

CO 4

1. Develop an application that use GridView with images and display Alert box on selection

xml

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

<GridView xmlns:android="http://schemas.android.com/apk/res/android"
xmlns:app="http://schemas.android.com/apk/res-auto" xmlns:tools="http://schemas.android.com/tools"
android:layout_width="match_parent" android:layout_height="match_parent"

tools:context=".MainActivity" android:id="@+id/gridview" android:numColumns="auto_fit"
android:columnWidth="90dp" android:verticalSpacing="6dp" android:horizontalSpacing="6dp"
android:stretchMode="columnWidth" android:gravity="center">

</GridView>
```

java

```
package com.example.gridviewadapter;

import androidx.appcompat.app.AlertDialog;
import androidx.appcompat.app.AppCompatActivity;

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

import android.widget.AdapterView; import android.widget.AdapterView; import
android.widget.GridView;

public class MainActivity extends AppCompatActivity { GridView gv;

String[] gridViewString={"desktop","raspberry","sensor","ultrasonic","desk","raspberr","sensors",
"ul trasonics","fff"};

public int

imgarray[]={R.drawable.a,R.drawable.b,R.drawable.c,R.drawable.d,R.drawable.e,R.drawabl
e.a,R.drawable.b,R.drawable.c,R.drawable.d};

@Override
```

```

protected void onCreate(Bundle savedInstanceState) { super.onCreate(savedInstanceState);
setContentView(R.layout.activity_main); ImageAdapter imageAdapter=new
ImageAdapter(this,R.layout.iview,gridViewString,imgarray); gv=findViewById(R.id.gridview);
gv.setAdapter(imageAdapter);

gv.setOnItemClickListener(new AdapterView.OnItemClickListener() { @Override
public void onItemClick(AdapterView<?> parent, View view, int position, long id) { AlertDialog.Builder bldr;
bldr=new AlertDialog.Builder(MainActivity.this); bldr.setTitle("Title");
AlertDialog alrt=bldr.create(); alrt.setMessage(gridViewString[position]); alrt.show();

}

});

}

}

```

ImageAdapter

```

package com.example.gridviewadapter;

import android.content.Context; import android.view.LayoutInflater; import android.view.View;
import android.view.ViewGroup; import android.widget.AdapterView; import android.widget.ImageView;
import android.widget.TextView;

public class ImageAdapter extends ArrayAdapter { private Context mContext;
private String gvstring[]; private int imgarray[]; LayoutInflater inflater = null;

public ImageAdapter(Context c, int resource, String s[], int x[]) { super(c, resource);
mContext = c; gvstring = s; imgarray = x;
inflater = LayoutInflater.from(c);
}

```

```
@Override  
  
public int getCount() { return imgarray.length;  
}
```

```
@Override  
  
public Object getItem(int position) { return null;  
  
}
```

```
@Override  
  
public View getView(int position, View convertView, ViewGroup parent) { if (convertView == null)  
    convertView = inflater.inflate(R.layout.iview, null);  
  
    TextView country = (TextView) convertView.findViewById(R.id.tv1); ImageView icon = (ImageView)  
    convertView.findViewById(R.id.imageV1); icon.setImageResource(imgarray[position]);  
    country.setText(gvstring[position]);  
  
    return convertView;  
}  
}
```

iview.xml

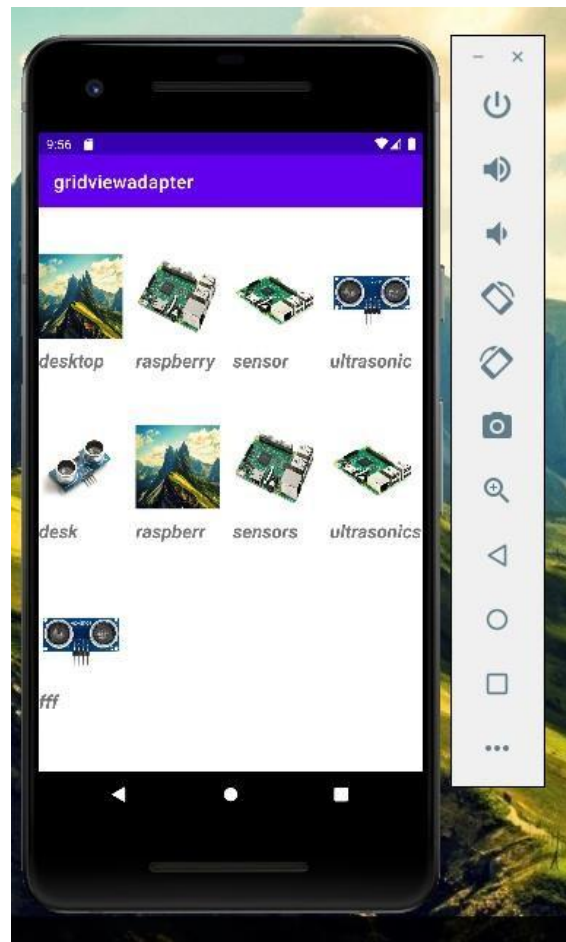
```
<?xml version="1.0" encoding="utf-8"?>  
  
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"  
    xmlns:app="http://schemas.android.com/apk/res-auto" xmlns:tools="http://schemas.android.com/tools"  
    android:orientation="vertical"  
  
    android:layout_width="match_parent" android:layout_height="match_parent">  
  
  
  
    <ImageView android:id="@+id/imageV1" android:layout_width="90dp" android:layout_height="90dp"  
        android:scaleType="fitXY" android:layout_marginTop="50dp"  
  
        tools:srcCompat="@tools:sample/avatars" />
```

```
<TextView android:id="@+id/tv1"
    android:layout_width="match_parent" android:layout_height="wrap_content"

    android:text="TextView" android:textSize="20dp" android:textStyle="italic|bold"
    android:layout_marginTop="10dp"/>

</LinearLayout>
```

Output



2. Develop an application that implements Spinner component and perform event handling

XML

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

<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".MainActivity"
    android:orientation="horizontal">

    <Spinner
        android:id="@+id/spinner1"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_weight="1" />

    <Spinner
        android:id="@+id/spinner2"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_weight="1" />

</LinearLayout>
```

Curserview.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/tv1"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="TextView" />
</LinearLayout>
```

tv.xml

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

JAVA

```
package com.example.spinner;

import androidx.appcompat.app.AppCompatActivity;

import android.database.Cursor;
import android.database.sqlite.SQLiteDatabase;
import android.os.Bundle;
import android.view.View;
import android.widget.AdapterView;
import android.widget.AdapterView.OnItemClickListener;
import android.widget.ArrayAdapter;
import android.widget.CursorAdapter;
import android.widget.SimpleCursorAdapter;
import android.widget.Spinner;
import android.widget.Toast;

public class MainActivity extends AppCompatActivity {

    Spinner sp,sp1;

    SQLiteDatabase db;

    String s[]={"Red","Green","Blue","Yellow"};

    @Override
    protected void onCreate(Bundle savedInstanceState) {

        super.onCreate(savedInstanceState);

        setContentView(R.layout.activity_main);

        sp = (Spinner) findViewById(R.id.spinner1);
        sp1 = (Spinner) findViewById(R.id.spinner2);

        ArrayAdapter<String> aa = new ArrayAdapter<String>(this, R.layout.tv, s);
        sp.setAdapter(aa);
```

```

sp.setOnItemSelectedListener(new AdapterView.OnItemSelectedListener() {

    @Override

    public void onItemSelected(AdapterView<?> parent, View view, int position, long id) {

        Toast.makeText(getApplicationContext(), s[position], Toast.LENGTH_SHORT).show();

    }

    @Override

    public void onNothingSelected(AdapterView<?> parent) {

        return;

    }

});

try {

    db = openOrCreateDatabase("db1", MODE_PRIVATE, null);

    db.execSQL("create table emp(eno integer,ename ttext,ecountry text)");

    db.execSQL("insert into emp values(1,'Nisha','India')");

    db.execSQL("insert into emp values(2,'Maya','US')");

    db.execSQL("insert into emp values(3,'Manju','UK')");

    db.execSQL("insert into emp values(4,'Neenu','Canada')");

} catch (Exception e) {

}

Cursor c = db.rawQuery("select eno as _id,ename as name,ecountry as country from emp", null);

String s1[] = {"country"};

int ids[] = {R.id.tv1};

CursorAdapter ca = new SimpleCursorAdapter(this, R.layout.cursorview, c, s1, ids);

sp1.setAdapter(ca);

sp1.setOnItemSelectedListener(new AdapterView.OnItemSelectedListener() {

    @Override

    public void onItemSelected(AdapterView<?> parent, View view, int position, long id) {

        Cursor item=(Cursor)parent.getSelectedItem();

```



```
String value=(item.getString(2));

Toast.makeText(getApplicationContext(),value,Toast.LENGTH_SHORT).show();
}

@Override
public void onNothingSelected(AdapterView<?> parent) {

}

});

}

}
```

OUTPUT



3. Apply themes via code and manifest file

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" android:theme="@style/mystyle">

<TextView android:layout_width="wrap_content" android:layout_height="wrap_content"
android:text="Hello World!"

app:layout_constraintBottom_toBottomOf="parent" app:layout_constraintLeft_toLeftOf="parent"
app:layout_constraintRight_toRightOf="parent" app:layout_constraintTop_toTopOf="parent"
style="@style/mystyle"/>

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

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

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

<application android:allowBackup="true" android:icon="@mipmap/ic_launcher"
android:label="@string/app_name"

android:roundIcon="@mipmap/ic_launcher_round" android:supportsRtl="true"
android:theme="@style/Theme.AppCompat.Light.NoActionBar">

<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>
```

night/themes

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

<!-- Base application theme. -->

<style name="Theme.Stylesandthemes" parent="Theme.MaterialComponents.DayNight.DarkActionBar">
```

```

<!-- Primary brand color. -->
<item name="colorPrimary">@color/purple_200</item>
<item name="colorPrimaryVariant">@color/purple_700</item>
<item name="colorOnPrimary">@color/black</item>
<!-- Secondary brand color. -->
<item name="colorSecondary">@color/teal_200</item>
<item name="colorSecondaryVariant">@color/teal_200</item>
<item name="colorOnSecondary">@color/black</item>
<!-- Status bar color. -->
<item name="android:statusBarColor" tools:targetApi="I"?attr/colorPrimaryVariant</item>
<!-- Customize your theme here. -->
</style>
</resources> value/themes
<resources xmlns:tools="http://schemas.android.com/tools">
<!-- Base application theme. -->
<style name="Theme.Stylesandthemes" parent="Theme.MaterialComponents.DayNight.DarkActionBar">
<!-- Primary brand color. -->
<item name="colorPrimary">@color/purple_500</item>
<item name="colorPrimaryVariant">@color/purple_700</item>
<item name="colorOnPrimary">@color/white</item>
<!-- Secondary brand color. -->
<item name="colorSecondary">@color/teal_200</item>
<item name="colorSecondaryVariant">@color/teal_700</item>
<item name="colorOnSecondary">@color/black</item>
<!-- Status bar color. -->
<item name="android:statusBarColor" tools:targetApi="I"?attr/colorPrimaryVariant</item>
<!-- Customize your theme here. -->
</style>
<style name="mystyle" parent="Theme.AppCompat.DayNight.DarkActionBar">
<item name="android:textColor">#1BEF70</item>
<item name="android:textStyle">italic</item>

```

```
<item name="android:textSize">30sp</item>
```

```
</style>
```

```
</resources>
```

Mainactivity

```
package com.example.stylesandthemes;
```

```
import androidx.appcompat.app.AppCompatActivity; import android.os.Bundle;
```

```
public class MainActivity extends AppCompatActivity {
```

```
@Override
```

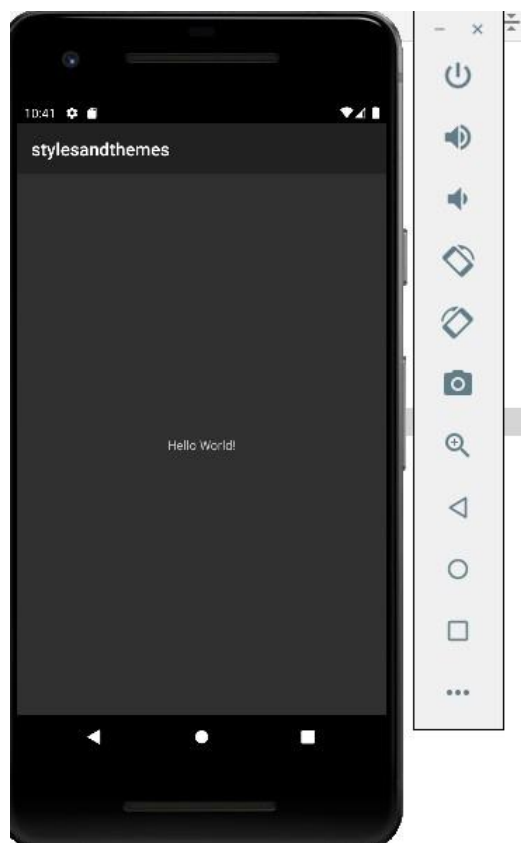
```
protected void onCreate(Bundle savedInstanceState) { super.onCreate(savedInstanceState);
```

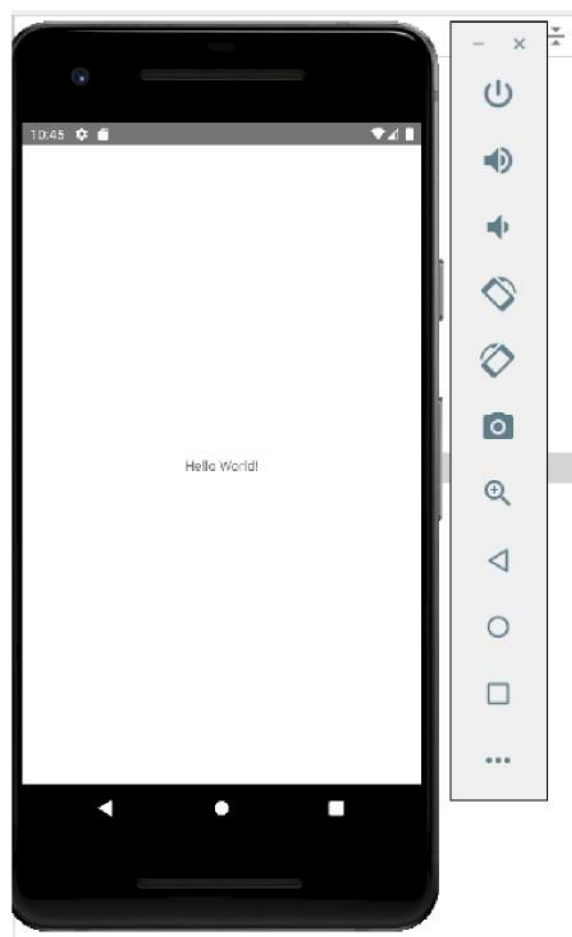
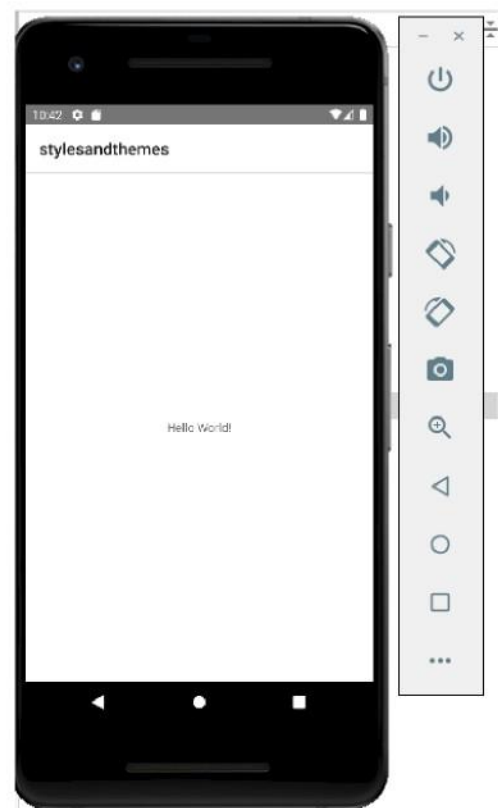
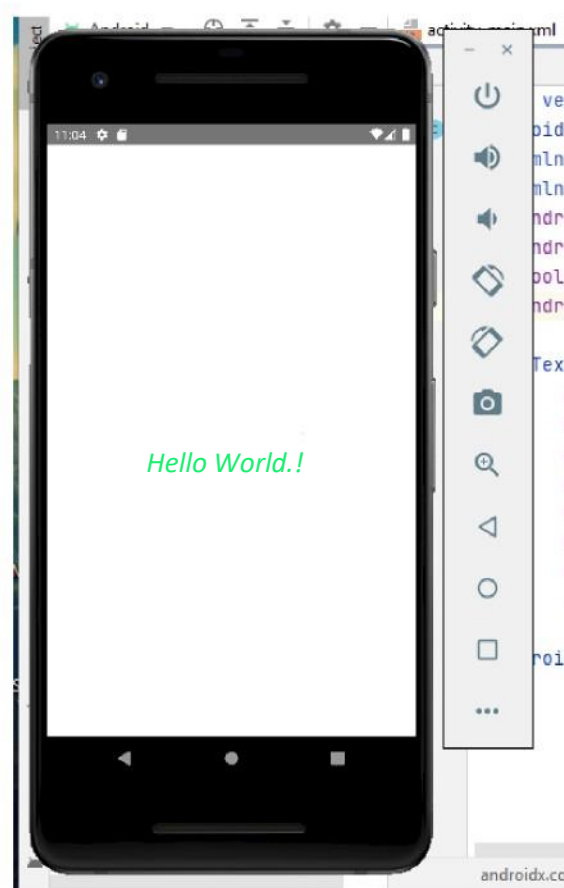
```
setContentView(R.layout.activity_main);
```

```
}
```

```
}
```

Output





4. Develop application using Fragments

xml

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

<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto" xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent" android:layout_height="match_parent"
    android:orientation="vertical" tools:context=".MainActivity">

    <TextView android:layout_width="wrap_content" android:layout_height="wrap_content"
        android:text="Hello World!"

        app:layout_constraintBottom_toBottomOf="parent" app:layout_constraintLeft_toLeftOf="parent"
        app:layout_constraintRight_toRightOf="parent" app:layout_constraintTop_toTopOf="parent" />

    <Button

        android:id="@+id/firstFragment" android:layout_width="match_parent"
        android:layout_height="wrap_content" android:background="@color/black" android:textSize="20sp"
        android:textColor="#CDDC39" android:text="firstfragment" />

    <Button

        android:id="@+id/secondFragment" android:layout_width="match_parent"
        android:layout_height="wrap_content" android:layout_marginTop="10dp"
        android:background="@color/black" android:textSize="20sp" android:textColor="#CDDC39"
        android:text="secondfragment" />

    <FrameLayout android:id="@+id/FrameLayout" android:layout_width="match_parent"

        android:layout_height="match_parent" android:layout_marginTop="10dp">

    </FrameLayout>
```

```
</LinearLayout>
```

java

```
package com.example.fragmentexample;
```

```
import androidx.appcompat.app.AppCompatActivity; import androidx.fragment.app.Fragment;
```

```
import androidx.fragment.app.FragmentManager; import androidx.fragment.app.FragmentTransaction;
```

```
import android.os.Bundle; import android.view.View; import android.widget.Button;
```

```
public class MainActivity extends AppCompatActivity { Button firstFragment,secondFragment;
```

```
@Override
```

```
protected void onCreate(Bundle savedInstanceState) { super.onCreate(savedInstanceState);  
setContentView(R.layout.activity_main); firstFragment=(Button) findViewById(R.id.firstFragment);
```

```
secondFragment=(Button) findViewById(R.id.secondFragment); firstFragment.setOnClickListener(new  
View.OnClickListener() {
```

```
@Override
```

```
public void onClick(View v) { loadFragment(new FirstFragment());
```

```
}
```

```
});
```

```
secondFragment.setOnClickListener(new View.OnClickListener() { @Override
```

```
public void onClick(View v) { loadFragment(new SecondFragment());
```

```
}
```

```
});
```

```
}
```

```
private void loadFragment(Fragment fragment)
```

```
{
```

```
FragmentManager fm=getSupportFragmentManager(); FragmentTransaction  
fragmentTransaction=fm.beginTransaction(); fragmentTransaction.replace(R.id.FrameLayout,fragment);  
fragmentTransaction.commit();
```

```
}  
}
```

firstFragment.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=".FirstFragment">  
  
    <!-- TODO: Update blank fragment layout -->  
  
    <TextView android:layout_width="match_parent" android:layout_height="match_parent"  
        android:text="This is first fragment" />  
  
    <Button  
  
        android:id="@+id/firstButton" android:layout_marginTop="40dp" android:layout_width="wrap_content"  
        android:layout_height="wrap_content" android:text="FirstFragment" />  
  
</FrameLayout>
```

firstFragment.java

```
package com.example.fragmentexample; import android.os.Bundle;  
  
import androidx.fragment.app.Fragment;  
  
  
import android.view.LayoutInflater; import android.view.View;  
import android.view.ViewGroup;  
  
  
/**  
  
    A simple {@link Fragment} subclass.
```


Use the {@link FirstFragment#newInstance} factory method to create an instance of this fragment.

**/*

```
public class FirstFragment extends Fragment {
```

```
// TODO: Rename parameter arguments, choose names that match
```

```
// the fragment initialization parameters, e.g. ARG_ITEM_NUMBER
```

```
private static final String ARG_PARAM1 = "param1"; private static final String ARG_PARAM2 = "param2";
```

```
// TODO: Rename and change types of parameters
```

```
private String mParam1; private String mParam2;
```

```
public FirstFragment() {
```

```
// Required empty public constructor
```

```
}
```

```
/**
```

```
Use this factory method to create a new instance of  
this fragment using the provided parameters.
```

```
*
```

```
@param param1 Parameter 1.
```

```
@param param2 Parameter 2.
```

```
@return A new instance of fragment FirstFragment.
```

```
*/
```

```
// TODO: Rename and change types and number of parameters
```

```
public static FirstFragment newInstance(String param1, String param2) { FirstFragment fragment = new  
FirstFragment();
```

```
Bundle args = new Bundle(); args.putString(ARG_PARAM1, param1); args.putString(ARG_PARAM2,  
param2); fragment.setArguments(args);
```

```
return fragment;
```

```
}
```

@Override

```
public void onCreate(Bundle savedInstanceState) { super.onCreate(savedInstanceState);  
if (getArguments() != null) {  
    mParam1 = getArguments().getString(ARG_PARAM1); mParam2 =  
    getArguments().getString(ARG_PARAM2);  
}  
}
```

@Override

```
public View onCreateView(LayoutInflater inflater, ViewGroup container, Bundle savedInstanceState) {  
// Inflate the layout for this fragment  
return inflater.inflate(R.layout.fragment_first, container, false);  
}  
}
```

secondFragment.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=".SecondFragment">
```

```
<!-- TODO: Update blank fragment layout -->
```

```
<TextView android:layout_width="match_parent" android:layout_height="match_parent"  
    android:text="This is secondfragment" />
```

```
<Button
```

```
android:id="@+id/secondbutton" android:layout_marginTop="40dp"
android:layout_width="wrap_content" android:layout_height="wrap_content"
android:text="secondFragment" />
```

</FrameLayout> **secondFragment.java**

```
package com.example.fragmentexample; import android.os.Bundle;
```

```
import androidx.fragment.app.Fragment;
```

```
import android.view.LayoutInflater; import android.view.View;
```

```
import android.view.ViewGroup;
```

```
/**
```

```
A simple {@link Fragment} subclass.
```

```
Use the {@link SecondFragment#newInstance} factory method to  
create an instance of this fragment.
```

```
*/
```

```
public class SecondFragment extends Fragment {
```

```
// TODO: Rename parameter arguments, choose names that match
```

```
// the fragment initialization parameters, e.g. ARG_ITEM_NUMBER
```

```
private static final String ARG_PARAM1 = "param1"; private static final String ARG_PARAM2 = "param2";
```

```
// TODO: Rename and change types of parameters
```

```
private String mParam1; private String mParam2;
```

```
public SecondFragment() {
```

```
// Required empty public constructor
```

```
}
```

```
/**
```

*Use this factory method to create a new instance of
this fragment using the provided parameters.*

@param param1 Parameter 1.

@param param2 Parameter 2.

@return A new instance of fragment SecondFragment.

**/*

// TODO: Rename and change types and number of parameters

```
public static SecondFragment newInstance(String param1, String param2) { SecondFragment fragment =  
new SecondFragment();
```

```
Bundle args = new Bundle(); args.putString(ARG_PARAM1, param1); args.putString(ARG_PARAM2,  
param2); fragment.setArguments(args);
```

```
return fragment;
```

```
}
```

@Override

```
public void onCreate(Bundle savedInstanceState) { super.onCreate(savedInstanceState);
```

```
if (getArguments() != null) {
```

```
mParam1 = getArguments().getString(ARG_PARAM1); mParam2 =  
getArguments().getString(ARG_PARAM2);
```

```
}
```

```
}
```

@Override

```
public View onCreateView(LayoutInflater inflater, ViewGroup container, Bundle savedInstanceState) {
```

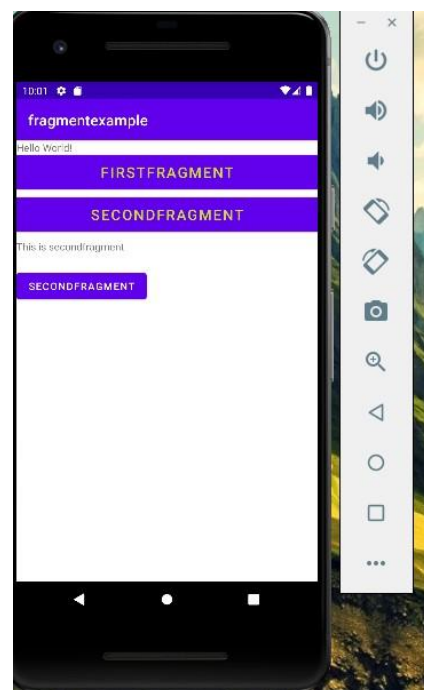
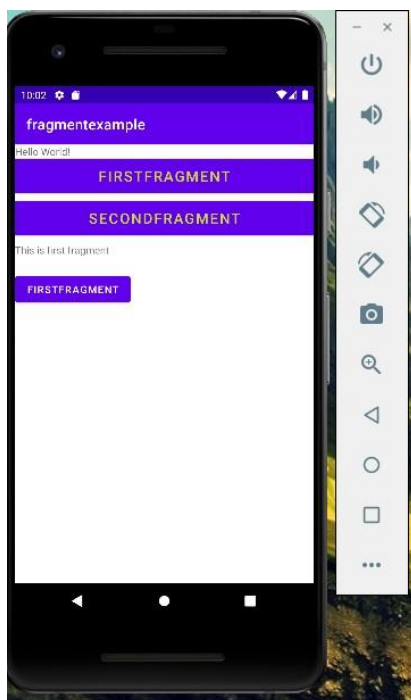
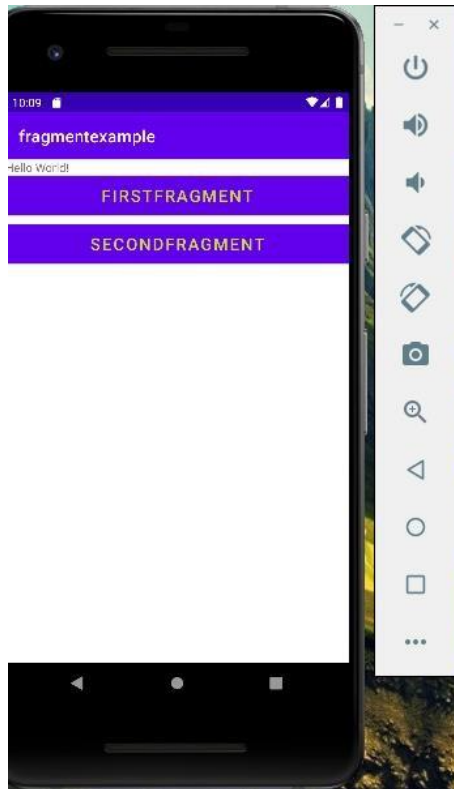
```
// Inflate the layout for this fragment
```

```
return inflater.inflate(R.layout.fragment_second, container, false);
```

```
}
```

```
}
```

Output



5. Implement Navigation drawer

Activity_main.xml

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

<androidx.drawerlayout.widget.DrawerLayout
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/my_drawer_layout" android:layout_width="match_parent"
android:layout_height="match_parent" tools:context=".MainActivity">

<LinearLayout android:layout_width="wrap_content" android:layout_height="wrap_content"
android:orientation="vertical" android:gravity="center">

<TextView android:layout_width="wrap_content" android:layout_height="wrap_content"
android:textSize="50dp" android:text="Navigation!!!" android:textColor="#B71DD1"

app:layout_constraintBottom_toBottomOf="parent" app:layout_constraintLeft_toLeftOf="parent"
app:layout_constraintRight_toRightOf="parent" app:layout_constraintTop_toTopOf="parent" />

</LinearLayout>

<com.google.android.material.navigation.NavigationView android:layout_width="wrap_content"
android:layout_height="match_parent" android:id="@+id/nav_view" android:layout_gravity="start"
app:menu="@menu/navigationmenu"/>

</androidx.drawerlayout.widget.DrawerLayout >
```

Mainactivity.java

```
package com.example.navigationdrawer; import androidx.annotation.NonNull;

import androidx.appcompat.app.ActionBarDrawerToggle; import
androidx.appcompat.app.AppCompatActivity; import androidx.drawerlayout.widget.DrawerLayout;
import android.os.Bundle;

import android.view.MenuItem; import android.widget.Toast;

import com.example.navigationdrawer.R;

import com.google.android.material.navigation.NavigationView; public class MainActivity extends
AppCompatActivity {

DrawerLayout dl; ActionBarDrawerToggle abd; @Override
```

```

protected void onCreate(Bundle savedInstanceState) { super.onCreate(savedInstanceState);
setContentView(R.layout.activity_main); dl=(DrawerLayout)findViewById(R.id.my_drawer_layout);

abdg =new ActionBarDrawerToggle(this,dl, R.string.nav_open,R.string.nav_close);
dl.addDrawerListener(abdg);

abdg.syncState(); getSupportActionBar().setDisplayHomeAsUpEnabled(true);

NavigationView navView = (NavigationView) findViewById(R.id.nav_view);
navView.setNavigationItemSelectedListener(new

NavigationView.OnNavigationItemSelectedListener() { @Override

public boolean onNavigationItemSelectedListener(@NonNull MenuItem menuItem)

{

int itemId = menuItem.getItemId(); if (itemId == R.id.nav_account) {

Toast.makeText(getApplicationContext(), menuItem.getTitle(), Toast.LENGTH_SHORT).show();

} else if (itemId == R.id.nav_settings) { Toast.makeText(getApplicationContext(),

menuItem.getTitle(), Toast.LENGTH_SHORT).show();

} else if (itemId == R.id.nav_logout) { Toast.makeText(getApplicationContext(),

menuItem.getTitle(), Toast.LENGTH_SHORT).show();

} return false;

}

});

}

public boolean onOptionsItemSelected(@NonNull MenuItem item)

{

if (abdg.onOptionsItemSelected(item)) return true;

return super.onOptionsItemSelected(item);

}

}

```

Navigationmenu.xml

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

  android:title="Settings" />
  <item android:id="@+id/nav_logout" android:title="Logout" />
</menu>
```

Strings.xml

```
<resources>
  <string name="app_name">navigation</string>
  <string name="nav_open">Open</string>
  <string name="nav_close">Close</string>
</resources>
```


Output

