

Practical # 1

Introduction to android, android studio ide application fundamentals

activity_main.xml

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

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

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

MainActivity.kt

```
package com.example.practical11
```

```
import android.content.Context
import android.net.ConnectivityManager
import androidx.appcompat.app.AppCompatActivity
import android.os.Bundle
import android.widget.Toast
```

```
class MainActivity : AppCompatActivity() {
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_main)
        val c=applicationContext.getSystemService(CONNECTIVITY_SERVICE) as
ConnectivityManager
        val networkInfo = c.activeNetworkInfo

        if(networkInfo!=null && networkInfo.isConnected)
        {
            if (networkInfo.type==ConnectivityManager.TYPE_MOBILE)
```

```

        {
            Toast.makeText(applicationContext,"Connected to
Mobile",Toast.LENGTH_LONG).show()
        }
        if (networkInfo.type==ConnectivityManager.TYPE_WIFI)
        {
            Toast.makeText(applicationContext,"Connected to
Wifi",Toast.LENGTH_LONG).show()
        }
    }
    else
    {
        Toast.makeText(applicationContext,"You are
offline",Toast.LENGTH_LONG).show()
    }
}
}

```

AndroidManifest.xml

```

<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.example.practical11"
    xmlns:tools="http://schemas.android.com/tools">

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

    <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:roundIcon="@mipmap/ic_launcher_round"
        android:supportRtl="true"
        android:theme="@style/Theme.Practical11"
        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>

```

```
        </intent-filter>
    </activity>
</application>

</manifest>
```

Practical # 2

Programing resources: color,theme, string, drawable, dimension,image

strings.xml

```
<resources>
    <string name="app_name">Android Resources</string>
    <string name="my_text">TYIT</string>

</resources>
```

colors.xml

```
<?xml version="1.0" encoding="utf-8"?>
<resources>
    <color name="colorPrimary">#FFEB3B</color>
    <color name="colorPrimaryDark">#18463F</color>
    <color name="colorAccent">#D81B60</color>

</resources>
```

themes.xml

```
<resources xmlns:tools="http://schemas.android.com/tools">
    <!-- Base application theme. -->
    <style name="Base.Theme.Practical21"
parent="Theme.AppCompat.DayNight.DarkActionBar">
        <!-- Customize your light theme here. -->
        <item name="colorPrimary">@color/colorPrimary</item>
        <item name="colorPrimaryDark">@color/colorPrimaryDark</item>
        <item name="colorAccent">@color/colorAccent</item>

    </style>

    <style name="Theme.Practical21" parent="Base.Theme.Practical21" />
</resources>
```

MainActivity.kt

```
package com.example.practical21

import androidx.appcompat.app.AppCompatActivity
```

```
import android.os.Bundle

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

Practical 3

Programing activities and fragments, activity life cycle, method, multiple activities life cycle of fragments and multiple fragments

activity_main.xml

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

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

    <Button
        android:id="@+id/about_us"
        android:layout_width="130dp"
        android:layout_height="102dp"
        android:text="AboutUs"
        tools:layout_editor_absoluteX="154dp"
        tools:layout_editor_absoluteY="323dp"
        tools:ignore="MissingConstraints" />

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

MainActivity.kt

```
package com.example.myapplication111

import android.content.Intent
import android.os.Bundle
import android.util.Log
import android.widget.Button
import androidx.appcompat.app.AppCompatActivity

class MainActivity : AppCompatActivity() {

    val TAG="Main Activity"
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_main)
        Log.d("ActivityLifecycle", "onCreate called")
        val about_us = findViewById<Button>(R.id.about_us)
        about_us.setOnClickListener{
            val i = Intent(this,AboutUs::class.java)
            startActivity(i)
        }

    }

    override fun onStart() {
        super.onStart()
        Log.d("ActivityLifecycle", "onStart called")
    }

    override fun onResume() {
        super.onResume()
        Log.d("ActivityLifecycle", "onResume called")
    }

    override fun onPause() {
        super.onPause()
        Log.d("ActivityLifecycle", "onPause called")
    }

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

```

        Log.d("ActivityLifecycle", "onStop called")
    }

    override fun onDestroy() {
        super.onDestroy()
        Log.d("ActivityLifecycle", "onDestroy called")
    }
}

```

AboutUs.kt

```

package com.example.myapplication111

import android.os.Bundle
import android.util.Log
import android.widget.Toast
import androidx.appcompat.app.AppCompatActivity

class AboutUs : AppCompatActivity() {
    val TAG = "About Us"

    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        // setContentView(R.layout.activity_main)
        Log.d(TAG, "Inside onCreate")
        Toast.makeText(this, "You are under about us", Toast.LENGTH_LONG).show()
    }
}

```

AndroidManifest.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="@string/app_name"
        android:roundIcon="@mipmap/ic_launcher_round"

```

```

        android:supportsRtl="true"
        android:theme="@style/Theme.MyApplication111"
        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>
        <activity android:name=".AboutUs"></activity>
    </application>

</manifest>

```

Practical 4

Programing related to different layouts: linear, relative, table

Linear Layout

Activity_main.xml

```

<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    tools:context=".MainActivity"
    android:orientation="vertical"
    android:background="@color/material_dynamic_primary70"
    >

    <LinearLayout
        android:layout_height="wrap_content"
        android:layout_width="wrap_content"
        android:orientation="horizontal"
        android:layout_marginTop="20dp">

        <EditText
            android:id="@+id/editTextText2"
            android:layout_width="200dp"
            android:layout_height="wrap_content"
            android:ems="10"

```

```

        android:inputType="text"

        android:hint="@string/search"/>

<Button
    android:id="@+id/button3"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginLeft="10dp"
    android:text="@string/search" />

</LinearLayout>
<EditText
    android:id="@+id/editTextText3"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_weight="1"
    android:ems="10"
    android:inputType="text"
    android:text="Name" />

<ImageView
    android:id="@+id/imageView"
    android:layout_width="wrap_content"
    android:layout_height="164dp"
    android:layout_gravity="bottom"
    app:srcCompat="@drawable/ic_launcher_background" />
</LinearLayout>

```

Relative Layout

Activity_main.xml

```

<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    tools:context=".MainActivity"
    android:orientation="vertical"
    android:background="@color/material_dynamic_primary70"
    >

    <LinearLayout

```



```
android:layout_height="wrap_content"
android:layout_width="wrap_content"
android:orientation="horizontal"
android:layout_marginTop="20dp">
```

```
<EditText
    android:id="@+id/editTextText2"
    android:layout_width="200dp"
    android:layout_height="wrap_content"
    android:ems="10"
    android:inputType="text"
```

```
    android:hint="@string/search"/>
```

```
<Button
    android:id="@+id/button3"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginLeft="10dp"
    android:text="@string/search" />
```

```
</LinearLayout>
```

```
<EditText
    android:id="@+id/editTextText3"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_weight="1"
    android:ems="10"
    android:inputType="text"
    android:text="Name" />
```

```
<ImageView
    android:id="@+id/imageView"
    android:layout_width="wrap_content"
    android:layout_height="164dp"
    android:layout_gravity="bottom"
    app:srcCompat="@drawable/ic_launcher_background" />
```

```
</RelativeLayout>
```

Table Layout(activity_main.xml)

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

```
<TableLayout 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">
```

```
<TableRow
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:layout_marginTop="150dp"
    android:layout_marginLeft="50dp">
```

```
<EditText
    android:id="@+id/mytext"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:ems="10"
    android:hint="username"
    android:inputType="text" />
```

```
</TableRow>
```

```
<TableRow
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:layout_marginTop="50dp"
    android:layout_marginLeft="50dp">
```

```
<EditText
    android:id="@+id/editTextText2"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:ems="10"
    android:inputType="text"
    android:hint="password" />
```

```
</TableRow>
```

```
<TableRow
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:layout_marginTop="50dp"
    android:layout_marginLeft="50dp">
```

```
<Button
    android:text="Login"
```

```

        android:id="@+id/login"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
    />
</TableRow>
</TableLayout>

```

Table Layout(MainActivity.kt)

```

package com.example.myapplicationtable

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

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

        var text=findViewById<EditText>(R.id.mytext)
        var btn=findViewById<Button>(R.id.login)
        btn.setOnClickListener{
            var strMessage:String
            strMessage=text.getText().toString()
            Toast.makeText(this,"hello $strMessage",Toast.LENGTH_LONG).show()
        }
    }
}

```

PRACTICAL NO 5

Programing ui elements

ACTIVITY_MAIN.XML

No changes

MainActivity.kt

```

package com.example.appbar

```

```

import android.os.Bundle
import android.view.Menu
import android.view.MenuItem
import android.widget.Toast
import androidx.activity.enableEdgeToEdge
import androidx.appcompat.app.AppCompatActivity
import androidx.core.view.ViewCompat
import androidx.core.view.WindowInsetsCompat
import com.example.appbar.R.menu.menu

class MainActivity : AppCompatActivity() {
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        enableEdgeToEdge()
        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 {
        val id=item.itemId

        if (id==R.id.search)
        {
            Toast.makeText(this,"search",Toast.LENGTH_LONG).show()
        }
        if (id==R.id.aboutus)
        {
            Toast.makeText(this,"AboutUs",Toast.LENGTH_LONG).show()
        }
        if (id==R.id.contactus)
        {
            Toast.makeText(this,"Contact Us",Toast.LENGTH_LONG).show()
        }
        return super.onOptionsItemSelected(item)
    }
}

```

menu.xml

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

    <item
        android:id="@+id/search"
        android:title="@string/search"
        app:showAsAction="always" />

    <item
        android:id="@+id/aboutus"
        android:title="@string/about_us" />

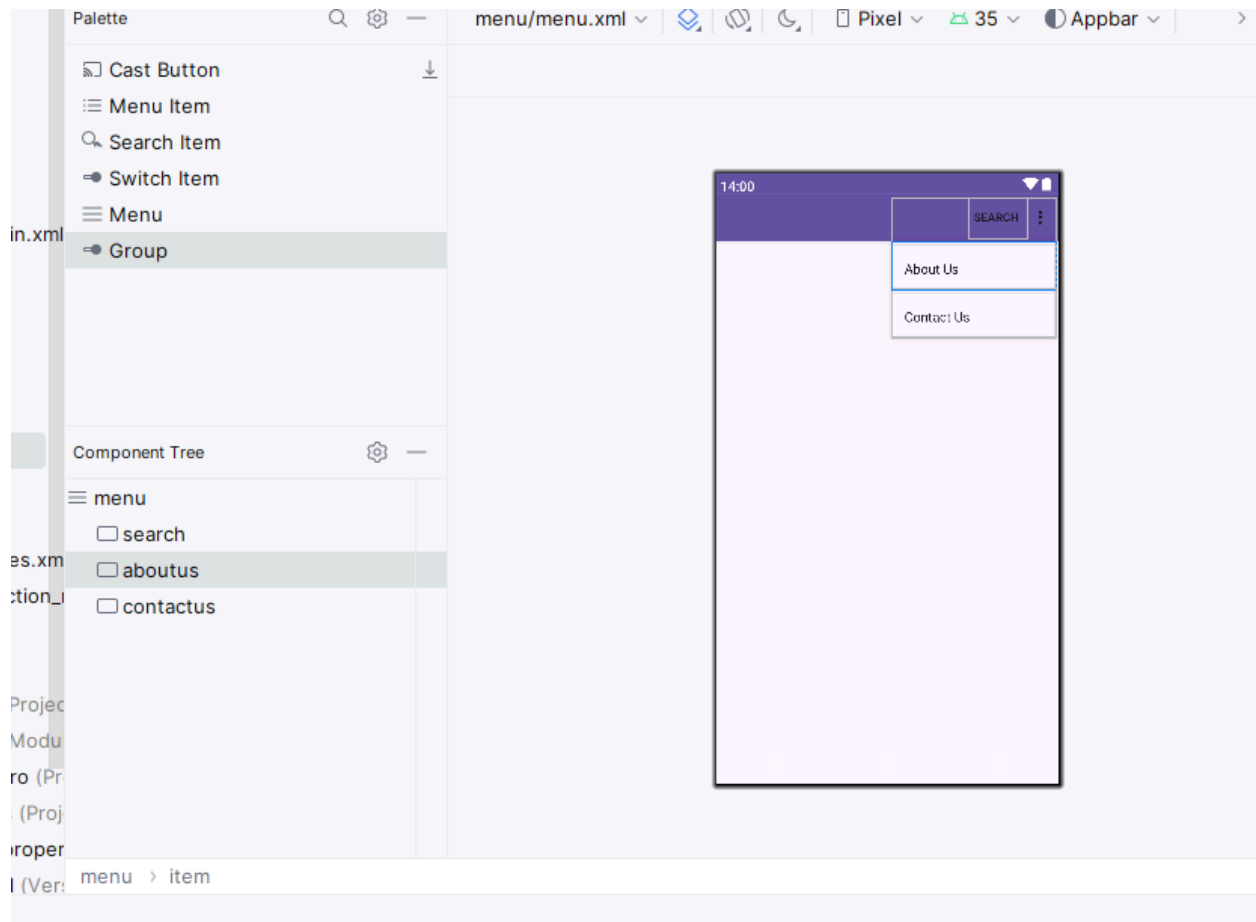
    <item
        android:id="@+id/contactus"
        android:title="@string/contact_us" />

</menu>
```

String.xml

```
<resources>
    <string name="search">Search</string>
    <string name="about_us">About Us</string>
    <string name="contact_us">Contact Us</string>
</resources>
```

Design part



PRACTICAL 6 : FRAGMENT

ACTIVITY_MAIN.XML

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

```
<LinearLayout
    android:layout_width="432dp"
    android:layout_height="match_parent"
    android:orientation="horizontal"
    tools:layout_editor_absoluteX="-19dp"
    tools:layout_editor_absoluteY="54dp">
```

```

<androidx.fragment.app.FragmentContainerView
    android:id="@+id/fragmentContainerView"
    android:name="com.example.fragment.Fragmentone"
    android:layout_width="wrap_content"
    android:layout_height="match_parent"
    android:layout_weight="1" />

<androidx.fragment.app.FragmentContainerView
    android:id="@+id/fragmentContainerView2"
    android:name="com.example.fragment.Fragmenttwo"
    android:layout_width="wrap_content"
    android:layout_height="match_parent"
    android:layout_weight="1" />
</LinearLayout>
</androidx.constraintlayout.widget.ConstraintLayout>

```

MAINACTIVITY.KT

```

package com.example.fragment

import android.os.Bundle
import androidx.activity.enableEdgeToEdge
import androidx.appcompat.app.AppCompatActivity
import androidx.core.view.ViewCompat
import androidx.core.view.WindowInsetsCompat

class MainActivity : AppCompatActivity() {
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        enableEdgeToEdge()
        setContentView(R.layout.activity_main)
        ViewCompat.setOnApplyWindowInsetsListener(findViewById(R.id.main)) { v, insets
->
            val systemBars = insets.getInsets(WindowInsetsCompat.Type.systemBars())
            v.setPadding(systemBars.left, systemBars.top, systemBars.right,
systemBars.bottom)
            insets
        }
    }
}

```

FRAGMENTONE.KT

```
package com.example.fragment

import android.os.Bundle
import androidx.fragment.app.Fragment
import android.view.LayoutInflater
import android.view.View
import android.view.ViewGroup

/**
 * A simple [Fragment] subclass.
 * Use the [Fragmentone.newInstance] factory method to
 * create an instance of this fragment.
 */
class Fragmentone : Fragment() {
    override fun onCreateView(
        inflater: LayoutInflater, container: ViewGroup?,
        savedInstanceState: Bundle?
    ): View? {
        // Inflate the layout for this fragment
        return inflater.inflate(R.layout.fragment_fragmentone, container, false)
    }
}
```

FRAGMENTTWO.KT

```
package com.example.fragment

import android.os.Bundle
import androidx.fragment.app.Fragment
import android.view.LayoutInflater
import android.view.View
import android.view.ViewGroup

/**
 * A simple [Fragment] subclass.
```


* Use the [Fragmenttwo.newInstance] factory method to
* create an instance of this fragment.
*/

```
class Fragmenttwo : Fragment() {  
    override fun onCreateView(  
        inflater: LayoutInflater, container: ViewGroup?,  
        savedInstanceState: Bundle?  
    ): View? {  
        // Inflate the layout for this fragment  
        return inflater.inflate(R.layout.fragment_two, container, false)  
    }  
  
}
```

FRAGMENTONE.XML

```
<?xml version="1.0" encoding="utf-8"?>  
<FrameLayout xmlns:android="http://schemas.android.com/apk/res/android"  
    xmlns:tools="http://schemas.android.com/tools"  
    android:layout_width="match_parent"  
    android:layout_height="match_parent"  
    android:background="#FF5722"  
    tools:context=".Fragmentone">  
  
    <!-- TODO: Update blank fragment layout -->  
    <TextView  
        android:layout_width="wrap_content"  
        android:layout_height="wrap_content"  
        android:layout_gravity="center"  
        android:text="fragment 1"  
        android:textSize="34sp" />  
  
</FrameLayout>
```

FRAGMENTTWO.XML

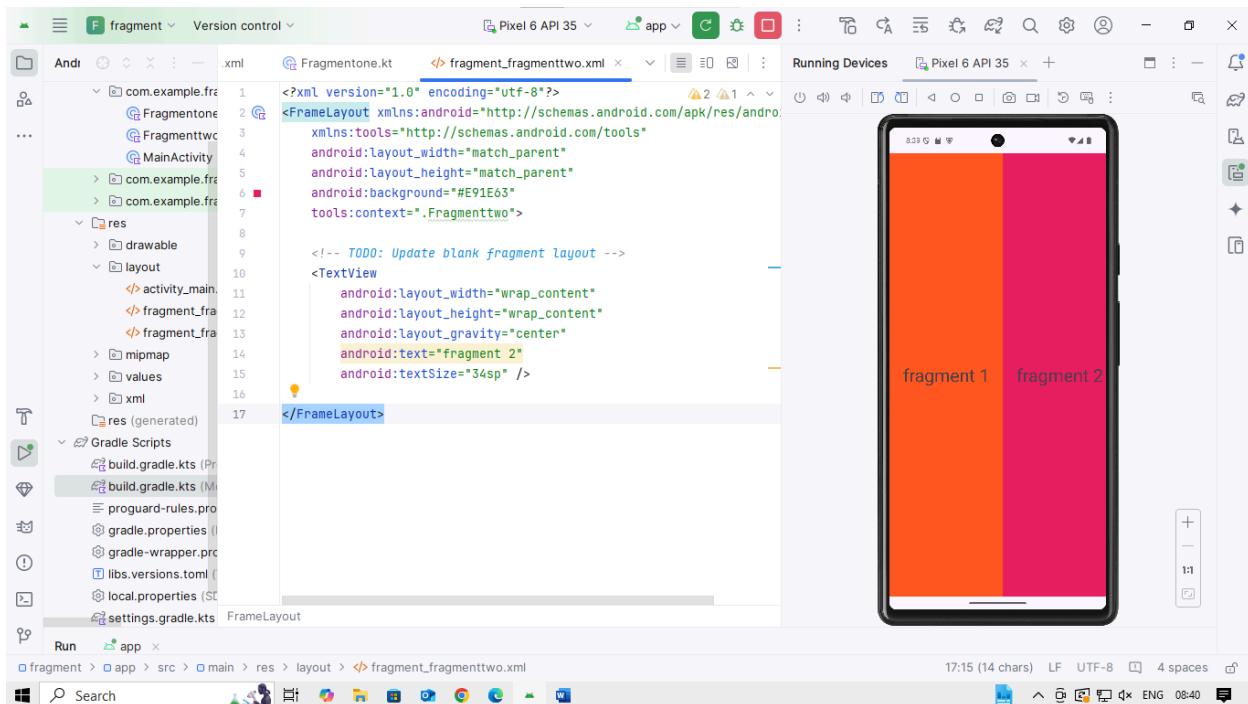
```
<?xml version="1.0" encoding="utf-8"?>  
<FrameLayout xmlns:android="http://schemas.android.com/apk/res/android"  
    xmlns:tools="http://schemas.android.com/tools"  
    android:layout_width="match_parent"
```

```
android:layout_height="match_parent"
android:background="#E91E63"
tools:context=".Fragmenttwo">
```

```
<!-- TODO: Update blank fragment layout -->
<TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_gravity="center"
    android:text="fragment 2"
    android:textSize="34sp" />
```

```
</FrameLayout>
```

OUTPUT



Practical 7 :- Programs on Intents, Events, Listeners and Adapters

Activity_main.xml

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

```
xmlns:app="http://schemas.android.com/apk/res-auto"
android:layout_width="match_parent"
android:layout_height="match_parent"
tools:context=".MainActivity"
android:background="@color/black">
<TableLayout android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginLeft="50dp"
    android:layout_marginTop="150dp">
<TableRow>
    <Button
        android:id="@+id/btn1"
        android:text="1"
        android:layout_gravity="center"
    />
    <Button
        android:id="@+id/btn2"
        android:text="2"
        android:layout_gravity="center"

    />
    <Button
        android:id="@+id/btn3"
        android:text="3"
        android:layout_gravity="center"
    />
</TableRow>
<TableRow>
    <Button
        android:id="@+id/btn4"
        android:text="4"

        android:layout_gravity="center"
    />
    <Button
        android:id="@+id/btn5"
        android:text="5"
        android:layout_gravity="center"

    />
    <Button
        android:id="@+id/btn6"
        android:text="6"
        android:layout_gravity="center"
```

```

        />
    </TableRow>
    <TableRow>
        <Button
            android:id="@+id/btn7"
            android:text="7"

            android:layout_gravity="center"

        />
        <Button
            android:id="@+id/btn8"
            android:text="8"
            android:layout_gravity="center"

        />
        <Button
            android:id="@+id/btn9"
            android:text="9"
            android:layout_gravity="center"
        />
    </TableRow>
</TableLayout>
</LinearLayout>

```

MainActivity.xml

```

package com.example.intent

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

class MainActivity : AppCompatActivity() {
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        enableEdgeToEdge() // Make sure this is available for
your Android version
        setContentView(R.layout.activity_main)

        // Initialize buttons
        val btn1 = findViewById<Button>(R.id.btn1)
        val btn2 = findViewById<Button>(R.id.btn2)
    }
}

```

```

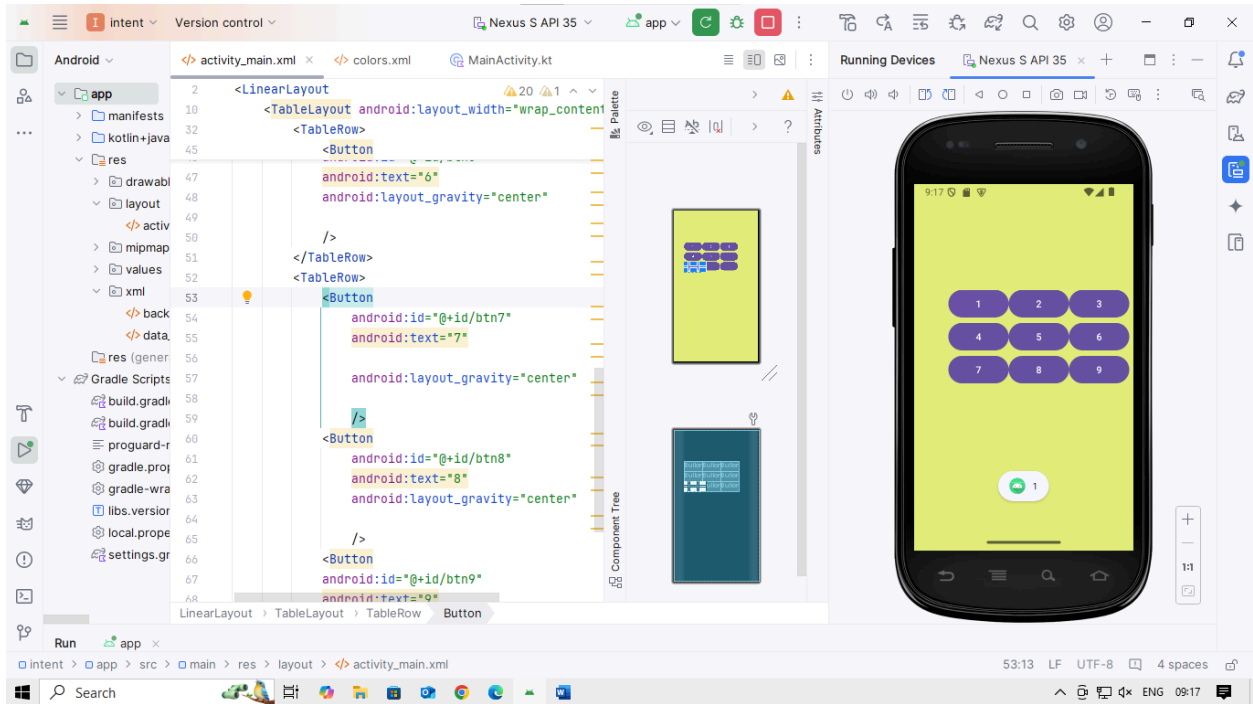
val btn3 = findViewById<Button>(R.id.btn3)
val btn4 = findViewById<Button>(R.id.btn4)
val btn5 = findViewById<Button>(R.id.btn5)
val btn6 = findViewById<Button>(R.id.btn6)
val btn7 = findViewById<Button>(R.id.btn7)
val btn8 = findViewById<Button>(R.id.btn8)
val btn9 = findViewById<Button>(R.id.btn9)

// Set click listeners
btn1.setOnClickListener { toast("1") }
btn2.setOnClickListener { toast("2") }
btn3.setOnClickListener { toast("3") }
btn4.setOnClickListener { toast("4") }
btn5.setOnClickListener { toast("5") }
btn6.setOnClickListener { toast("6") }
btn7.setOnClickListener { toast("7") }
btn8.setOnClickListener { toast("8") }
btn9.setOnClickListener { toast("9") }
}

private fun toast(s: String) {
    Toast.makeText(this, s, Toast.LENGTH_SHORT).show()
}
}

```

Output



Practical 8

Notification:

A) clickable and non- clickable notification

Activity_main.xml

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

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

```
    xmlns:tools="http://schemas.android.com/tools"
```

```
    android:layout_width="match_parent"
```

```
    android:layout_height="match_parent"
```

```
    android:orientation="vertical"
```

```
    android:padding="20dp"
```

```
    android:gravity="center_vertical"
```

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

```
    <Button
```

```
        android:id="@+id/btnGetNonClickableNotification"
        android:layout_width="match_parent"
        android:text="Get Non-Clickable Notification"
        android:textSize="20sp"
        android:padding="15dp"
        android:layout_marginBottom="10dp"
        android:layout_height="wrap_content"/>
<Button
    android:id="@+id/btnGetClickableNotification"
    android:layout_width="match_parent"
    android:text="Get Clickable Notification"
    android:textSize="20sp"
    android:padding="15dp"
    android:layout_height="wrap_content"/>
</LinearLayout>
```

MainActivity.kt

```
package com.example.notification

import android.app.Notification
import android.app.NotificationChannel
import android.app.NotificationManager
import android.app.PendingIntent
import android.content.Intent
import android.os.Bundle
import android.widget.Button
```

```
import androidx.activity.enableEdgeToEdge
import androidx.appcompat.app.AppCompatActivity
import androidx.core.view.ViewCompat
import androidx.core.view.WindowInsetsCompat

class MainActivity : AppCompatActivity() {

    lateinit var getClickableNotificationButton: Button
    lateinit var getNonClickableNotificationButton: Button
    private val CHANNEL_ID = "my_channel"
    private val NOTIFICATION_ID_NON_CLICKABLE = 1
    private val NOTIFICATION_ID_CLICKABLE = 2
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)

        setContentView(R.layout.activity_main)
        createNotificationChannel()
        getClickableNotificationButton =
            findViewById(R.id.btnGetClickableNotification)
        getNonClickableNotificationButton =
            findViewById(R.id.btnGetNonClickableNotification)
        getClickableNotificationButton.setOnClickListener {
            showClickableNotification()
        }
        getNonClickableNotificationButton.setOnClickListener {
            showNonClickableNotification()
        }
    }
}
```



```

}

private fun createNotificationChannel() {
    val name = "Practical"
    val descriptionText = "Home Page"
    val importance = NotificationManager.IMPORTANCE_DEFAULT
    val channel = NotificationChannel(CHANNEL_ID, name, importance).apply {
        description = descriptionText
    }
    val notificationManager: NotificationManager =
        getSystemService(NOTIFICATION_SERVICE) as NotificationManager
    notificationManager.createNotificationChannel(channel)
}

```

```

private fun showClickableNotification() {
    val intent = Intent(this, MainActivity::class.java)
    val pendingIntent = PendingIntent.getActivity(this, 0, intent,
        PendingIntent.FLAG_IMMUTABLE)
    val builder = Notification.Builder(this, CHANNEL_ID)
        .setSmallIcon(R.drawable.ic_launcher_background)
        .setContentTitle("WELCOME TO SN COLLEGE")
        .setContentText("Click to reload the app")
        .setContentIntent(pendingIntent)
        .setAutoCancel(true)
    val notificationManager: NotificationManager =
        getSystemService(NOTIFICATION_SERVICE) as NotificationManager
    notificationManager.notify(NOTIFICATION_ID_CLICKABLE, builder.build())
}

```

```

}

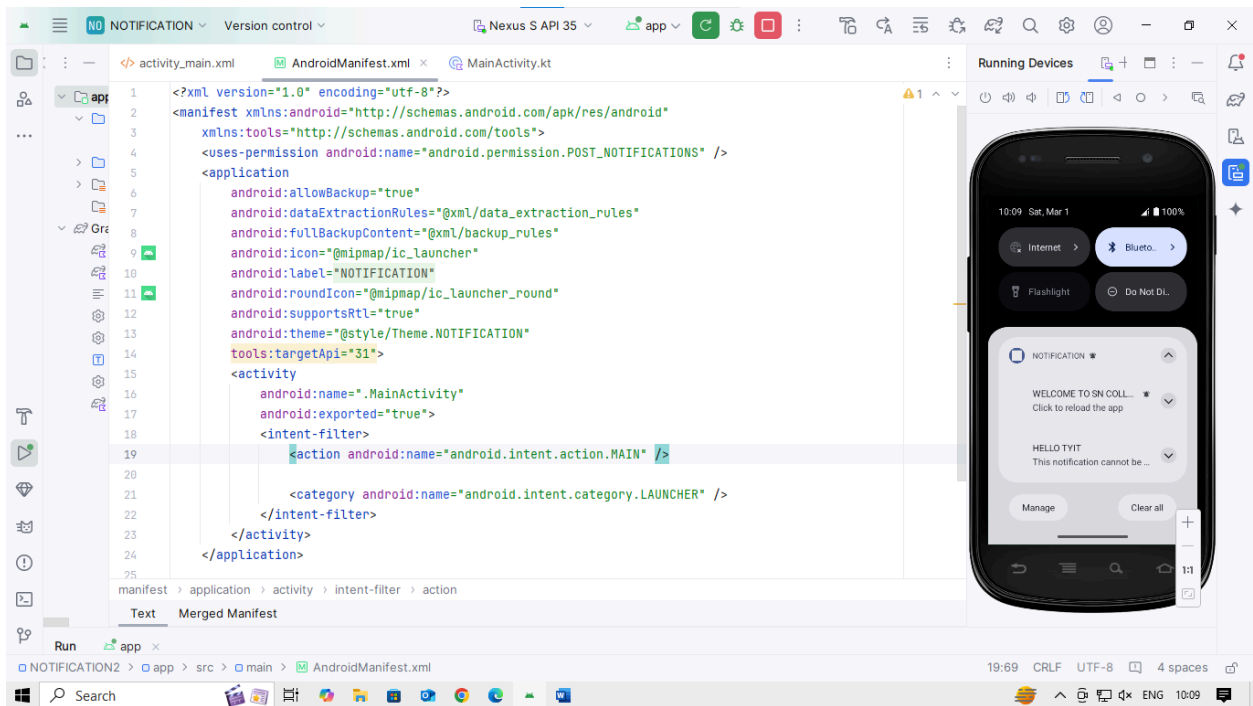
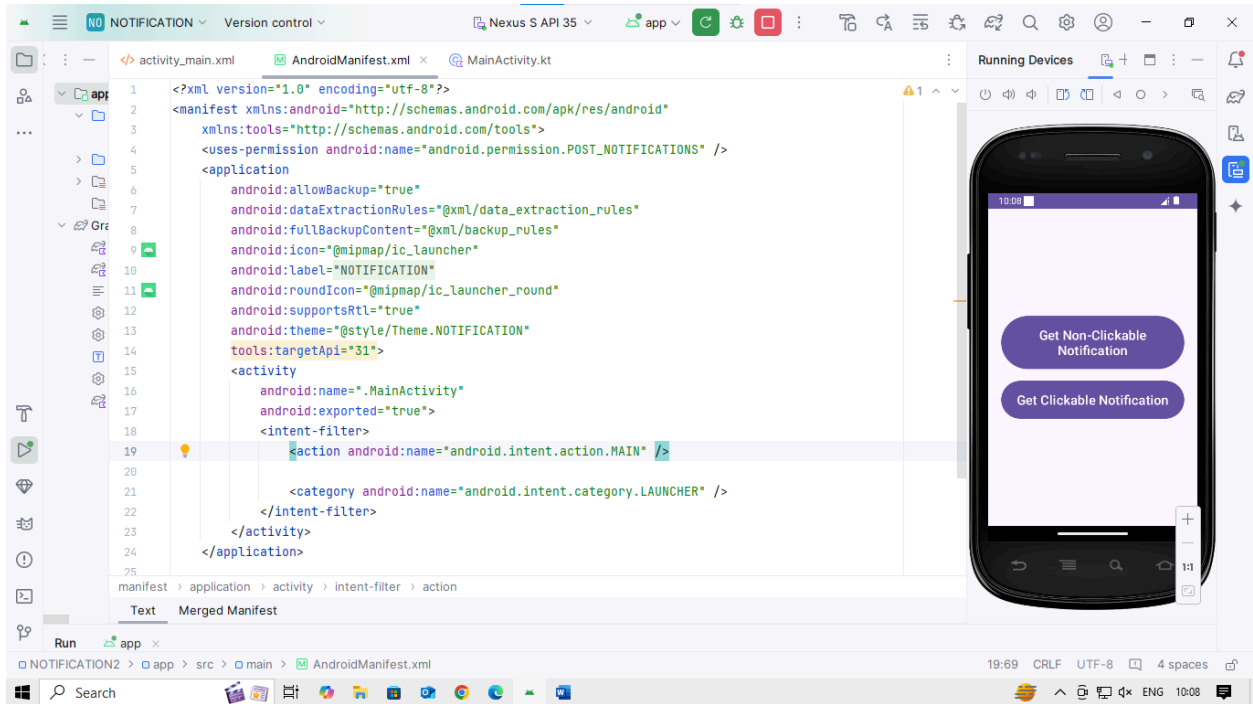
private fun showNonClickableNotification() {
    val builder = Notification.Builder(this, CHANNEL_ID)
        .setSmallIcon(R.drawable.ic_launcher_background)
        .setContentTitle("HELLO TYIT")
        .setContentText("This notification cannot be clicked.")
        .setAutoCancel(true)
    val notificationManager: NotificationManager =
        getSystemService(NOTIFICATION_SERVICE) as NotificationManager
    notificationManager.notify(NOTIFICATION_ID_NON_CLICKABLE, builder.build())
}
}

```

Androidmanifest.xml

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

Output



Practical 8 :

Notification

B)automatic broadcast receiver

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:id="@+id/main"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:gravity="center"
    android:padding="20dp"
    tools:context=".MainActivity">
    <TextView
        android:id="@+id/infoText"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:textAlignment="center"
        android:textStyle="bold"
        android:textSize="30sp"
        android:text="Hello World!" />
</LinearLayout>
```

AndroidManifest.xml

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

Mainactivity.kt

```
package com.example.broadcast
```

```
import android.content.Context
import android.net.ConnectivityManager
import android.net.NetworkCapabilities
import android.os.Bundle
import android.widget.TextView
import androidx.appcompat.app.AppCompatActivity

class MainActivity : AppCompatActivity() {
    lateinit var infoText: TextView

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

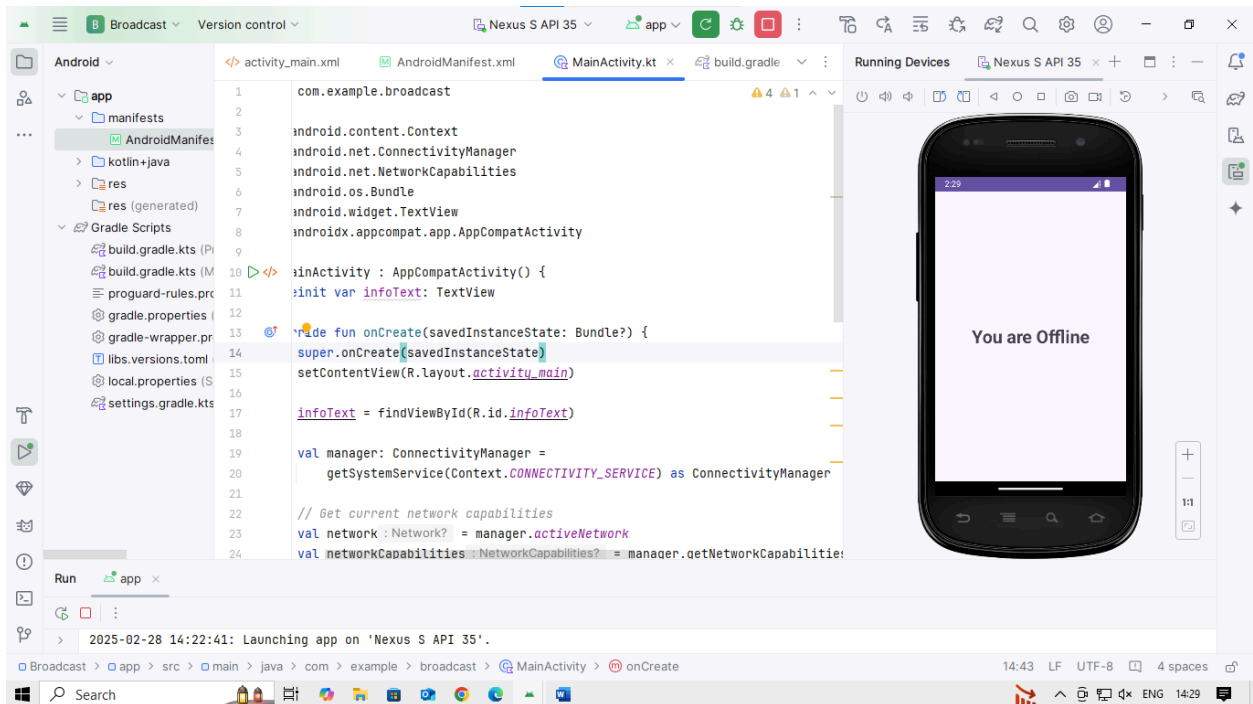
        infoText = findViewById(R.id.infoText)

        val manager: ConnectivityManager =
            getSystemService(Context.CONNECTIVITY_SERVICE) as ConnectivityManager

        val network = manager.activeNetwork
        val networkCapabilities = manager.getNetworkCapabilities(network)
```

```
    if (networkCapabilities != null &&
networkCapabilities.hasCapability(NetworkCapabilities.NET_CAPABILITY_INTERNET)) {
        when {
            networkCapabilities.hasTransport(NetworkCapabilities.TRANSPORT_CELLULAR) -> {
                infoText.text = "Connected to Cellular"
            }
            networkCapabilities.hasTransport(NetworkCapabilities.TRANSPORT_WIFI) -> {
                infoText.text = "Connected to WiFi"
            }
            else -> {
                infoText.text = "Connected to other network"
            }
        }
    } else {
        infoText.text = "You are Offline"
    }
}
}
```

output:



Practical 10: Programming Security and permissions

AndroidManifest.xml

```
<uses-feature
    android:name="android.hardware.camera"
    android:required="false" />
<uses-permission android:name="android.permission.INTERNET" />
<uses-permission android:name="android.permission.CAMERA"/>
<uses-permission
    android:name="android.permission.READ_CONTACTS"/>
<uses-permission
    android:name="android.permission.READ_CALENDAR"/>
```

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:id="@+id/main"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    android:padding="30dp">
```

```

        android:gravity="center"
        tools:context=".MainActivity">
        <Button
            android:id="@+id/cameraButton"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:text="Give Camera Permission" />
        <Button
            android:id="@+id/contactsButton"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:text="Give Contacts Permission" />
        <Button
            android:id="@+id/calendarButton"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:text="Give Calendar Permission" />
    </LinearLayout>

```

MainActivity.kt

```

package com.example.permission

import android.Manifest
import android.content.pm.PackageManager
import android.os.Bundle
import android.widget.Button
import android.widget.Toast
import androidx.activity.enableEdgeToEdge
import androidx.appcompat.app.AppCompatActivity
import androidx.core.view.ViewCompat
import androidx.core.view.WindowInsetsCompat

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

    lateinit var calendarBtn: Button
    lateinit var contactsBtn: Button
    private val MY_CAMERA_REQUEST_CODE = 100;
    private val MY_CALENDAR_REQUEST_CODE = 101;
    private val MY_CONTACTS_REQUEST_CODE = 102;
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_main)
        cameraBtn = findViewById(R.id.cameraButton)

```



```

        calendarBtn = findViewById(R.id.calendarButton)
        contactsBtn = findViewById(R.id.contactsButton)
        cameraBtn.setOnClickListener {
            checkAndRequestPermission("Camera",
Manifest.permission.CAMERA,MY_CAMERA_REQUEST_CODE)
        }
        calendarBtn.setOnClickListener {
            checkAndRequestPermission("Calendar",
Manifest.permission.READ_CALENDAR,MY_CALENDAR_REQUEST_CODE)
        }
        contactsBtn.setOnClickListener {
            checkAndRequestPermission("Contacts",
Manifest.permission.READ_CONTACTS,
MY_CONTACTS_REQUEST_CODE)
        }
    }
    private fun checkAndRequestPermission(whichButton: String,
whichPermission:
String, requestCode: Int) {
        if (checkSelfPermission(whichPermission) !=
PackageManager.PERMISSION_GRANTED) {
            Toast.makeText(this, "$whichButton Button Clicked",
                Toast.LENGTH_SHORT).show()
            requestPermissions(arrayOf(whichPermission),
requestCode);
        } else {
            Toast.makeText(this, "Already has $whichButton
Access",
                Toast.LENGTH_SHORT).show()
        }
    }
}

```

Output

