student"

        // below is the variable for id column
        val ID_COL = "id"
```

```
        // below is the variable for name column
        val NAME_COl = "name"

        // below is the variable for age column
        val AGE_COL = "age"
    }

}
```

MainActivity.kt

```
package com.example.sqldatabaseexample

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

import androidx.activity.ComponentActivity
import androidx.activity.compose.setContent
import androidx.compose.foundation.layout.fillMaxSize
import androidx.compose.material3.MaterialTheme
import androidx.compose.material3.Surface
import androidx.compose.material3.Text
import androidx.compose.runtime.Composable
import androidx.compose.ui.Modifier
import androidx.compose.ui.tooling.preview.Preview
import com.example.sqldatabaseexample.ui.theme.SqlDatabaseExampleTheme

class MainActivity : AppCompatActivity() {
    lateinit var addName:Button
    lateinit var enterName:EditText
    lateinit var enterAge:EditText
    lateinit var printName:Button
    lateinit var Name:TextView
    lateinit var Age:TextView
```

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

    addName=findViewById(R.id.addName)
    enterName=findViewById(R.id.enterName)
    enterAge=findViewById(R.id.enterAge)
    printName=findViewById(R.id.printName)
    Name=findViewById(R.id.Name)
    Age=findViewById(R.id.Age)

    addName.setOnClickListener{

        // below we have created a new DBHelper class,and passed context to it
        val db = DBHelper(this, null)

        // creating variables for values in name and age edit texts
        val name = enterName.text.toString()
        val age = enterAge.text.toString()

        // calling method to add name to our database
        db.addName(name, age)

        // Toast to message on the screen
        Toast.makeText(this, name + " added to database", Toast.LENGTH_LONG).show()

        // at last, clearing edit texts
        enterName.text.clear()
        enterAge.text.clear()
    }

    printName.setOnClickListener{

        // creating a DBHelper class
        // and passing context to it
        val db = DBHelper(this, null)

        // below is the variable for cursor
        // we have called method to get
        // all names from our database
        // and add to name text view
```

```
        val cursor = db.getName()

        // moving the cursor to first position and
        // appending value in the text view
        cursor!!.moveToFirst()
        Name.append(cursor.getString(cursor.getColumnIndexOrThrow(DBHelper.NAME_COl)) +
"\n")
        Age.append(cursor.getString(cursor.getColumnIndexOrThrow(DBHelper.AGE_COL)) +
"\n")

        // moving our cursor to next
        // position and appending values
        while(cursor.moveToNext()){
            Name.append(cursor.getString(cursor.getColumnIndexOrThrow(DBHelper.NAME_COl))
+ "\n")
            Age.append(cursor.getString(cursor.getColumnIndexOrThrow(DBHelper.AGE_COL)) +
"\n")
        }

        // at last we close our cursor
        cursor.close()
    }

  }
}
```

Output:


Live Edit disabled
10:37
sqlDatabaseExample
Enter Name
Enter Age
ADD NAME
PRINT NAME
Name
Age
ashish
yadav
ashish
20
GIF
?123
1:1