Practical -1

Aim : Create a List View with 4 items on the click of the item it should display item in a Toast.

Code:

activity_main.xml

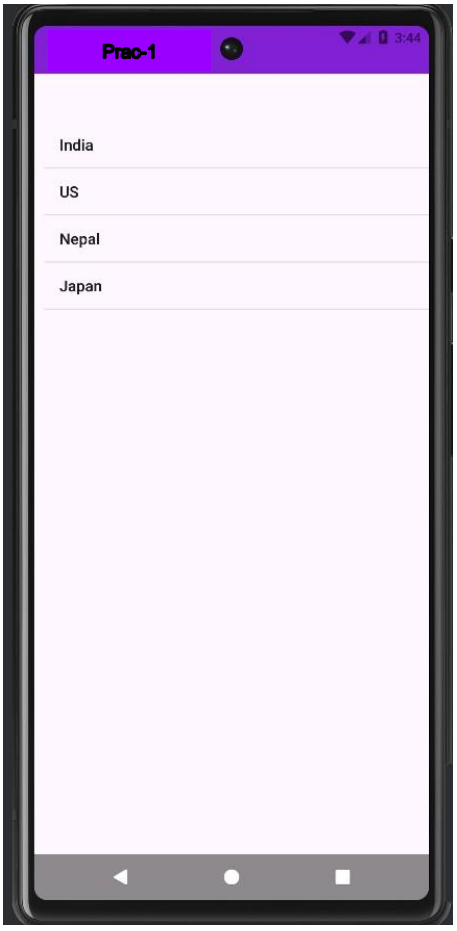
```
<?xml version="1.0" encoding="utf-8"?>
<androidx.appcompat.widget.LinearLayoutCompat
xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:id="@+id/main"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".MainActivity">
<Toolbar
    android:layout_width="520dp"
    android:title="Prac-1"
    android:layout_marginRight="-510dp"
    android:background="@color/purple"
    android:layout_height="50dp">
</Toolbar>
    <ListView
        android:id="@+id/listview"
        android:layout_width="500dp"
        android:layout_marginTop="100dp"
        android:layout_height="500dp">
    </ListView>

</androidx.appcompat.widget.LinearLayoutCompat>
```

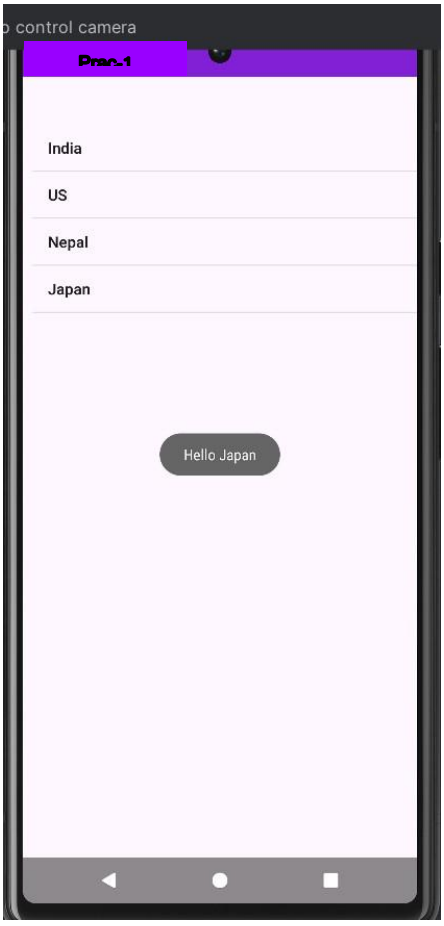
MainActivity.kt

```
package com.example.practical_1a

import android.os.Bundle
import android.view.Gravity
import android.widget.ArrayAdapter
import android.widget.ListView
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 mylist:ListView
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
```

```
        enableEdgeToEdge()
        setContentView(R.layout.activity_main)
        mylist = findViewById(R.id.listview)
        var country=arrayOf<String>("India","US","Nepal","Japan")
        mylist.adapter=
ArrayAdapter<String>(this,android.R.layout.simple_list_item_1,country)
        mylist.setOnItemClickListener { adapterView, view, i, l ->
            val toast = Toast.makeText(this, "Hello " + country[i],
Toast.LENGTH_LONG,)
            toast.setGravity(Gravity.CENTER, 0, 0)
            toast.show()
        }
    }
}
```

Output :





Practical -2

Aim : Create an android application to pass the data from one activity to another activity in the same application using intent.

Code:

activity_main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.appcompat.widget.LinearLayoutCompat
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"
   tools:context=".MainActivity">
   <Toolbar
       android:layout_width="520dp"
       android:title="Prac-2"
       android:layout_marginRight="-510dp"
       android:background="@color/purple"
       android:layout_height="50dp">
   </Toolbar>

   <EditText
       android:layout_width="match_parent"
       android:layout_marginTop="250dp"
       android:id="@+id/edit"
       android:hint="Enter you Name"
       android:layout_height="wrap_content"
       >
   </EditText>
   <Button
       android:layout_width="match_parent"
       android:id="@+id/btn"
       android:text="Submit"
       android:layout_height="wrap_content"
       >
   </Button>


</androidx.appcompat.widget.LinearLayoutCompat>
```

MainActivity.kt

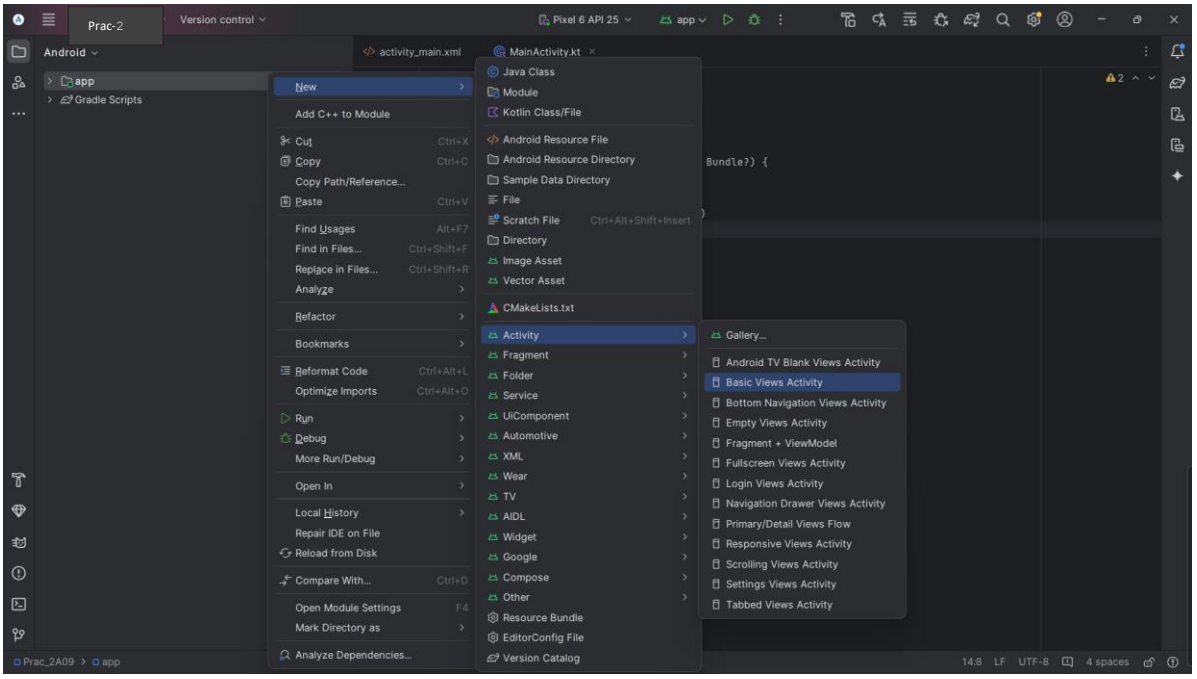
```
package com.example.prac_2a_09

import android.content.Intent
import android.os.Bundle
```

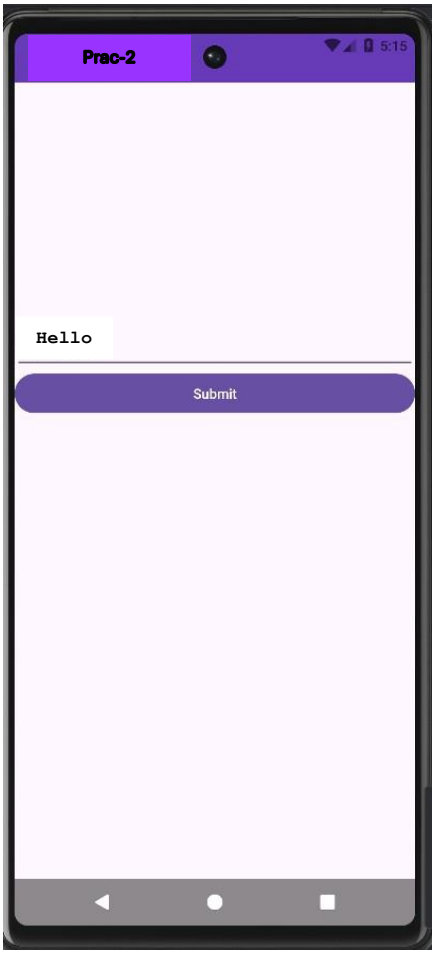
```
import android.widget.Button
import android.widget.EditText
import androidx.activity.enableEdgeToEdge
import androidx.appcompat.app.AppCompatActivity
import androidx.core.view.ViewCompat
import androidx.core.view.WindowInsetsCompat

class MainActivity : AppCompatActivity() {
   lateinit var editText:EditText
   lateinit var button: Button
   override fun onCreate(savedInstanceState: Bundle?) {
       super.onCreate(savedInstanceState)
       enableEdgeToEdge()
       setContentView(R.layout.activity_main)
       editText=findViewById(R.id.edit)
       button=findViewById(R.id.btn)
       button.setOnClickListener{
           var intent= Intent(this,home::class.java)
           intent.putExtra("msg",editText.text.toString())
           startActivity(intent)
       }

   }
}
```

Home.kt

```
package com.example. practical_2a

import android.os.Bundle
import android.widget.Toast
import com.google.android.material.snackbar.Snackbar
import androidx.appcompat.app.AppCompatActivity
import androidx.navigation.findNavController
import androidx.navigation.ui.AppBarConfiguration
import androidx.navigation.ui.navigateUp
import androidx.navigation.ui.setupActionBarWithNavController
import com.example.practical_2a.databinding.ActivityHomeBinding

class home : AppCompatActivity() {
   override fun onCreate(savedInstanceState: Bundle?) {
       super.onCreate(savedInstanceState)
       var data:Bundle?=intent.extras

       var msg=data!!.get("msg")
       Toast.makeText(this, ""+msg, Toast.LENGTH_SHORT).show()

   }
}
```

Output :


Prac-2
Version control
Pixel 6 API 25
app
Android
activity_main.xml
MainActivity.kt
app
Gradle Scripts
New
Add C++ to Module
Cut
Copy
Copy Path/Reference...
Paste
Find Usages
Find in Files...
Replace in Files...
Analyze
Refactor
Bookmarks
Reformat Code
Optimize Imports
Run
Debug
More Run/Debug
Open In
Local History
Repair IDE on File
Reload from Disk
Compare With...
Open Module Settings
Mark Directory as
Analyze Dependencies...
Java Class
Module
Kotlin Class/File
Android Resource File
Android Resource Directory
Sample Data Directory
File
Scratch File
Directory
Image Asset
Vector Asset
CMakeLists.txt
Activity
Fragment
Folder
Service
UiComponent
Automotive
XML
Wear
TV
AIDL
Widget
Google
Compose
Other
Resource Bundle
EditorConfig File
Version Catalog
Gallery...
Android TV Blank Views Activity
Basic Views Activity
Bottom Navigation Views Activity
Empty Views Activity
Fragment + ViewModel
Fullscreen Views Activity
Login Views Activity
Navigation Drawer Views Activity
Primary/Detail Views Flow
Responsive Views Activity
Scrolling Views Activity
Settings Views Activity
Tabbed Views Activity
Prac_2A09
app


Prac-2
5:15
Hello
Submit


5:18
Hello

Practical -3

Aim : Create an android application to check whether the App is connected to mobile device or Wi-Fi.

Code:

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">
   <uses-permission  android:name="android.permission.ACCESS_NETWORK_STATE" />
   <uses-permission android:name="android.permission.INTERNET"/>
   <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.Prac_3"
       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>
```

MainActivity.kt

```
package com.example.prac_3a_

import android.content.Context
import android.net.ConnectivityManager
import android.net.NetworkInfo
import android.os.Bundle
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() {
   override fun onCreate(savedInstanceState: Bundle?) {
       super.onCreate(savedInstanceState)
       enableEdgeToEdge()
       setContentView(R.layout.activity_main)
       var manager: ConnectivityManager
=getSystemService(Context.CONNECTIVITY_SERVICE) as ConnectivityManager
       var netwrokinfo: NetworkInfo?=manager.activeNetworkInfo
       if(netwrokinfo!=null && netwrokinfo.isConnectedOrConnecting)
       {
           if(netwrokinfo.type==ConnectivityManager.TYPE_MOBILE)
           {
               Toast.makeText(this,"You are Connected to Mobile
Network",Toast.LENGTH_LONG).show()
           }
           if(netwrokinfo.type==ConnectivityManager.TYPE_WIFI)
           {
               Toast.makeText(this,"You are Connected to
Wifi",Toast.LENGTH_LONG).show()
           }
       }
       else
       {
           Toast.makeText(this,"Offline",Toast.LENGTH_LONG).show()
       }
   }

}
```

Output:

Practical -4

Aim : Create the Telephone API in android to make call.

Code:

activity_main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.appcompat.widget.LinearLayoutCompat
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">
    <Toolbar
        android:layout_width="520dp"
        android:title="Prac-4"
        android:layout_marginRight="-520dp"
        android:background="@color/purple"
        android:layout_height="50dp">
    </Toolbar>

    <LinearLayout
        android:layout_width="match_parent"
        android:orientation="vertical"
        android:padding="55dp"
        android:layout_height="wrap_content">
        <EditText
            android:layout_width="match_parent"
            android:id="@+id/input"
            android:hint="Enter your number"
            android:layout_height="wrap_content">
        </EditText>
        <Button
            android:layout_width="match_parent"
            android:id="@+id/btn"
            android:text="Call"
            android:layout_height="wrap_content">
        </Button>
    </LinearLayout>

</androidx.appcompat.widget.LinearLayoutCompat>
```

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">
    <uses-permission android:name="android.permission.CALL_PHONE"
        tools:ignore="PermissionImpliesUnsupportedChromeOsHardware">
    </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:supportsRtl="true"
        android:theme="@style/Theme.Prac_4a"
        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>
```

MainActivity.kt

```
package com.example.prac_4a
import android.Manifest
import android.content.Intent
import android.content.pm.PackageManager
import android.net.Uri
import android.os.Bundle
import android.widget.Button
import android.widget.EditText
import androidx.appcompat.app.AppCompatActivity
import androidx.core.app.ActivityCompat

class MainActivity : AppCompatActivity() {

    val REQUEST_CODE=1
    lateinit var btn:Button
    lateinit var edt:EditText
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_main)
        btn=findViewById(R.id.btn)
        edt=findViewById(R.id.input)
        btn.setOnClickListener {
if(ActivityCompat.checkSelfPermission(this,Manifest.permission.CALL_PHONE)!=
PackageManager.PERMISSION_GRANTED){
                ActivityCompat.requestPermissions(this,
arrayOf(Manifest.permission.CALL_PHONE,Manifest.permission.ACCESS_FINE_LOCAT
ION),REQUEST_CODE)
            }else{
                makecall();
            }
```

```
        }
    }
    private fun makecall() {
        val callint=Intent(Intent.ACTION_CALL)
        callint.data= Uri.parse("tel:"+edt.text.toString().trim())
        startActivity(callint)
    }


    override fun onRequestPermissionsResult(
        requestCode: Int,
        permissions: Array<out String>,
        grantResults: IntArray
    ) {
        super.onRequestPermissionsResult(requestCode, permissions,
grantResults)
        if(requestCode==REQUEST_CODE)
            makecall()
    }
}
```

Output :

Practical -5

Aim : Create an android application to give audio recording permission from the android application.

Code:

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">
   <uses-permission android:name="android.permission.RECORD_AUDIO">

   </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:supportsRtl="true"
       android:theme="@style/Theme.Prac_5"
       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>
```

MainActivity.kt

```
package com.example.prac_5a
import android.content.pm.PackageManager
import android.os.Bundle
import android.widget.Toast
import androidx.activity.enableEdgeToEdge
import androidx.appcompat.app.AlertDialog
import androidx.appcompat.app.AppCompatActivity
import androidx.core.app.ActivityCompat
import androidx.core.content.ContextCompat
import androidx.core.view.ViewCompat
import androidx.core.view.WindowInsetsCompat
class MainActivity : AppCompatActivity() {
   val RECORD_AUDIO=1
   override fun onCreate(savedInstanceState: Bundle?) {
       super.onCreate(savedInstanceState)
       setContentView(R.layout.activity_main)
```

```
        setPermision();
    }
    private fun setPermision() {
        val
permission=ContextCompat.checkSelfPermission(this,android.Manifest.permission
.RECORD_AUDIO)
        if(permission!=PackageManager.PERMISSION_GRANTED){
            Toast.makeText(this, "Permision
denied",Toast.LENGTH_SHORT).show()
        }
if(ActivityCompat.shouldShowRequestPermissionRationale(this,android.Manifest.
permission.RECORD_AUDIO )){
            val builder=AlertDialog.Builder(this)
            builder.setMessage("Permision required to record audio")
            builder.setTitle("Permision required");
            builder.setPositiveButton("OK"){
                    dialog,which->
                Toast.makeText(this, "Permision
given",Toast.LENGTH_SHORT).show()
                giverequest()
            }
            builder.create()
            builder.show()
        }else{
            giverequest()
        }
    }
    private fun giverequest() {
        ActivityCompat.requestPermissions(this,
arrayOf(android.Manifest.permission.RECORD_AUDIO),RECORD_AUDIO)
    }
    override fun onRequestPermissionsResult(
        requestCode: Int,
        permissions: Array<out String>,
        grantResults: IntArray
    ) {
        super.onRequestPermissionsResult(requestCode,  permissions,
grantResults)
        when(requestCode){
            RECORD_AUDIO->{

if(grantResults.isEmpty()||grantResults[0]!=PackageManager.PERMISSION_GRANTE
D){
                    Toast.makeText(this, "Permision
denied",Toast.LENGTH_SHORT).show()
                }else{
                    Toast.makeText(this, "Permision
granted",Toast.LENGTH_SHORT).show()
                }
            }
        }
    }
}
```

Output :


12:15
Vighnesh(A-09)
Allow Prac_5 to record audio?
DENY
ALLOW
1:1


12:20
Prac-5
Permision granted
1:1

Practical -6

Aim :Create an android application to demonstrate the use of sub menu the toast should be appeared by selecting the sub menu item.

Code:

MainActivity.kt

```
package com.example.prac_6

import android.os.Bundle
import android.view.Menu
import android.view.MenuItem
import android.widget.Toast
import androidx.appcompat.app.AppCompatActivity
import androidx.appcompat.widget.Toolbar
class MainActivity : AppCompatActivity() {
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_main)

        // Find the Toolbar and set it as the ActionBar
        val toolbar: Toolbar = findViewById(R.id.my_toolbar)
        setSupportActionBar(toolbar)
    }
    override fun onCreateOptionsMenu(menu: Menu?): Boolean {
        menuInflater.inflate(R.menu.menu, menu) // Inflate menu.xml
        return true // Ensure menu is displayed
    }
    override fun onOptionsItemSelected(item: MenuItem): Boolean {
        when (item.itemId) {
            R.id.about -> Toast.makeText(this, "About Selected",
Toast.LENGTH_SHORT).show()
            R.id.setting -> Toast.makeText(this, "Settings Selected",
Toast.LENGTH_SHORT).show()
            R.id.feedback -> Toast.makeText(this, "Feedback Selected",
Toast.LENGTH_SHORT).show()
            else -> return super.onOptionsItemSelected(item)
        }
        return true
    }
}
```

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/setting"
        android:title="Setting"
        app:showAsAction="never">
        <menu>
```

```
            <item android:id="@+id/about"
android:icon="@drawable/baseline_home_24" android:title="About"/>
            <item android:id="@+id/help" android:title="Help"/>
            <item android:id="@+id/feedback" android:title="Feedback"/>
        </menu>
    </item>
</menu>
```

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"
    tools:context=".MainActivity">


    <androidx.appcompat.widget.Toolbar
        android:id="@+id/my_toolbar"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:background="@color/purple"
        app:title="Prac-6"
        app:titleTextColor="@android:color/white" />

</LinearLayout>
```

Output :

Practical -7(a)

Aim : Create an android application to generate notifications, while user will click on notification it will redirect on the MainActivity.

Code:

activity_main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.appcompat.widget.LinearLayoutCompat
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">
   <Toolbar
       android:layout_width="520dp"
       android:title="Prac-7"
       android:layout_marginRight="-520dp"
       android:background="@color/purple"
       android:layout_height="50dp">
   </Toolbar>
   <Button
       android:layout_width="match_parent"
       android:id="@+id/btn"
       android:text="Notify"
       android:layout_marginTop="80dp"
       android:layout_height="wrap_content"
       tools:ignore="MissingConstraints">
   </Button>

</androidx.appcompat.widget.LinearLayoutCompat>
```

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">
   <uses-permission android:name="android.permission.POST_NOTIFICATIONS" />
   <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.Prac_7(1)"
       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>
```

MainActivity.kt

```
package com.example.prac_7(1)

import android.annotation.SuppressLint
import android.app.NotificationChannel
import android.app.NotificationManager
import android.app.PendingIntent
import android.content.Context
import android.content.Intent
import android.os.Build
import androidx.appcompat.app.AppCompatActivity
import android.os.Bundle
import android.widget.Button
import androidx.core.app.NotificationCompat

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

    @SuppressLint("MissingInflatedId")
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_main)
        btn = findViewById(R.id.btn)
        val mBuilder = NotificationCompat.Builder(this, "1")
            .setSmallIcon(R.drawable.ic_launcher_foreground)
            .setContentTitle("Notification Alert")
            .setContentText("Hi, This is your notification detail!")
            .setAutoCancel(true) // Closes the notification when clicked
        val intent = Intent(this, MainActivity::class.java)
        val pendingIntent = PendingIntent.getActivity(this, 0, intent,
PendingIntent.FLAG_IMMUTABLE)
        mBuilder.setContentIntent(pendingIntent)

        btn.setOnClickListener {
            val notificationManager =
getSystemService(Context.NOTIFICATION_SERVICE) as NotificationManager
            if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.O) {
                val channel = NotificationChannel("1", "My Channel",
NotificationManager.IMPORTANCE_DEFAULT)
                notificationManager.createNotificationChannel(channel)
            }
```

```
            val notificationId = 1
            notificationManager.notify(notificationId, mBuilder.build())
        }
    }
}
```

Output :

Practical -7(b)

Aim : Create an android application to generate two notifications, one notification will be non-clickable and another is clickable (it will reload the current application)

Code:

activity_main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.appcompat.widget.LinearLayoutCompat
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"
    tools:context=".MainActivity">

    <Toolbar
        android:layout_width="520dp"
        android:title="Prac-7"
        android:layout_marginRight="-520dp"
        android:background="@color/purple"
        android:layout_height="50dp">
    </Toolbar>

    <Button
        android:layout_width="match_parent"
        android:id="@+id/btnShowNonClickableNotification"
        android:text="Show Non-Clickable Notification"
        android:layout_height="wrap_content">
    </Button>
    <Button
        android:layout_width="match_parent"
        android:id="@+id/btnShowClickableNotification"
        android:text="Show Clickable Notification"
        android:layout_height="wrap_content">
    </Button>


</androidx.appcompat.widget.LinearLayoutCompat>
```

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">
    <uses-permission android:name="android.permission.POST_NOTIFICATIONS" />
    <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.Prac_7(2)A09"
        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>
```

MainActivity.kt

```
package com.example.prac_7(2)

import android.annotation.SuppressLint
import android.app.Notification
import android.app.NotificationChannel
import android.app.NotificationManager
import android.app.PendingIntent
import android.content.Context
import android.content.Intent
import android.os.Build
import android.os.Bundle
import android.widget.Button
import androidx.appcompat.app.AppCompatActivity
import androidx.core.app.NotificationCompat

class MainActivity : AppCompatActivity() {
    private val CHANNEL_ID = "my_channel"
    private val NOTIFICATION_ID_NON_CLICKABLE = 1
    private val NOTIFICATION_ID_CLICKABLE = 2
    @SuppressLint("MissingInflatedId")
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_main)
        createNotificationChannel()
        val showNonClickableNotificationButton: Button =
findViewById(R.id.btnShowNonClickableNotification)
        val showClickableNotificationButton: Button =
findViewById(R.id.btnShowClickableNotification)

        showNonClickableNotificationButton.setOnClickListener {
            showNonClickableNotification()
        }
        showClickableNotificationButton.setOnClickListener {
```

```
            showClickableNotification()
        }
    }
    private fun createNotificationChannel() {
        if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.O) {
            val name = "My Channel"
            val descriptionText = "My Channel Description"
            val importance = NotificationManager.IMPORTANCE_DEFAULT
            val channel = NotificationChannel(CHANNEL_ID, name,
importance).apply {
                description = descriptionText
            }
            val notificationManager: NotificationManager =
                getSystemService(Context.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_UPDATE_CURRENT or PendingIntent.FLAG_IMMUTABLE
        )
        val builder = NotificationCompat.Builder(this, CHANNEL_ID)
            .setSmallIcon(R.drawable.ic_launcher_foreground)
            .setContentTitle("Clickable Notification")
            .setContentText("Click to reload the app")
            .setContentIntent(pendingIntent)
            .setAutoCancel(true)
            .setPriority(NotificationCompat.PRIORITY_DEFAULT)

        val notificationManager: NotificationManager =
            getSystemService(Context.NOTIFICATION_SERVICE) as
NotificationManager
        notificationManager.notify(NOTIFICATION_ID_CLICKABLE, builder.build())
    }
    private fun showNonClickableNotification() {
        val builder = NotificationCompat.Builder(this, CHANNEL_ID)
            .setSmallIcon(R.drawable.ic_launcher_foreground)
            .setContentTitle("Non-Clickable Notification")
            .setContentText("This notification cannot be clicked.")
            .setAutoCancel(true)
            .setPriority(NotificationCompat.PRIORITY_DEFAULT)
        val notificationManager: NotificationManager =
            getSystemService(Context.NOTIFICATION_SERVICE) as
NotificationManager
        notificationManager.notify(NOTIFICATION_ID_NON_CLICKABLE,
builder.build())
    }
}
```

Output :


Prac-7
1:58
Show Non-Clickable Notification
Show Clickable Notification
1:1


1:58 PM • Wed, Feb 26
Prac_7(A-09) • now
Non-Clickable Notification
This notification cannot be clicked.
CLEAR ALL
1:1


1:59 PM • Wed, Feb 26
Prac_7(A-09) • now
Clickable Notification
Click to reload the app
CLEAR ALL
1:1


Prac-7
2:00
Show Non-Clickable Notification
Show Clickable Notification
1:1

Practical -8

Aim : Create an android application to create fragments, while user will click on button1 it will call fragment1 and while user will click on button2 it will call second fragment.

Code:

activity_main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.appcompat.widget.LinearLayoutCompat
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"
    tools:context=".MainActivity">
    <Toolbar
        android:layout_width="match_parent"
        android:title="Prac-8"
        android:background="@color/purple"
        android:titleTextColor="@color/white"
        android:layout_height="wrap_content">
    </Toolbar>
    <LinearLayout
        android:layout_width="match_parent"
        android:gravity="center"
        android:layout_marginTop="10dp"
        android:layout_height="wrap_content">
        <Button
            android:layout_width="wrap_content"
            android:id="@+id/btn1"
            android:text="Fragment1"
            android:layout_height="wrap_content">
        </Button>
        <Space
            android:layout_width="100dp"
            android:layout_height="wrap_content">
        </Space>
        <Button
            android:layout_width="wrap_content"
            android:id="@+id/btn2"
            android:text="Fragment2"
            android:layout_height="wrap_content">
        </Button>
    </LinearLayout>
    <FrameLayout
        android:id="@+id/frame_1"
        android:layout_width="match_parent"
```

```
        android:layout_height="match_parent">
    </FrameLayout>
</androidx.appcompat.widget.LinearLayoutCompat>
```

fragment_1.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"
    tools:context=".Fragment_1">
    <TextView
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:text="Hello world from Fragment_1"
        android:textColor="@color/white"
        android:background="@color/red"
        />
</FrameLayout>
```

fragment_2.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"
    tools:context=".Fragment_2">
    <TextView
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:background="@color/lightblue"
        android:text="Hello world from Fragment_2" />
</FrameLayout>
```

MainActvity.kt

```
package com.example.prac_8a
import android.annotation.SuppressLint
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 button1:Button
    lateinit var button2:Button
    @SuppressLint("MissingInflatedId")
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        enableEdgeToEdge()
        setContentView(R.layout.activity_main)
```

```
        button1=findViewById(R.id.btn1)
        button2=findViewById(R.id.btn2)

        button1.setOnClickListener{
            loadfrag1(Fragment_1())
        }
        button2.setOnClickListener{
            loadfrag2(Fragment_2())
        }
    }
    fun loadfrag1(fragment1: Fragment_1)
    {
        val ft1=supportFragmentManager.beginTransaction()
        ft1.replace(R.id.frame_1,fragment1)
        ft1.commit()
    }
    fun loadfrag2(fragment2: Fragment_2)
    {
        val ft1=supportFragmentManager.beginTransaction()
        ft1.replace(R.id.frame_1,fragment2)
        ft1.commit()
    }
}
```

Fragment_1.kt

```
package com.example.prac_8a
import android.os.Bundle
import androidx.fragment.app.Fragment
import android.view.LayoutInflater
import android.view.View
import android.view.ViewGroup
class Fragment_1 : Fragment() {
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
    }
    override fun onCreateView(
        inflater: LayoutInflater, container: ViewGroup?,
        savedInstanceState: Bundle?
    ): View? {
        return inflater.inflate(R.layout.fragment_1, container, false)
    }
}
```

Fragment_2.kt

```
package com.example.prac_8a_

import android.os.Bundle
import androidx.fragment.app.Fragment
import android.view.LayoutInflater
import android.view.View
import android.view.ViewGroup
class Fragment_2 : Fragment() {
```

```
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
    }
    override fun onCreateView(
        inflater: LayoutInflater, container: ViewGroup?,
        savedInstanceState: Bundle?
    ): View? {
        // Inflate the layout for this fragment
        return inflater.inflate(R.layout.fragment_2, container, false)
    }
}
```

Output :