



## **Code Day Assignment**

Oliver Njeru-663565

School of Sciences and Technology, United States International University-Africa

APT3060A: Mobile Programming

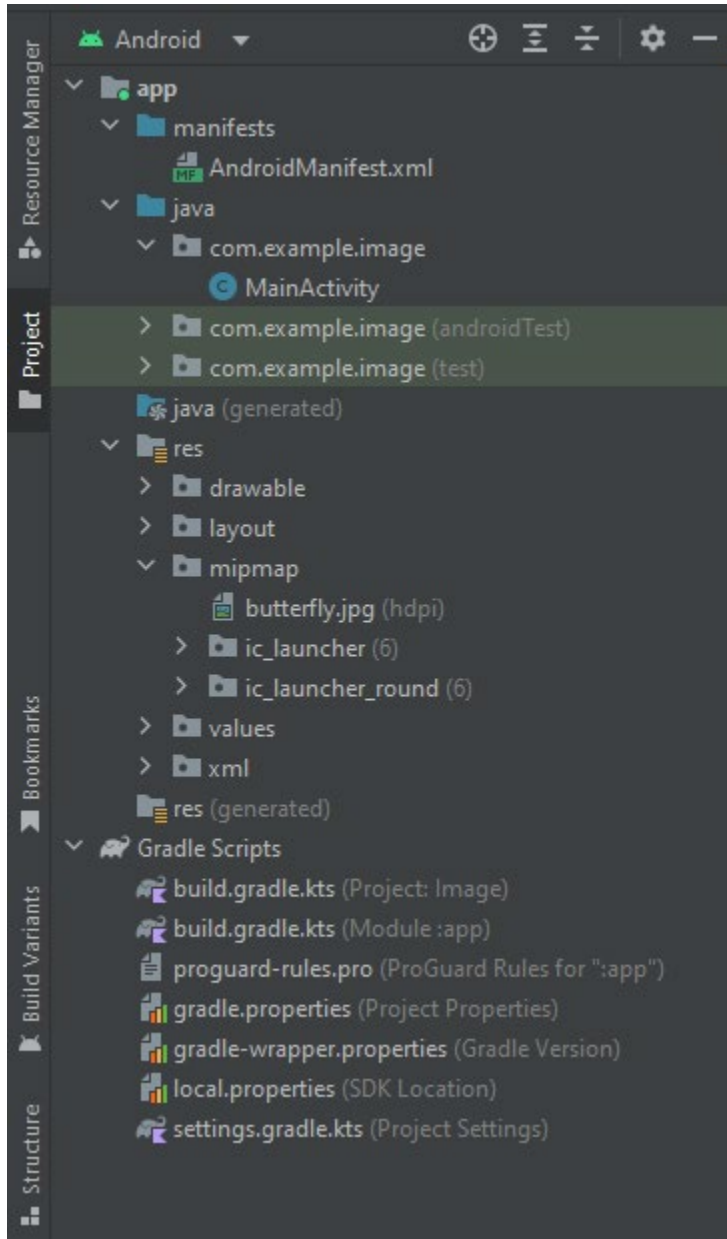
Dr. Lawrence Nderu

November 05, 2023

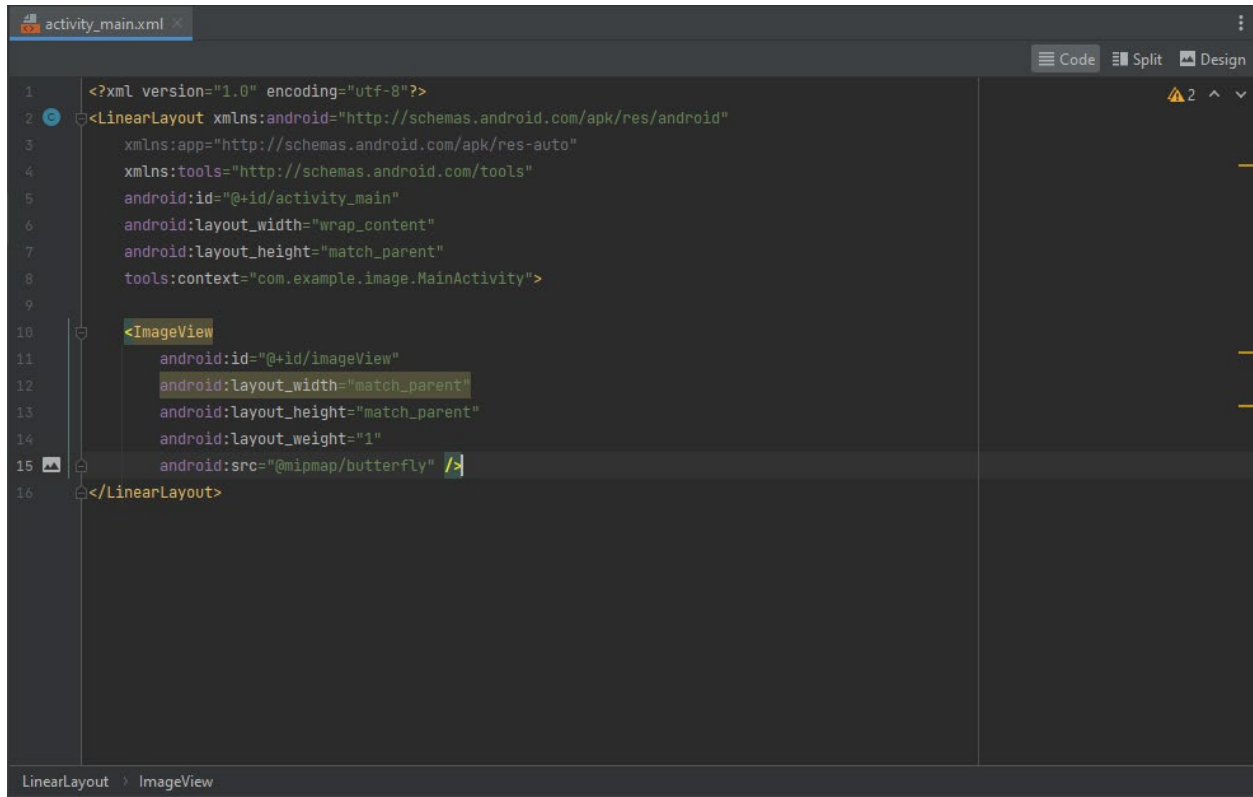
## Code Day Assignment.

### Image Project

In this project, our goal is to create an application that displays an image of a butterfly. The first thing I did was to add the butterfly image in the mipmap folder as shown below.



The next thing I did was navigate to the activity\_main.xml file and add a LinearLayout tag as well as an ImageView tag that is going to locate the image and display it when the application is run. The code added is shown below.



```
1  <?xml version="1.0" encoding="utf-8"?>
2  <LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
3      xmlns:app="http://schemas.android.com/apk/res-auto"
4      xmlns:tools="http://schemas.android.com/tools"
5      android:id="@+id/activity_main"
6      android:layout_width="wrap_content"
7      android:layout_height="match_parent"
8      tools:context="com.example.image.MainActivity">
9
10     <ImageView
11         android:id="@+id/imageView"
12         android:layout_width="match_parent"
13         android:layout_height="match_parent"
14         android:layout_weight="1"
15         android:src="@mipmap/butterfly" />
16 </LinearLayout>
```

The screenshot shows an IDE window titled 'activity\_main.xml'. The code is in XML format, defining a LinearLayout container with an ImageView inside. The ImageView is configured with 'match\_parent' dimensions and a weight of 1, pointing to a resource named 'mipmap/butterfly'. The IDE interface includes a 'Code' tab, a 'Split' button, and a 'Design' button. A breadcrumb at the bottom indicates the current view is 'LinearLayout > ImageView'.

The next step is to run the application. What the application displays is as shown below.

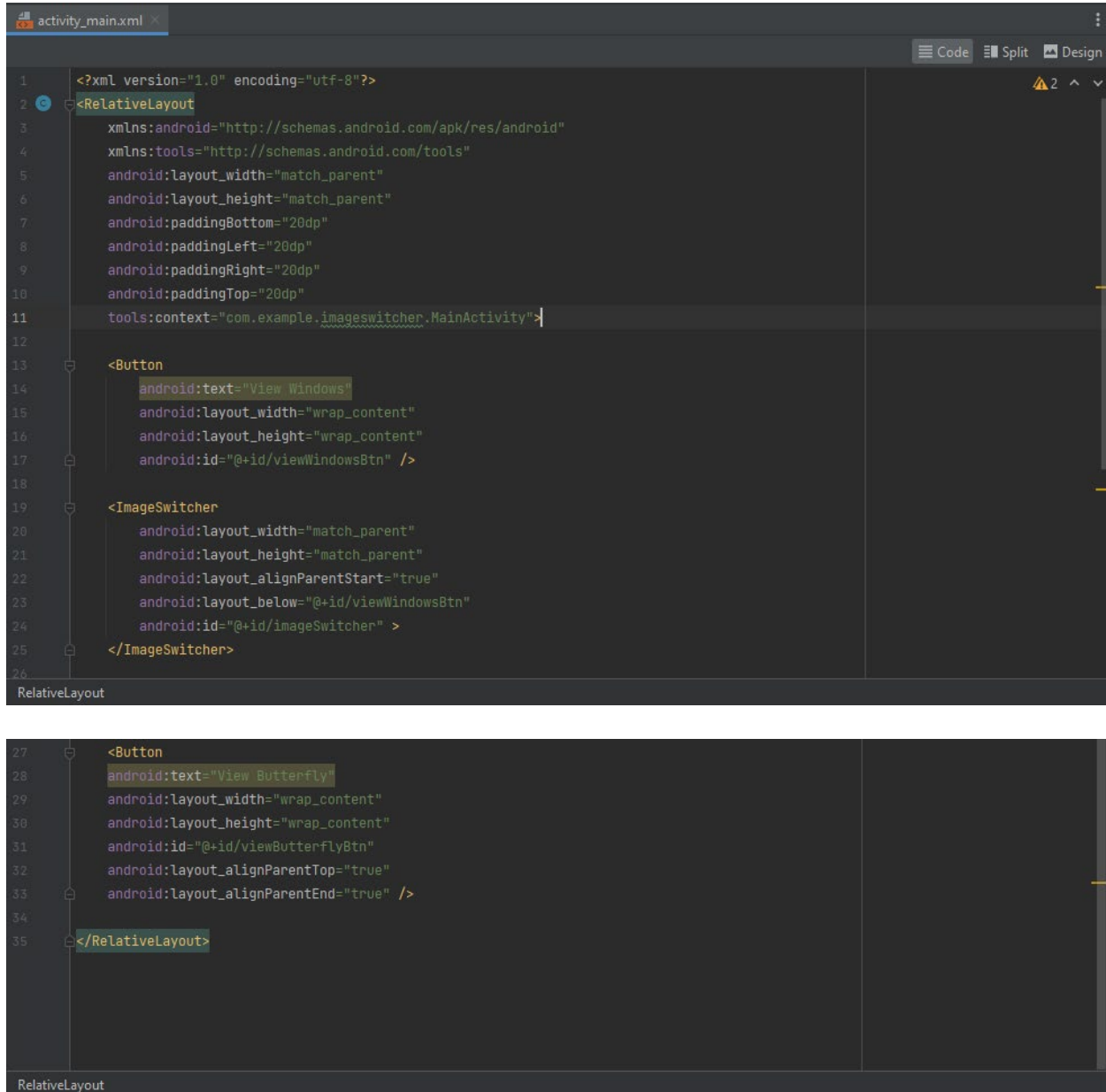


This marks the completion of this project's goal.

## ImageSwitcher

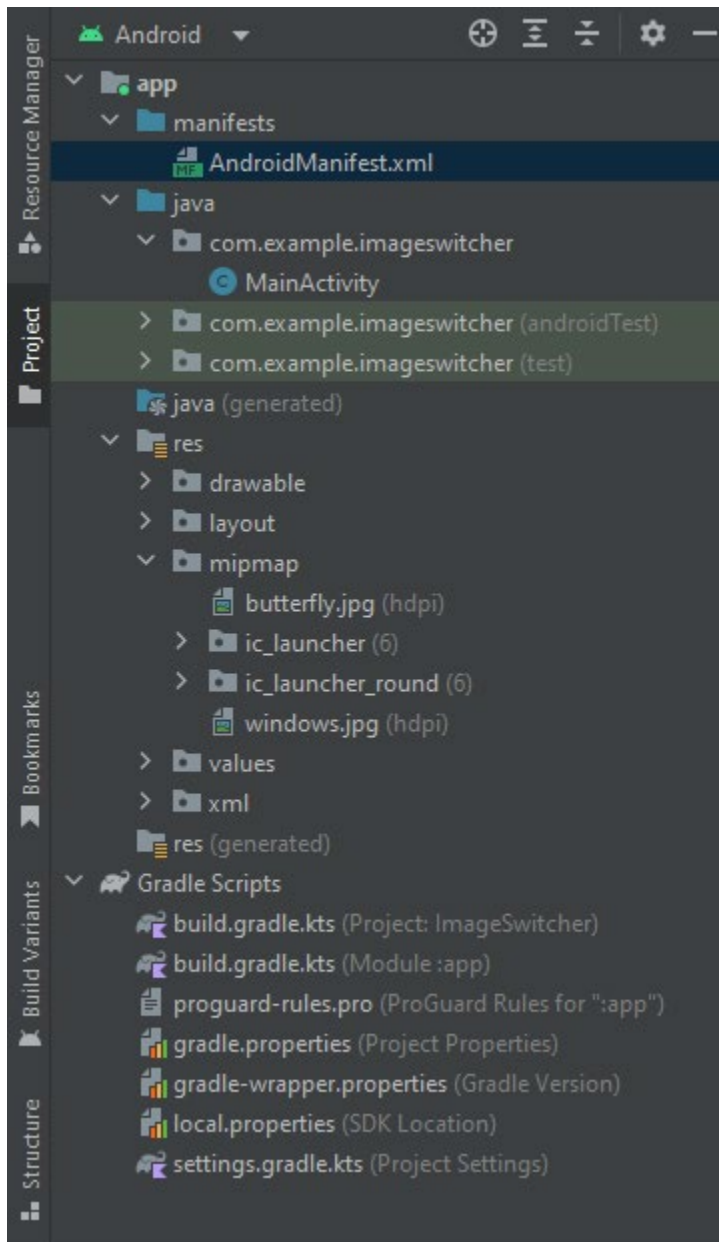
This project builds off on top of the previous project. It displays more than one image with a transition in between them.

In the activity\_main.xml file, I added a RelativeLayout tag, two Button tags and an ImageSwitcher tag as shown below.

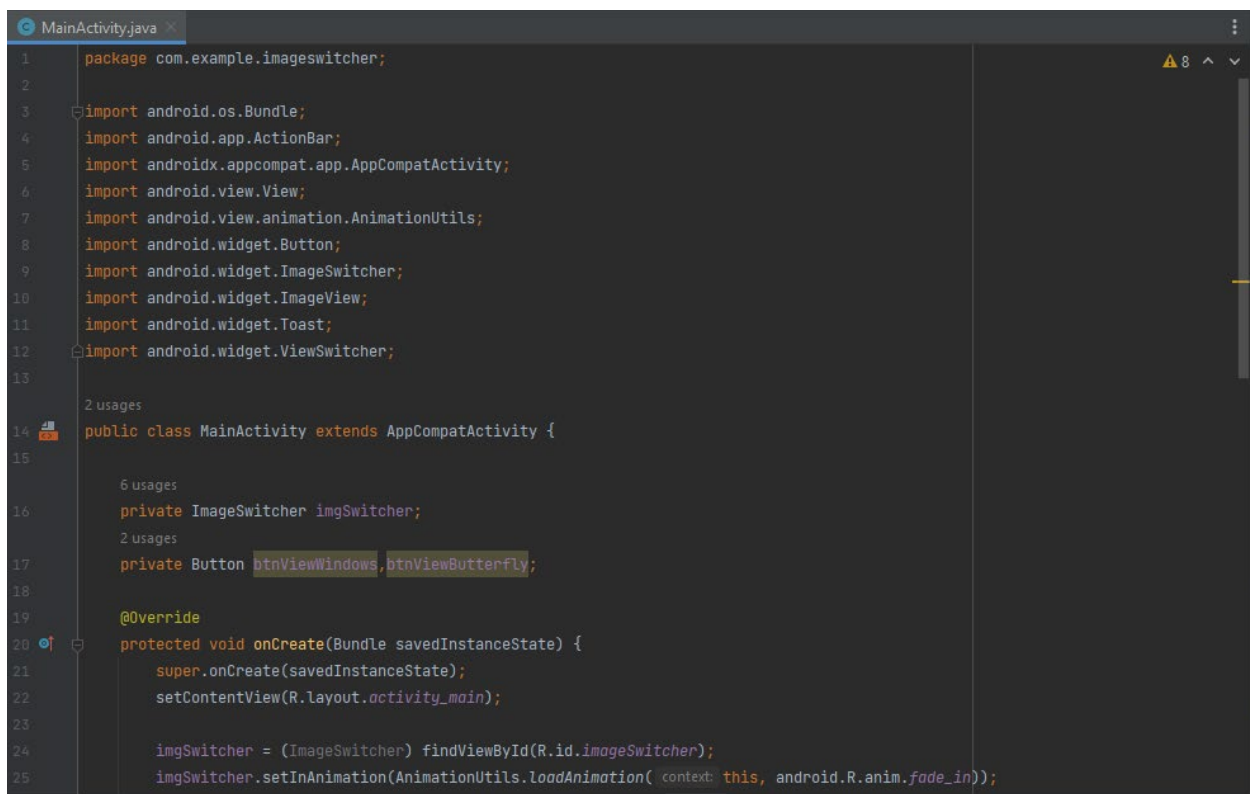


```
1  <?xml version="1.0" encoding="utf-8"?>
2  <RelativeLayout
3      xmlns:android="http://schemas.android.com/apk/res/android"
4      xmlns:tools="http://schemas.android.com/tools"
5      android:layout_width="match_parent"
6      android:layout_height="match_parent"
7      android:paddingBottom="20dp"
8      android:paddingLeft="20dp"
9      android:paddingRight="20dp"
10     android:paddingTop="20dp"
11     tools:context="com.example.imageswitcher.MainActivity">
12
13     <Button
14         android:text="View Windows"
15         android:layout_width="wrap_content"
16         android:layout_height="wrap_content"
17         android:id="@+id/viewWindowsBtn" />
18
19     <ImageSwitcher
20         android:layout_width="match_parent"
21         android:layout_height="match_parent"
22         android:layout_alignParentStart="true"
23         android:layout_below="@+id/viewWindowsBtn"
24         android:id="@+id/imageSwitcher" >
25     </ImageSwitcher>
26
27     <Button
28         android:text="View Butterfly"
29         android:layout_width="wrap_content"
30         android:layout_height="wrap_content"
31         android:id="@+id/viewButterflyBtn"
32         android:layout_alignParentTop="true"
33         android:layout_alignParentEnd="true" />
34
35 </RelativeLayout>
```

I then added two images in the mipmap folder that I am going to use in this project. The images are shown below.



In MainActivity.java file, I added the following code. Three private variables with an ImageSwitcher type and Button type. Initialize imgSwitcher variable by finding ImageSwitcher widget using its id and casting the result of findViewById to the ImageSwitcher type. I then set the animation for ImageSwitcher transitions using a method to load fade-in and fade-out animations from the android system resources. I initialized the two buttons using their ids. I added a method that makes the image view. I added onClickListener methods for each of the two buttons to set the value for the toast message and view switch the correct image accordingly using the imgSwitcher variable.

A screenshot of an IDE showing the MainActivity.java file. The code is written in Java and includes package declarations, imports for various Android classes, and the implementation of the MainActivity class. The MainActivity class extends AppCompatActivity and contains private variables for ImageSwitcher and two Buttons. The onCreate method is overridden, showing the initialization of the ImageSwitcher and the first button, along with setting the initial animation.

```
1 package com.example.imageswitcher;
2
3 import android.os.Bundle;
4 import android.app.ActionBar;
5 import androidx.appcompat.app.AppCompatActivity;
6 import android.view.View;
7 import android.view.animation.AnimationUtils;
8 import android.widget.Button;
9 import android.widget.ImageSwitcher;
10 import android.widget.ImageView;
11 import android.widget.Toast;
12 import android.widget.ViewSwitcher;
13
14 public class MainActivity extends AppCompatActivity {
15
16     private ImageSwitcher imgSwitcher;
17     private Button btnViewWindows, btnViewButterfly;
18
19     @Override
20     protected void onCreate(Bundle savedInstanceState) {
21         super.onCreate(savedInstanceState);
22         setContentView(R.layout.activity_main);
23
24         imgSwitcher = (ImageSwitcher) findViewById(R.id.imageSwitcher);
25         imgSwitcher.setInAnimation(AnimationUtils.loadAnimation(context: this, android.R.anim.fade_in));
```

```

MainActivity.java
26    imgSwitcher.setOutAnimation(AnimationUtils.loadAnimation( context: this, android.R.anim.fade_out));
27    btnViewWindows = (Button) findViewById(R.id.viewWindowsBtn);
28    btnViewButterfly = (Button) findViewById(R.id.viewButterflyBtn);
29
30    imgSwitcher.setFactory(new ViewSwitcher.ViewFactory() {
31        1 usage
32        @Override
33        public View makeView() {
34            ImageView myView = new ImageView(getApplicationContext());
35            myView.setScaleType(ImageView.ScaleType.FIT_CENTER);
36            myView.setLayoutParams(new ImageSwitcher.LayoutParams(ActionBar.LayoutParams.WRAP_CONTENT,
37                ActionBar.LayoutParams.WRAP_CONTENT));
38            return myView;
39        }
40    });
41
42    btnViewWindows.setOnClickListener(new View.OnClickListener() {
43        @Override
44        public void onClick(View v) {
45            Toast.makeText(getApplicationContext()
46                , text: "View Windows",Toast.LENGTH_LONG).show();
47            imgSwitcher.setImageResource(R.mipmap.windows);
48        }
49    });
50
51    btnViewButterfly.setOnClickListener(new View.OnClickListener() {
52        @Override
53        public void onClick(View v) {
54            Toast.makeText(getApplicationContext(), text: "View Butterfly" ,Toast.LENGTH_LONG).show();
55        }
56    });
57
58

```

```

53    imgSwitcher.setImageResource(R.mipmap.butterfly);
54    }
55    });
56    }
57    }
58

```



When the application is run, the following is seen.

On run:



On click of View Windows Button:



On click of View Butterfly button:

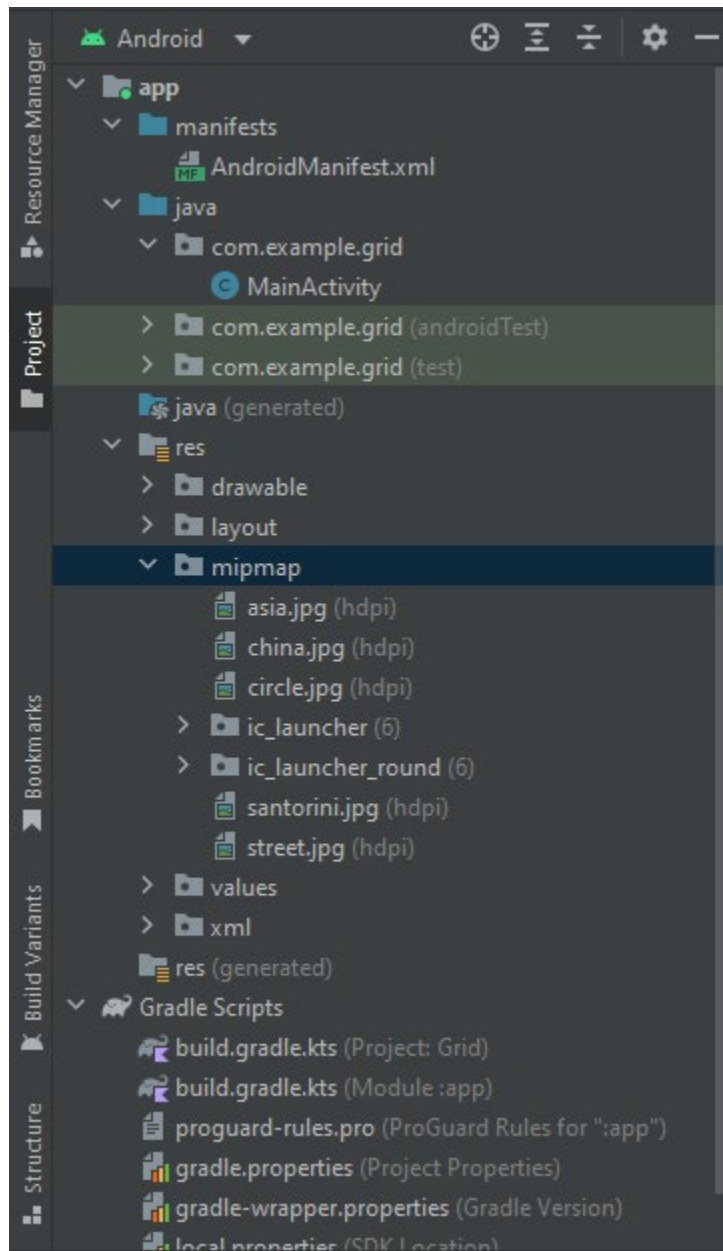


This marks the completion of the objective of this project.

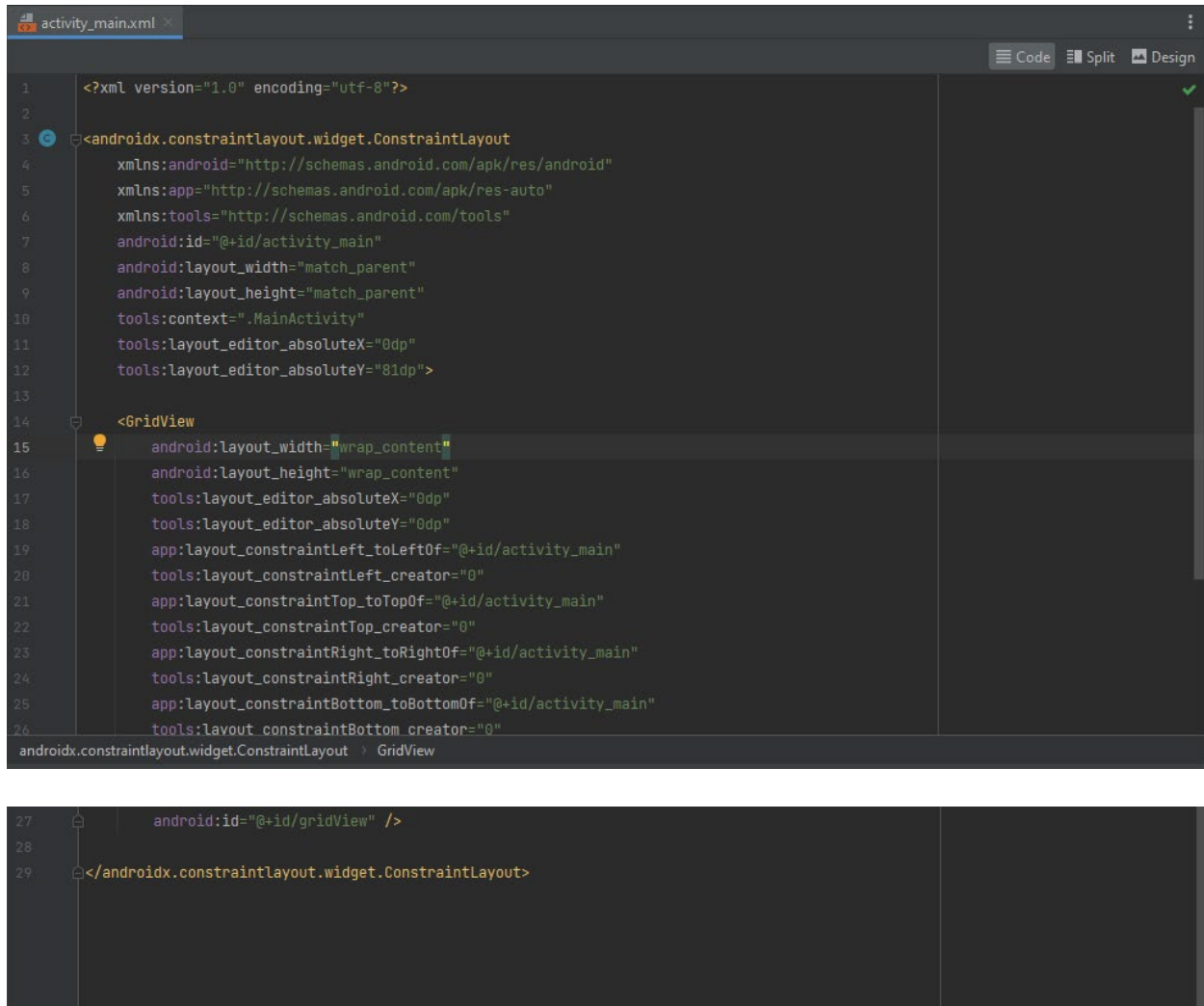
## Grid

The objective of this project is to display images in a grid view and show a toast message of the image selected and its position.

In the mipmap folder, I added five images, asia, china, circle, santorini and street that are going to be used in this project as shown below.

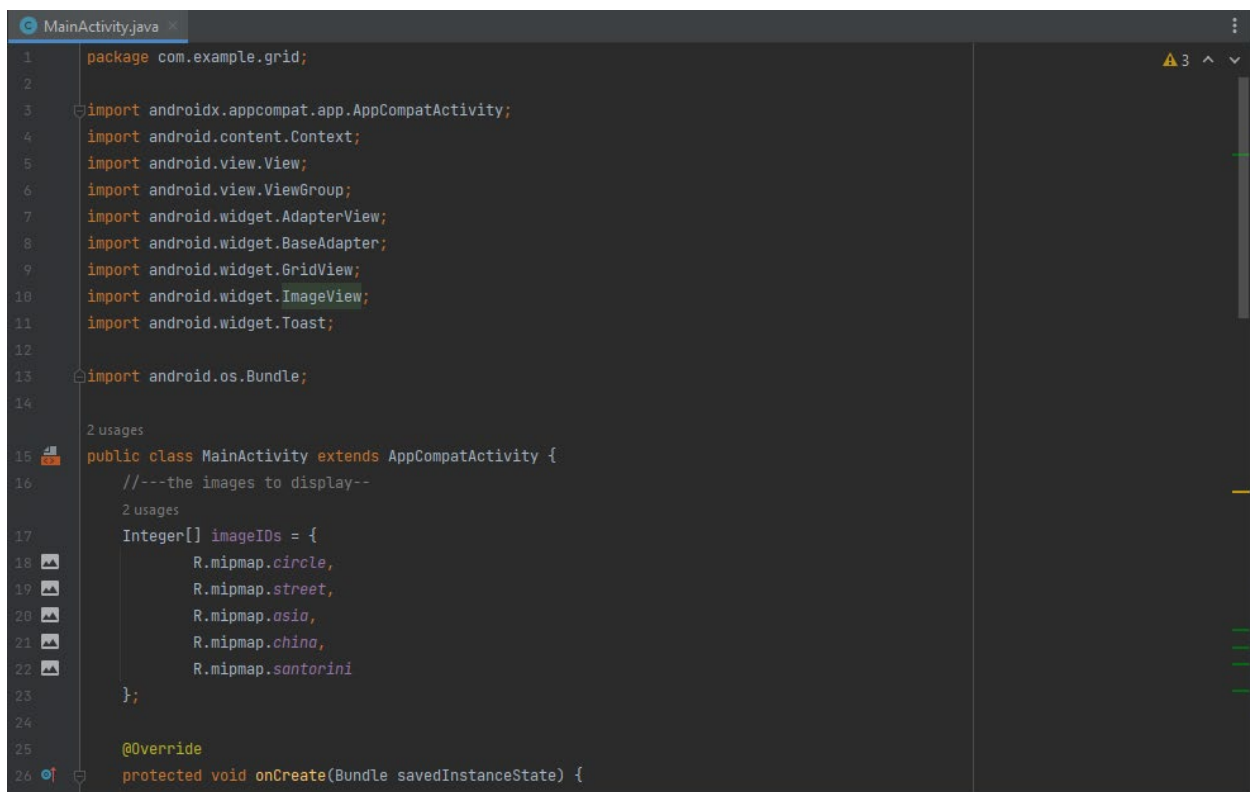


In the activity\_main.xml file, I added a constraint layout and a GridView tag as shown below.



```
1  <?xml version="1.0" encoding="utf-8"?>
2
3  <androidx.constraintlayout.widget.ConstraintLayout
4      xmlns:android="http://schemas.android.com/apk/res/android"
5      xmlns:app="http://schemas.android.com/apk/res-auto"
6      xmlns:tools="http://schemas.android.com/tools"
7      android:id="@+id/activity_main"
8      android:layout_width="match_parent"
9      android:layout_height="match_parent"
10     tools:context=".MainActivity"
11     tools:layout_editor_absoluteX="0dp"
12     tools:layout_editor_absoluteY="81dp">
13
14     <GridView
15         android:layout_width="wrap_content"
16         android:layout_height="wrap_content"
17         tools:layout_editor_absoluteX="0dp"
18         tools:layout_editor_absoluteY="0dp"
19         app:layout_constraintLeft_toLeftOf="@+id/activity_main"
20         tools:layout_constraintLeft_creator="0"
21         app:layout_constraintTop_toTopOf="@+id/activity_main"
22         tools:layout_constraintTop_creator="0"
23         app:layout_constraintRight_toRightOf="@+id/activity_main"
24         tools:layout_constraintRight_creator="0"
25         app:layout_constraintBottom_toBottomOf="@+id/activity_main"
26         tools:layout_constraintBottom_creator="0"
27         android:id="@+id/gridView" />
28
29 </androidx.constraintlayout.widget.ConstraintLayout>
```

In the MainActivity.java file, I created an array that is going to store the images located and referenced in the mipmap folder. In the onCreate method, I created a gridView variable of type GridView and initialized it to be the gridView widget in the activity\_main.xml file by finding its id. I then set an onClickListener to track the position of the image and set the toast message. I created an if statement to check on the convertView value and given it is null, to set the imageView layout, scale type and padding. Else, assign the variable imageView to be convertView casted to type ImageView. Then I set the imageView to the image depending on the position and returned the imageView. The code is as shown below.

A screenshot of an IDE showing the MainActivity.java file. The code includes package, imports, and a class definition. The imports section lists various Android classes like AppCompatActivity, Context, View, ViewGroup, AdapterView, BaseAdapter, GridView, ImageView, Toast, and Bundle. The class MainActivity extends AppCompatActivity and contains an array of image IDs and an onCreate method.

```
1 package com.example.grid;
2
3 import androidx.appcompat.app.AppCompatActivity;
4 import android.content.Context;
5 import android.view.View;
6 import android.view.ViewGroup;
7 import android.widget.AdapterView;
8 import android.widget.BaseAdapter;
9 import android.widget.GridView;
10 import android.widget.ImageView;
11 import android.widget.Toast;
12
13 import android.os.Bundle;
14
15 2 usages
16 public class MainActivity extends AppCompatActivity {
17     //---the images to display---
18     2 usages
19     Integer[] imageIDs = {
20         R.mipmap.circle,
21         R.mipmap.street,
22         R.mipmap.asia,
23         R.mipmap.china,
24         R.mipmap.santorini
25     };
26
27     @Override
28     protected void onCreate(Bundle savedInstanceState) {
```

```
MainActivity.java
27     super.onCreate(savedInstanceState);
28     setContentView(R.layout.activity_main);
29
30     GridView gridView = (GridView) findViewById(R.id.gridView);
31     gridView.setAdapter(new ImageAdapter( c this));
32     gridView.setOnItemClickListener(new AdapterView.OnItemClickListener() {
33     public void onItemClick(AdapterView parent,
34                             View v,
35                             int position,
36                             long id) {
37         Toast.makeText(getBaseContext(),
38                       text: "pic" + (position + 1) + " selected",
39                       Toast.LENGTH_SHORT).show();
40     }
41 });
42 }
43
44 1 usage
45 public class ImageAdapter extends BaseAdapter {
46     2 usages
47     private Context context;
48
49     1 usage
50     public ImageAdapter(Context c) { context = c; }
51
52     //---returns the number of images--
53     public int getCount() { return imageIDs.length; }
54
55 }
```

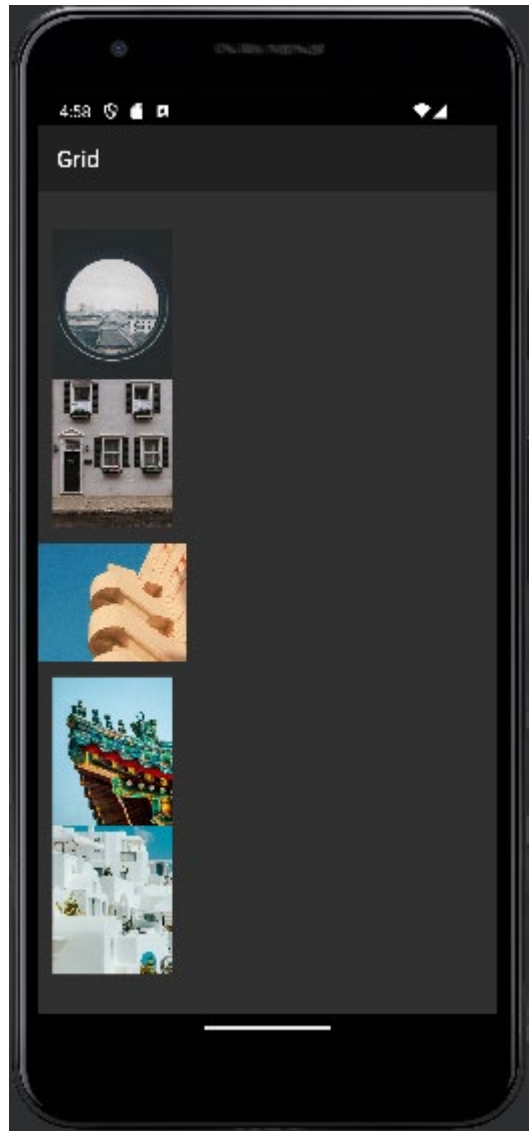
```
56     //---returns the item--
57     public Object getItem(int position) { return position; }
58
59
60     //---returns the ID of an item--
61     public long getItemId(int position) { return position; }
62
63
64     //---returns an ImageView view--
65     public View getView(int position, View convertView, ViewGroup parent) {
66         ImageView imageView;
67         if (convertView == null) {
68             imageView = new ImageView(context);
69             imageView.setLayoutParams(new GridView.LayoutParams( w: 350, h: 350));
70             imageView.setScaleType(ImageView.ScaleType.CENTER_CROP);
71             imageView.setPadding( left: 35, top: 35, right: 35, bottom: 35);
72         } else {
73             imageView = (ImageView) convertView;
74         }
75         imageView.setImageResource(imageIDs[position]);
76         return imageView;
77     }
78
79 }
80
81 }
```

com.example.grid.MainActivity

Integer[] imageIDs  
= {...

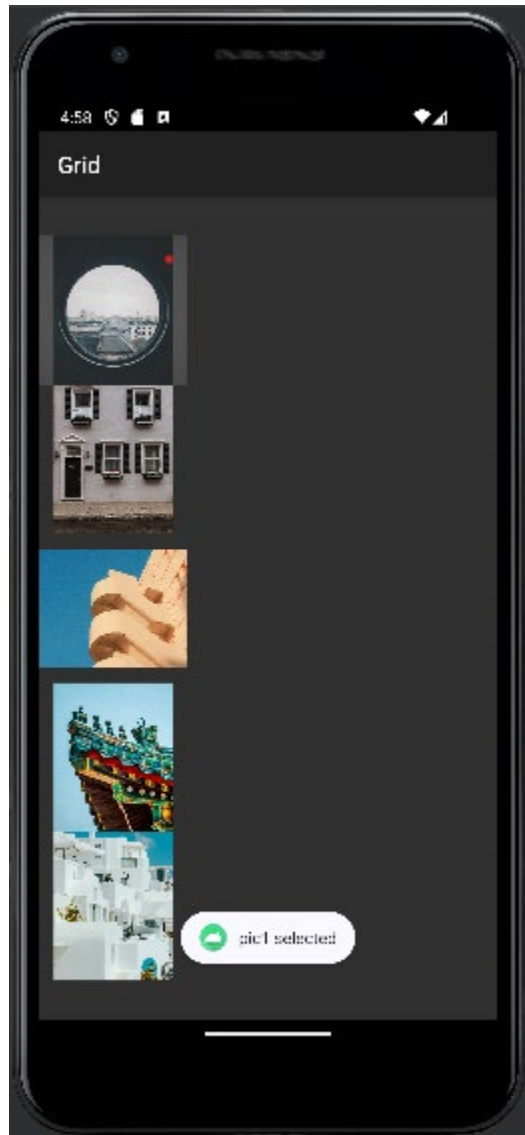
Grid.app.main

When the application is run, the following screen is displayed:

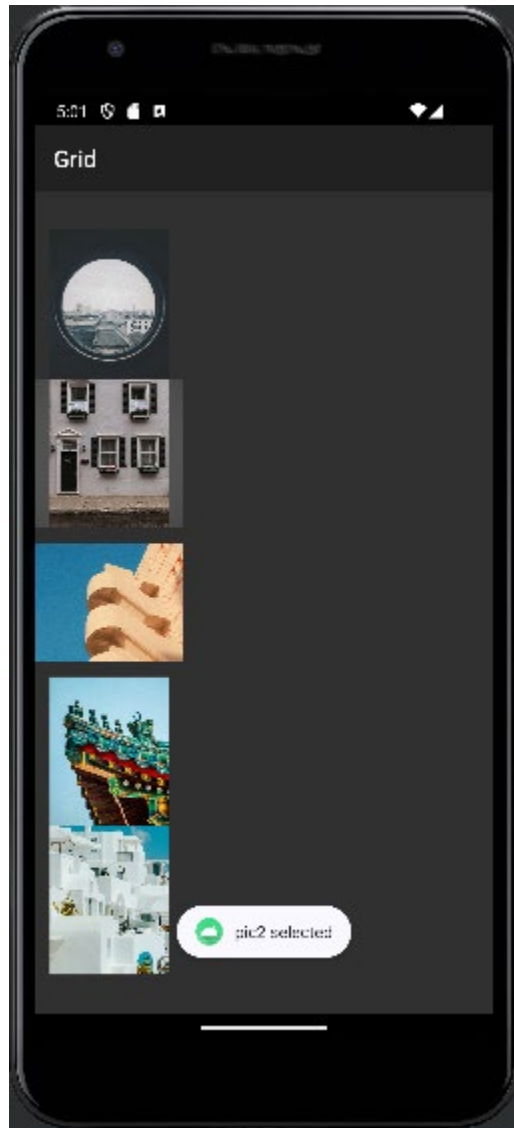




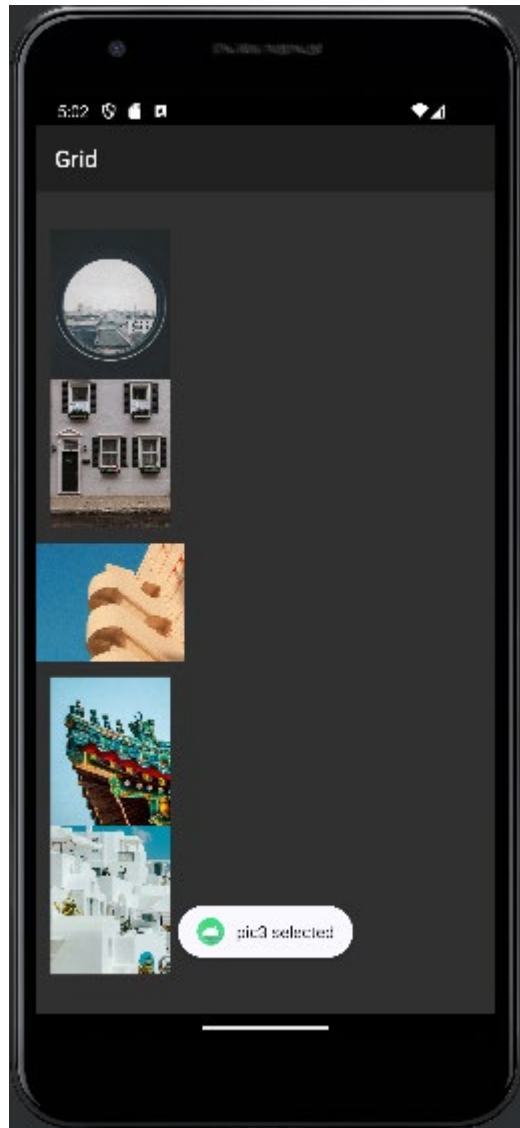
When I tap on the first image, the toast message is as show:



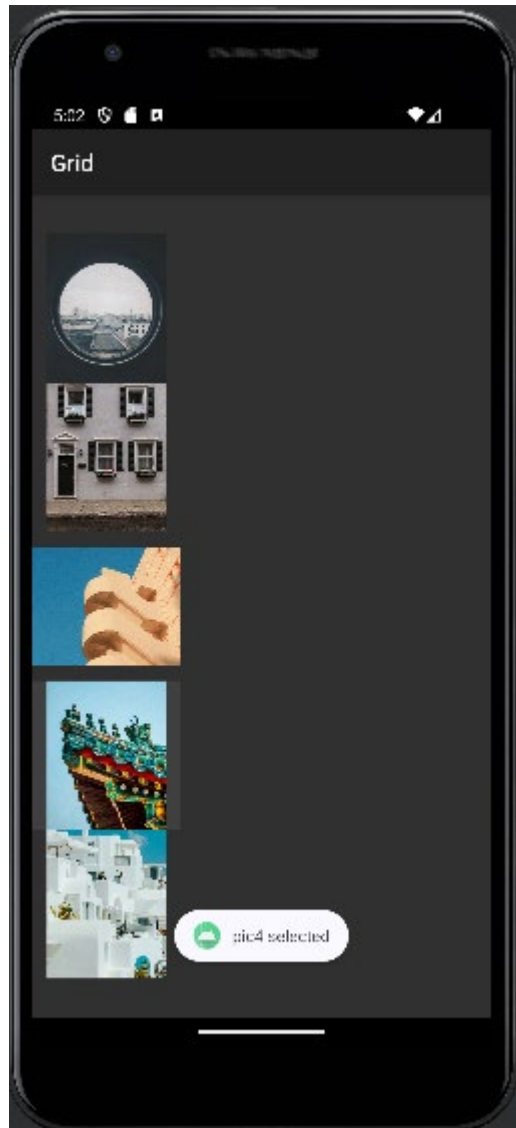
When I tap on the second image, the toast message is as shown:



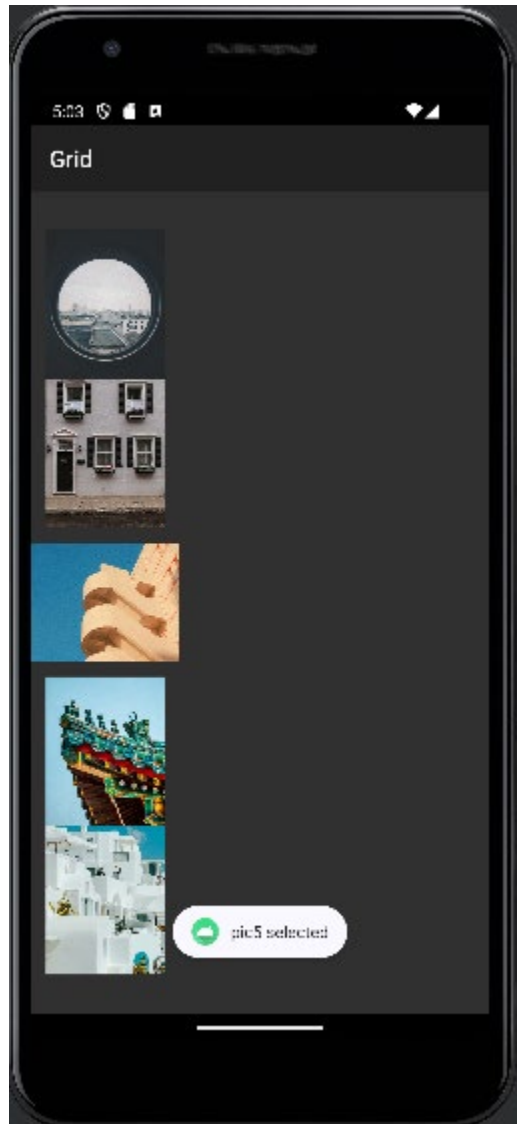
When I tap on the third image, the toast message is as shown:



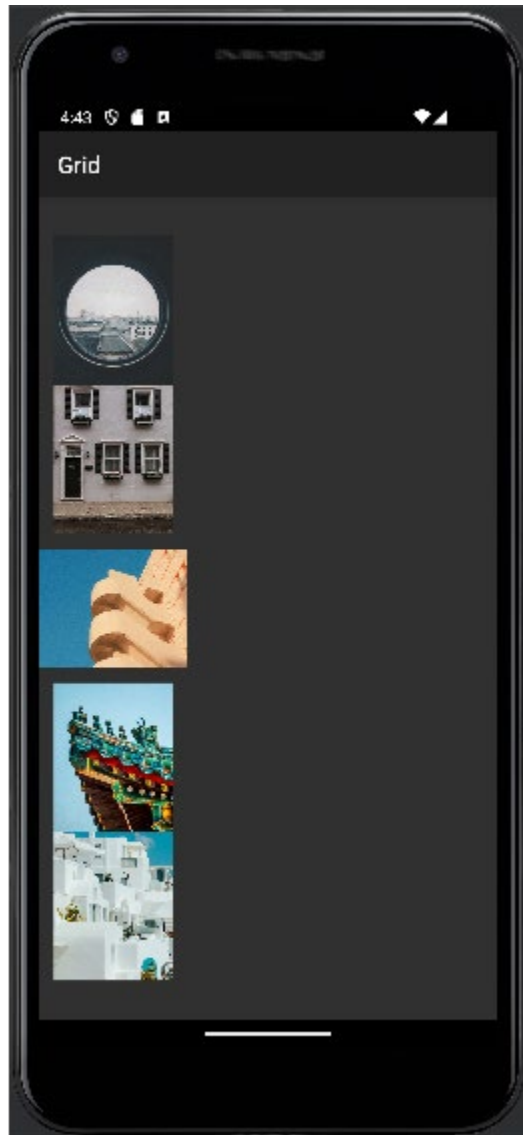
When I tap on the fourth image, the toast message is as shown:



When I tap on the fifth image, the toast message is as shown:



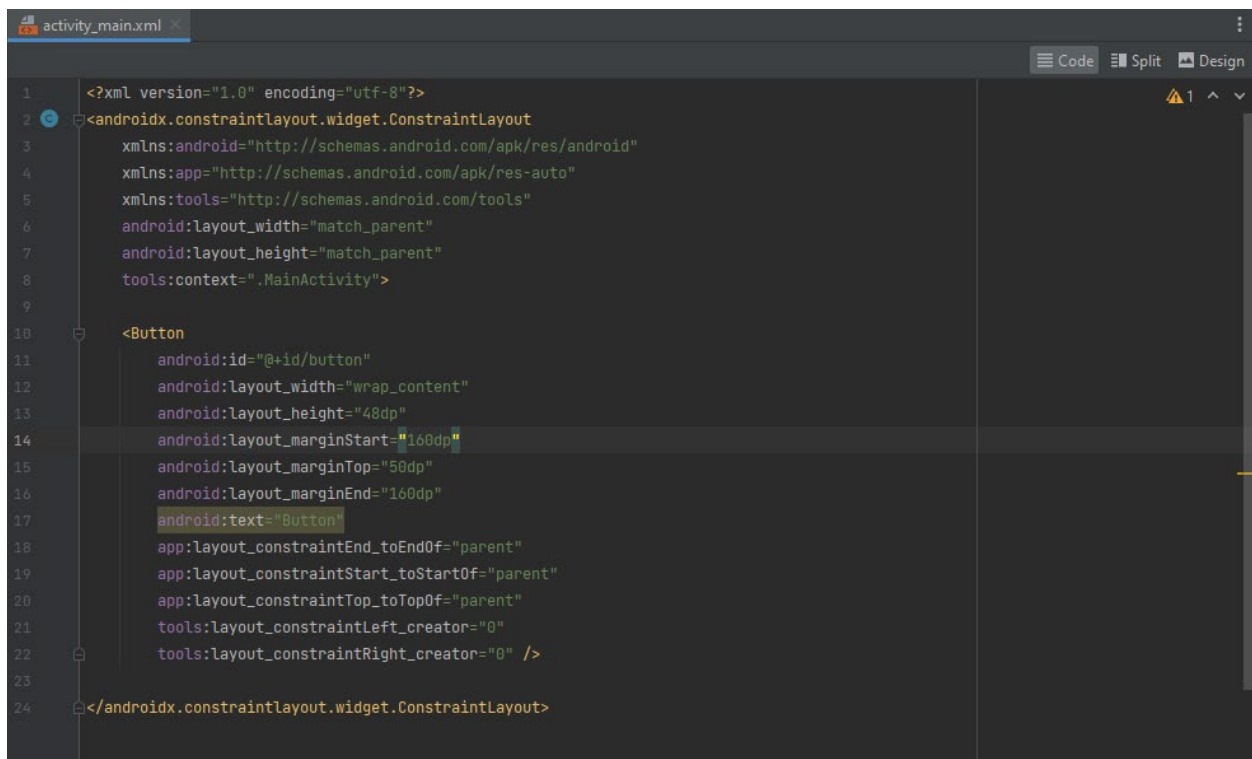
This shows that we have achieved the project's objective.



## Menu

The objective of this project is to create a menu list and have a button display a menu list when a user presses and holds the button. The menu options, when clicked, should display a toast message indicating that the user clicked on an option with its appropriate number.

In activity\_main.xml file, I added a constraint layout and a button widget with id button as show below.



```
1  <?xml version="1.0" encoding="utf-8"?>
2  <androidx.constraintlayout.widget.ConstraintLayout
3      xmlns:android="http://schemas.android.com/apk/res/android"
4      xmlns:app="http://schemas.android.com/apk/res-auto"
5      xmlns:tools="http://schemas.android.com/tools"
6      android:layout_width="match_parent"
7      android:layout_height="match_parent"
8      tools:context=".MainActivity">
9
10     <Button
11         android:id="@+id/button"
12         android:layout_width="wrap_content"
13         android:layout_height="48dp"
14         android:layout_marginStart="160dp"
15         android:layout_marginTop="50dp"
16         android:layout_marginEnd="160dp"
17         android:text="Button"
18         app:layout_constraintEnd_toEndOf="parent"
19         app:layout_constraintStart_toStartOf="parent"
20         app:layout_constraintTop_toTopOf="parent"
21         tools:layout_constraintLeft_creator="0"
22         tools:layout_constraintRight_creator="0" />
23
24 </androidx.constraintlayout.widget.ConstraintLayout>
```

In MainActivity.java file, below is the code. I initialized a button variable to be the id of the widget button. I created a void method that creates a contextMenu. I created a Boolean method that creates an options menu and return a true value. I created a Boolean method that returns the menu choice of the item selected. I created a void method that creates a menu with 7 items having the first four with alphabet shortcuts a-d. I created a Boolean method with a switch statement that checks on what was clicked and sets the toast message to display that the user clicked on an option with its number.

```
1 package com.example.menu;
2
3 import androidx.appcompat.app.AppCompatActivity;
4
5 import android.os.Bundle;
6 import android.view.ContextMenu;
7 import android.view.Menu;
8 import android.view.MenuItem;
9 import android.view.View;
10 import android.widget.Button;
11 import android.widget.Toast;
12
13 2 usages
14 public class MainActivity extends AppCompatActivity {
15
16     @Override
17     protected void onCreate(Bundle savedInstanceState) {
18         super.onCreate(savedInstanceState);
19         setContentView(R.layout.activity_main);
20
21         Button btn = (Button) findViewById(R.id.button);
22         btn.setOnCreateContextMenuListener(this);
23     }
24
25     @Override
26     public void onCreateContextMenu(ContextMenu menu, View view, ContextMenu.ContextMenuInfo menuInfo) {
27         super.onCreateContextMenu(menu, view, menuInfo);
28         createMenu(menu);
29     }
30 }
```



```
MainActivity.java x
28
29
30 @Override
31 public boolean onCreateOptionsMenu(Menu menu) {
32     super.onCreateOptionsMenu(menu);
33     createMenu(menu);
34     return true;
35 }
36
37 @Override
38 public boolean onOptionsItemSelected(MenuItem item) { return MenuChoice(item); }
39
40
41 2 usages
42 @ private void createMenu(Menu menu) {
43     MenuItem mnu1 = menu.add( groupId: 0, itemId: 0, order: 0, title: "Item 1"); {
44         mnu1.setAlphabeticShortcut('a');
45     }
46     MenuItem mnu2 = menu.add( groupId: 0, itemId: 1, order: 1, title: "Item 2"); {
47         mnu2.setAlphabeticShortcut('b');
48     }
49     MenuItem mnu3 = menu.add( groupId: 0, itemId: 2, order: 2, title: "Item 3"); {
50         mnu3.setAlphabeticShortcut('c');
51     }
52     MenuItem mnu4 = menu.add( groupId: 0, itemId: 3, order: 3, title: "Item 4"); {
53         mnu4.setAlphabeticShortcut('d');
54     }
55     menu.add( groupId: 0, itemId: 4, order: 4, title: "Item 5");
56     menu.add( groupId: 0, itemId: 5, order: 5, title: "Item 6");
57     menu.add( groupId: 0, itemId: 6, order: 6, title: "Item 7");
58 }
```

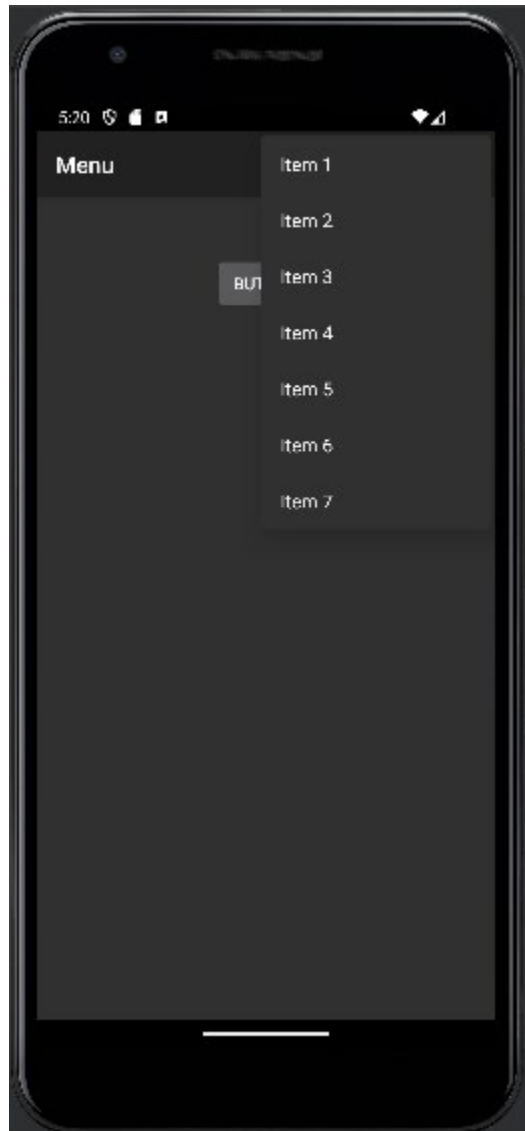
```
MainActivity.java x
57 }
58
59 1 usage
60 @ private boolean MenuChoice(MenuItem item) {
61     switch (item.getItemId()) {
62     case 0:
63         Toast.makeText( context: this, text: "You clicked on Item 1",
64             Toast.LENGTH_LONG).show();
65         return true;
66     case 1:
67         Toast.makeText( context: this, text: "You clicked on Item 2",
68             Toast.LENGTH_LONG).show();
69         return true;
70     case 2:
71         Toast.makeText( context: this, text: "You clicked on Item 3",
72             Toast.LENGTH_LONG).show();
73         return true;
74     case 3:
75         Toast.makeText( context: this, text: "You clicked on Item 4",
76             Toast.LENGTH_LONG).show();
77         return true;
78     case 4:
79         Toast.makeText( context: this, text: "You clicked on Item 5",
80             Toast.LENGTH_LONG).show();
81         return true;
82     case 5:
83         Toast.makeText( context: this, text: "You clicked on Item 6",
84             Toast.LENGTH_LONG).show();
85         return true;
86     }
87 }
```

```
84         case 6:
85             Toast.makeText(context: this, text: "You clicked on Item 7",
86                 Toast.LENGTH_LONG).show();
87             return true;
88         }
89         return false;
90     }
91 }
92
```

When the app is run, the following is the screen:



When the three dots at the top right of the screen are tapped, the following is displayed:



When I tap on Item1, the following toast message is displayed:



When I tap on Item2, the following toast message is displayed:



When I tap on Item3, the following toast message is displayed:



When I tap on Item4, the following toast message is displayed:





When I tap on Item5, the following toast message is displayed:



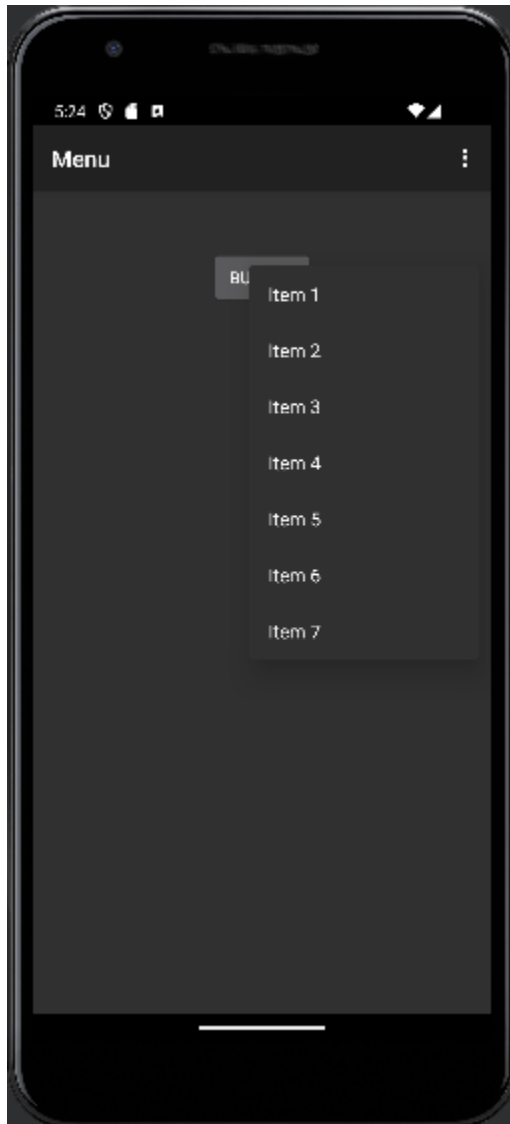
When I tap on Item6, the following toast message is displayed:



When I tap on Item7, the following toast message is displayed:



When I tap and hold the button at the center of the screen, I see the following:

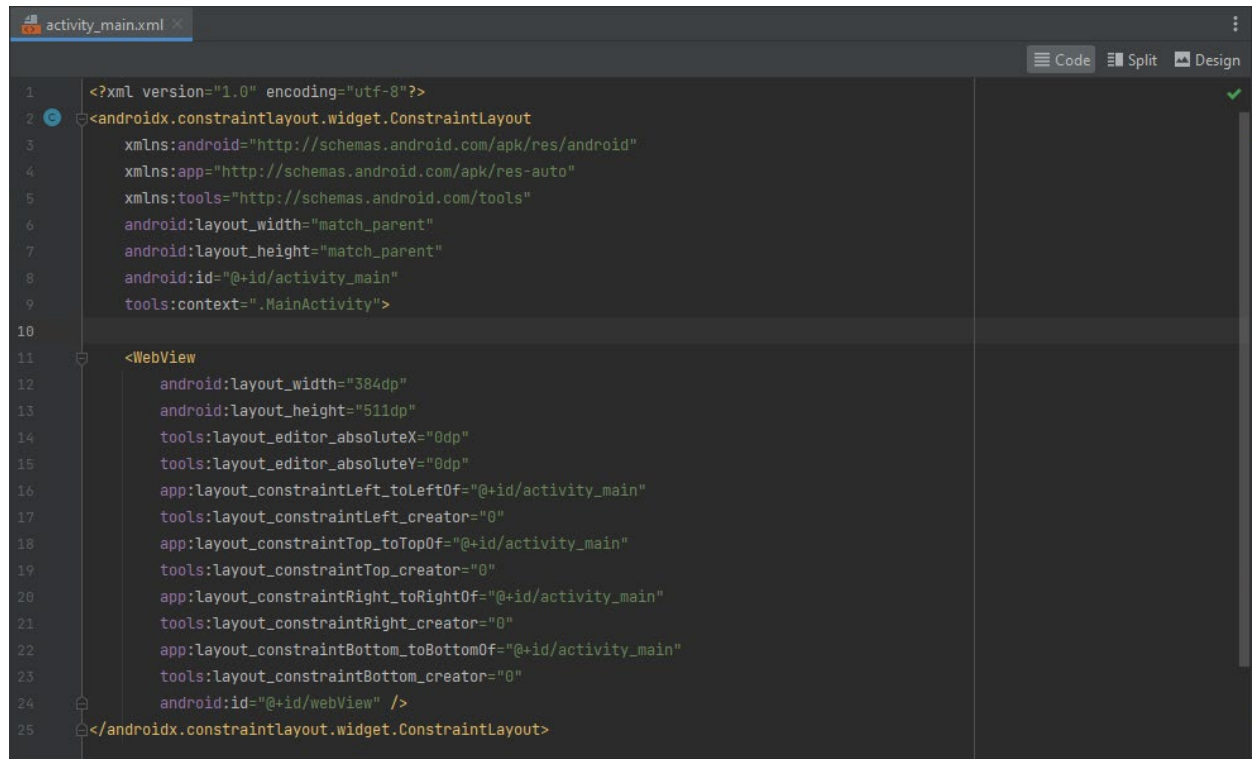


This shows that the objective of the project has been achieved.

## WebView

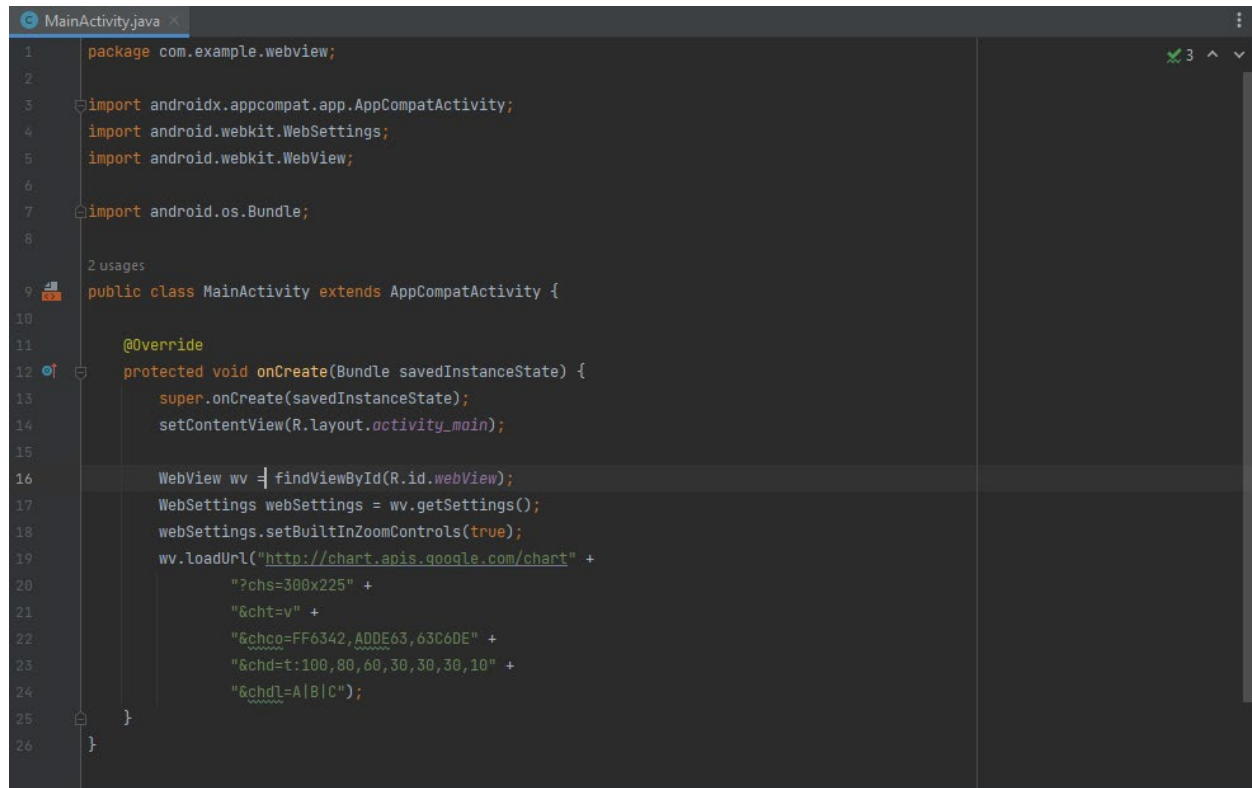
The objective of this project is to create a webview that shows the content of a specified url.

In the activity\_main.xml file, I created a constraint layout and added a WebView widget with an id of webView as shown below.



```
1 <?xml version="1.0" encoding="utf-8"?>
2 <androidx.constraintlayout.widget.ConstraintLayout
3     xmlns:android="http://schemas.android.com/apk/res/android"
4     xmlns:app="http://schemas.android.com/apk/res-auto"
5     xmlns:tools="http://schemas.android.com/tools"
6     android:layout_width="match_parent"
7     android:layout_height="match_parent"
8     android:id="@+id/activity_main"
9     tools:context=".MainActivity">
10
11     <WebView
12         android:layout_width="384dp"
13         android:layout_height="511dp"
14         tools:layout_editor_absoluteX="0dp"
15         tools:layout_editor_absoluteY="0dp"
16         app:layout_constraintLeft_toLeftOf="@+id/activity_main"
17         tools:layout_constraintLeft_creator="0"
18         app:layout_constraintTop_toTopOf="@+id/activity_main"
19         tools:layout_constraintTop_creator="0"
20         app:layout_constraintRight_toRightOf="@+id/activity_main"
21         tools:layout_constraintRight_creator="0"
22         app:layout_constraintBottom_toBottomOf="@+id/activity_main"
23         tools:layout_constraintBottom_creator="0"
24         android:id="@+id/webView" />
25 </androidx.constraintlayout.widget.ConstraintLayout>
```

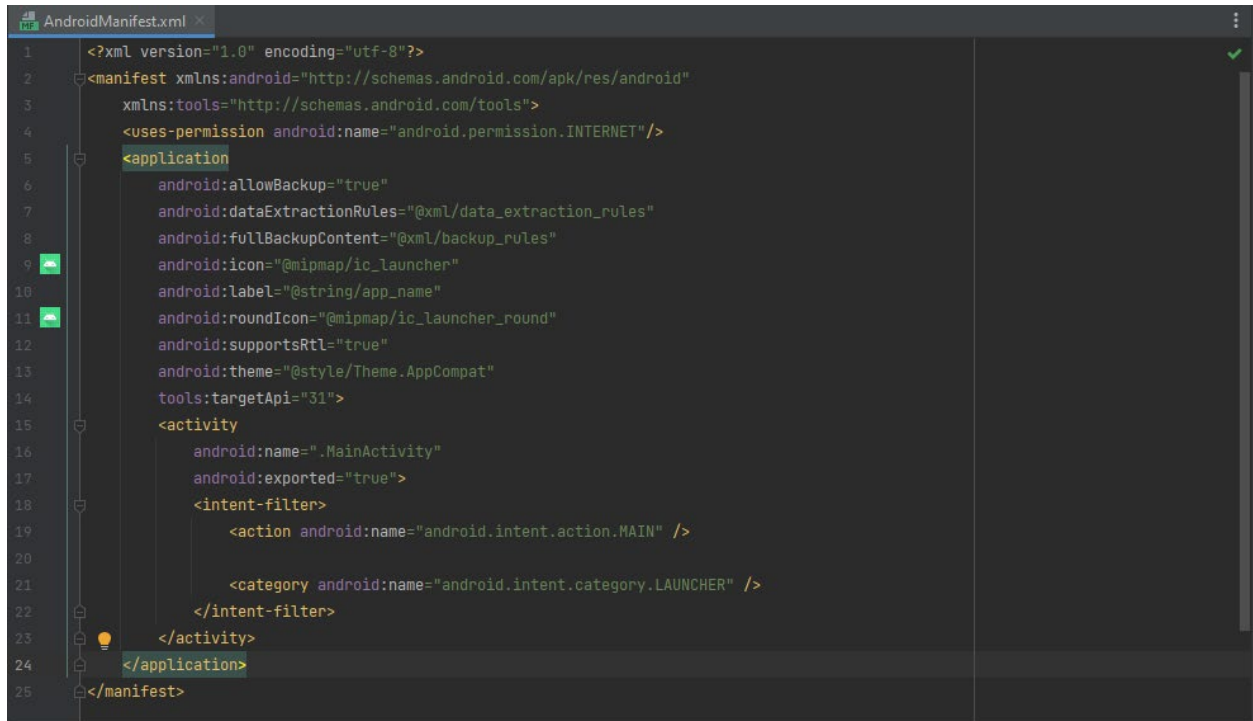
In the MainActivity.java file, I created a variable wv of type WebView and set it to be the webView widget I created earlier in activity\_main.xml file. I created variable websettings of type WebSettings and set it to be the getSettings method. I set the websettings built in zoom controls to be of value true. I specified the webview url as shown below.

A screenshot of an IDE showing the MainActivity.java file. The code is as follows:

```
1 package com.example.webview;
2
3 import androidx.appcompat.app.AppCompatActivity;
4 import android.webkit.WebSettings;
5 import android.webkit.WebView;
6
7 import android.os.Bundle;
8
9 public class MainActivity extends AppCompatActivity {
10
11     @Override
12     protected void onCreate(Bundle savedInstanceState) {
13         super.onCreate(savedInstanceState);
14         setContentView(R.layout.activity_main);
15
16         WebView wv = findViewById(R.id.webView);
17         WebSettings webSettings = wv.getSettings();
18         webSettings.setBuiltInZoomControls(true);
19         wv.loadUrl("http://chart.apis.google.com/chart" +
20                 "?chs=300x225" +
21                 "&cht=v" +
22                 "&chcq=FF6342,A0DE63,63C6DE" +
23                 "&chd=t:100,80,60,30,30,30,10" +
24                 "&chdl=A|B|C");
25     }
26 }
```

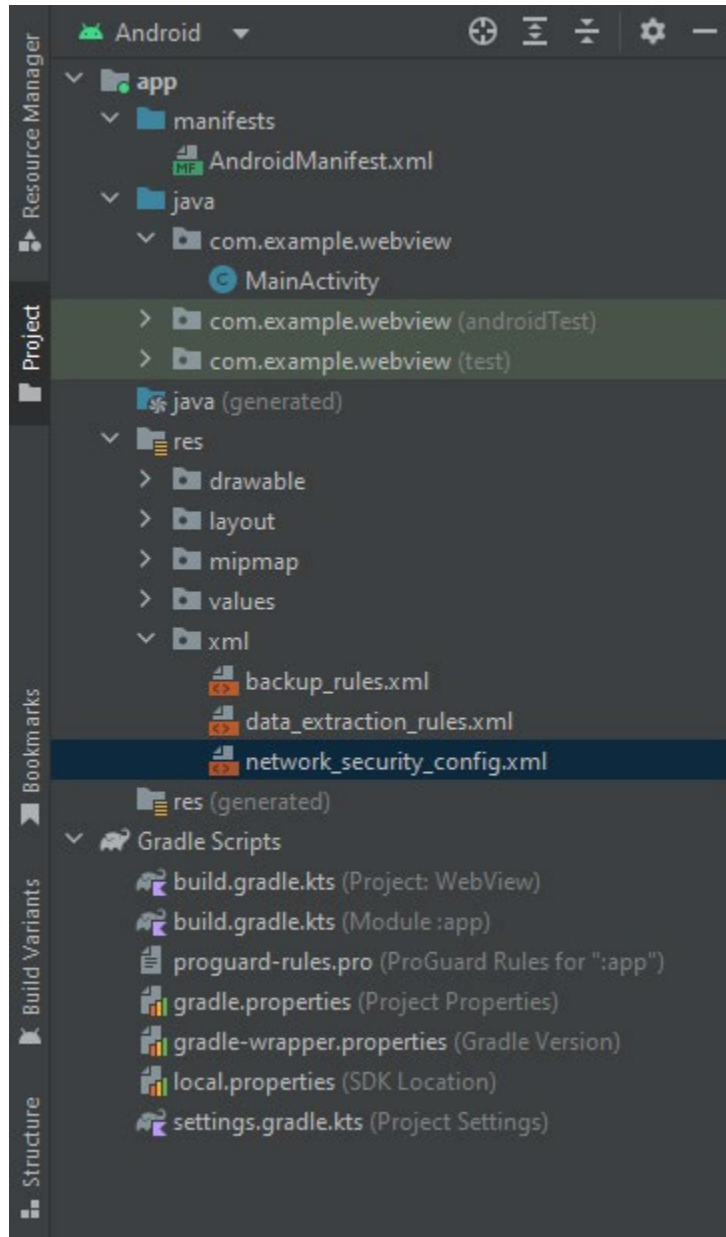
The IDE interface includes a tab at the top labeled 'MainActivity.java', a line number margin on the left, and a gutter with icons for breakpoints and a search icon. The code is color-coded with syntax highlighting.

I gave permission for the project to use the internet in the AndroidManifest.xml file as shown below on line 4.



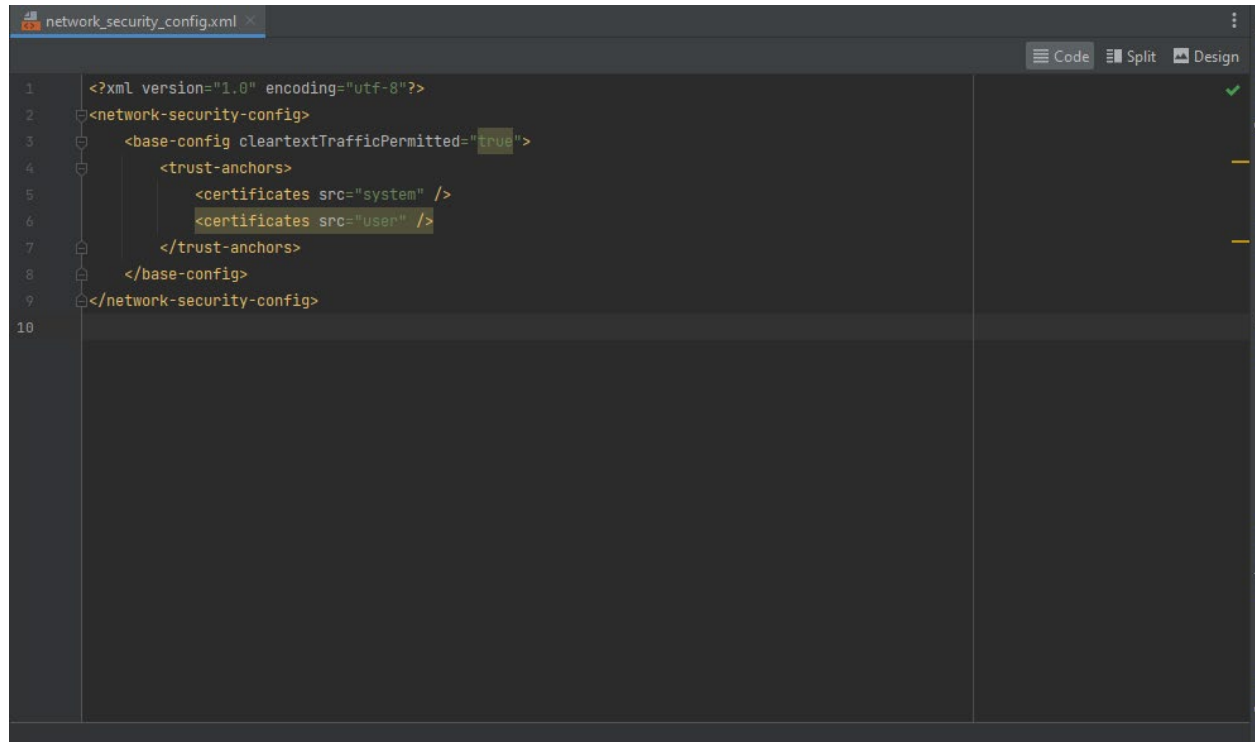
```
1  <?xml version="1.0" encoding="utf-8"?>
2  <manifest xmlns:android="http://schemas.android.com/apk/res/android"
3  xmlns:tools="http://schemas.android.com/tools">
4  <uses-permission android:name="android.permission.INTERNET"/>
5  <application
6      android:allowBackup="true"
7      android:dataExtractionRules="@xml/data_extraction_rules"
8      android:fullBackupContent="@xml/backup_rules"
9      android:icon="@mipmap/ic_launcher"
10     android:label="@string/app_name"
11     android:roundIcon="@mipmap/ic_launcher_round"
12     android:supportsRtl="true"
13     android:theme="@style/Theme.AppCompat"
14     tools:targetApi="31">
15     <activity
16         android:name=".MainActivity"
17         android:exported="true">
18         <intent-filter>
19             <action android:name="android.intent.action.MAIN" />
20
21             <category android:name="android.intent.category.LAUNCHER" />
22         </intent-filter>
23     </activity>
24 </application>
25 </manifest>
```

I created a `network_security_config.xml` file in the `xml` folder as shown below.



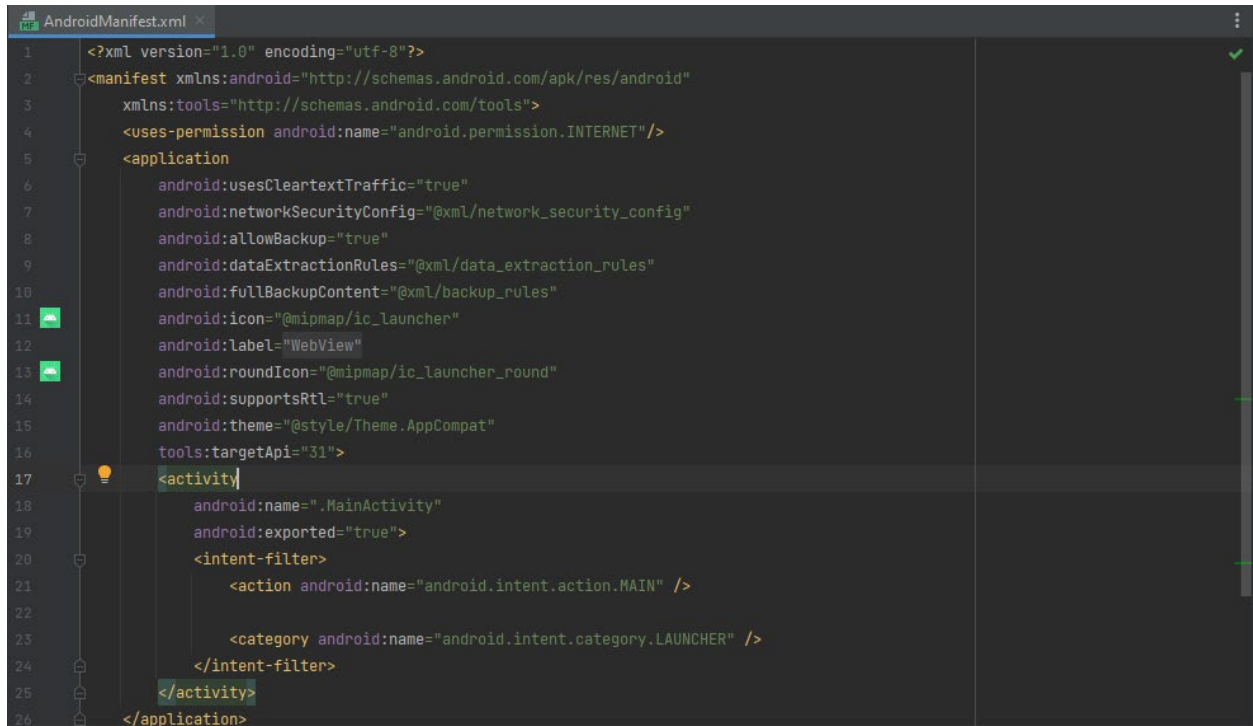


I configured it to allow access of the application to the internet as shown below.



```
1  <?xml version="1.0" encoding="utf-8"?>
2  <network-security-config>
3    <base-config cleartextTrafficPermitted="true">
4      <trust-anchors>
5        <certificates src="system" />
6        <certificates src="user" />
7      </trust-anchors>
8    </base-config>
9  </network-security-config>
```

Back in AndroidManifest.xml file, I set the usesCleartextTraffic value to be true and set the networkSecurityConfig value to look for the file I created previously in line 6 and 7 respectively. This is shown below.



```
1  <?xml version="1.0" encoding="utf-8"?>
2  <manifest xmlns:android="http://schemas.android.com/apk/res/android"
3  xmlns:tools="http://schemas.android.com/tools">
4  <uses-permission android:name="android.permission.INTERNET"/>
5  <application
6  android:usesCleartextTraffic="true"
7  android:networkSecurityConfig="@xml/network_security_config"
8  android:allowBackup="true"
9  android:dataExtractionRules="@xml/data_extraction_rules"
10 android:fullBackupContent="@xml/backup_rules"
11 android:icon="@mipmap/ic_launcher"
12 android:label="@string/app_name"
13 android:roundIcon="@mipmap/ic_launcher_round"
14 android:supportRtl="true"
15 android:theme="@style/Theme.AppCompat"
16 tools:targetApi="31">
17 <activity
18     android:name=".MainActivity"
19     android:exported="true">
20     <intent-filter>
21         <action android:name="android.intent.action.MAIN" />
22         <category android:name="android.intent.category.LAUNCHER" />
23     </intent-filter>
24 </activity>
25 </application>
26 </manifest>
```

This will allow the application to correctly run.

When I run the application, this is what I see.



When I drag the screen in the webView section, I see the zoom built in controls as show below:



When I tap the zoom in icon twice, it zooms in twice as show below:



When I tap the zoom out button twice, it zooms out as shown below:



This marks the completion of the project's objectives.