

Week 7: Develop an application that makes use of RSS Feed

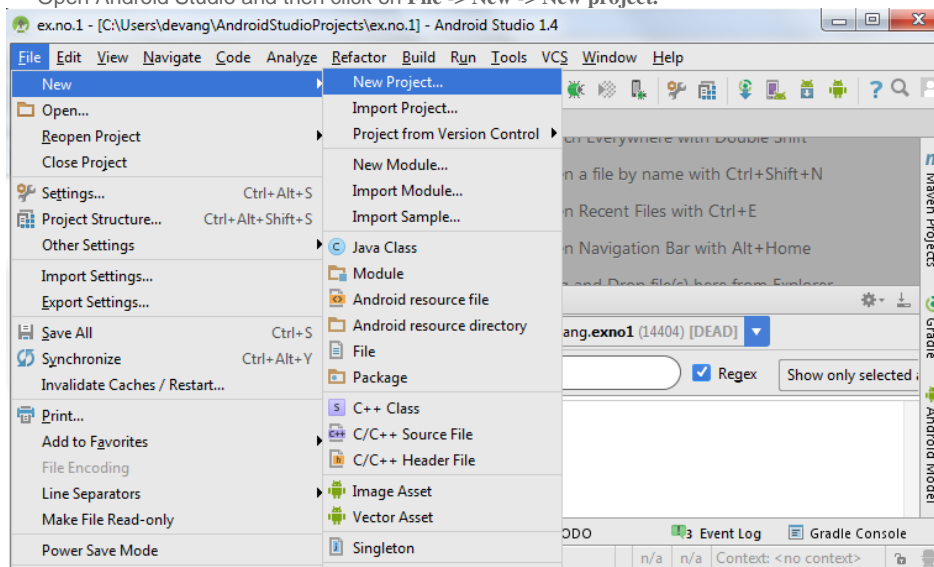
Aim:

To develop a Android Application that makes use of RSS Feed.

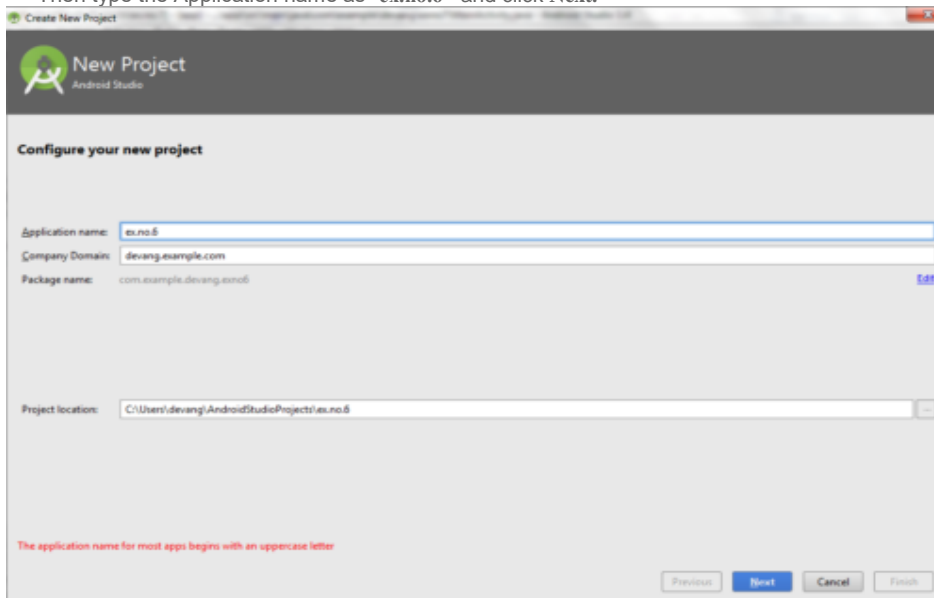
Procedure:

Creating a New project:

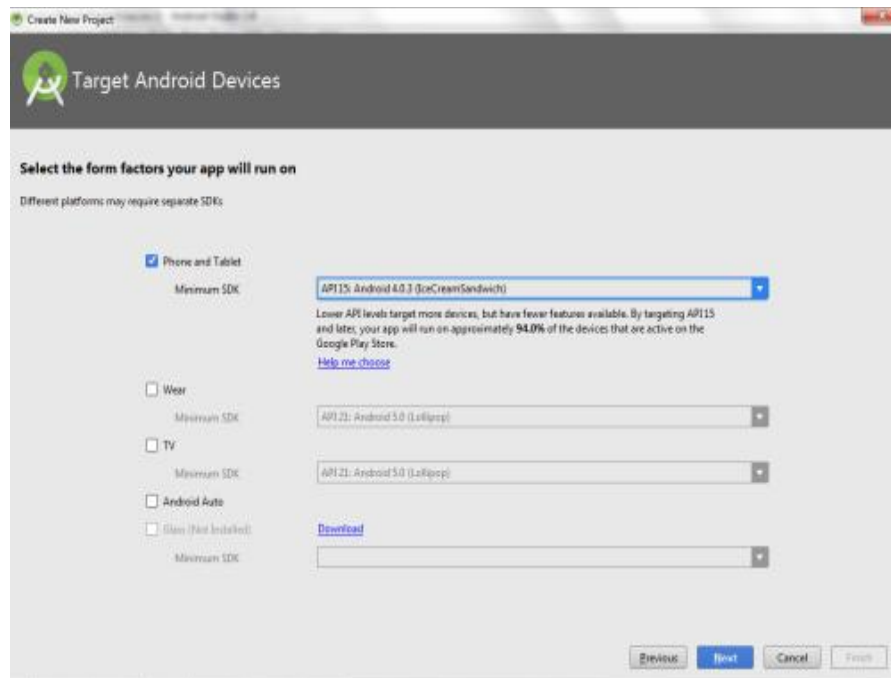
- Open Android Studio and then click on File -> New -> New project.



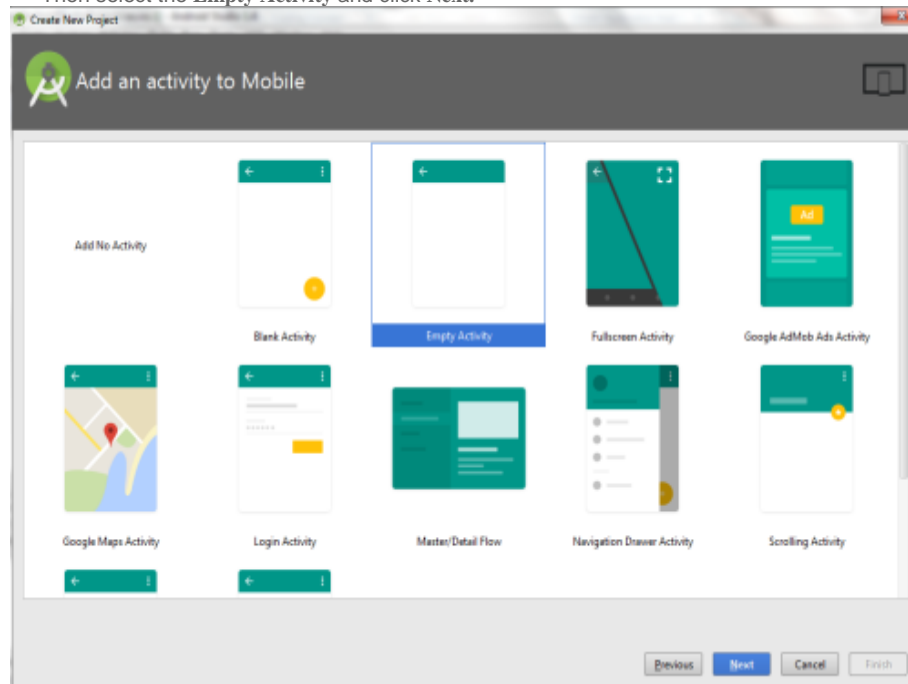
- Then type the Application name as “ex.no.6” and click Next.



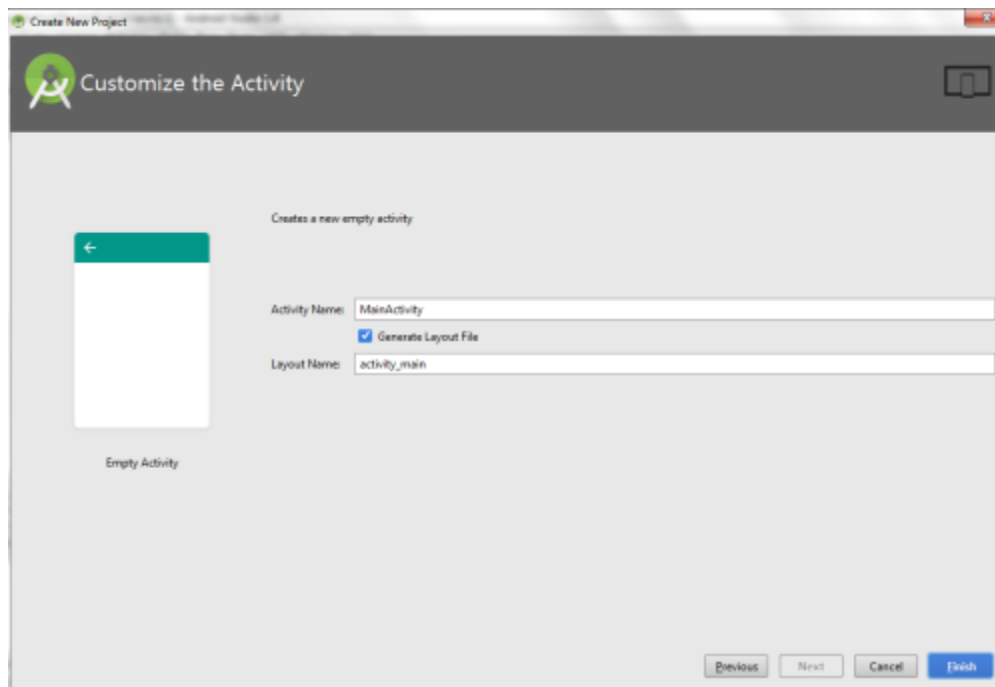
- Then select the Minimum SDK as shown below and click Next.



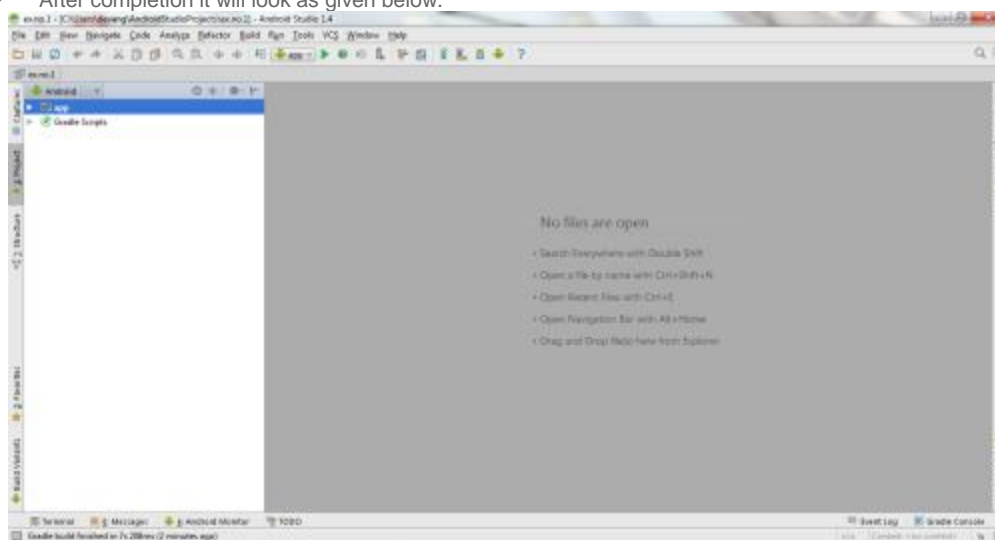
- Then select the **Empty Activity** and click Next.



- Finally click **Finish**.

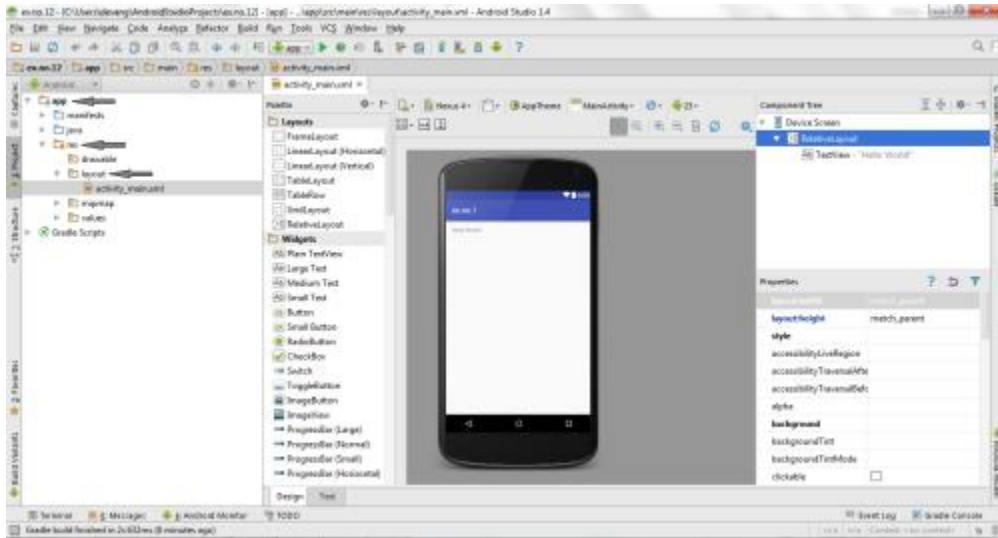


- It will take some time to build and load the project.
- After completion it will look as given below.

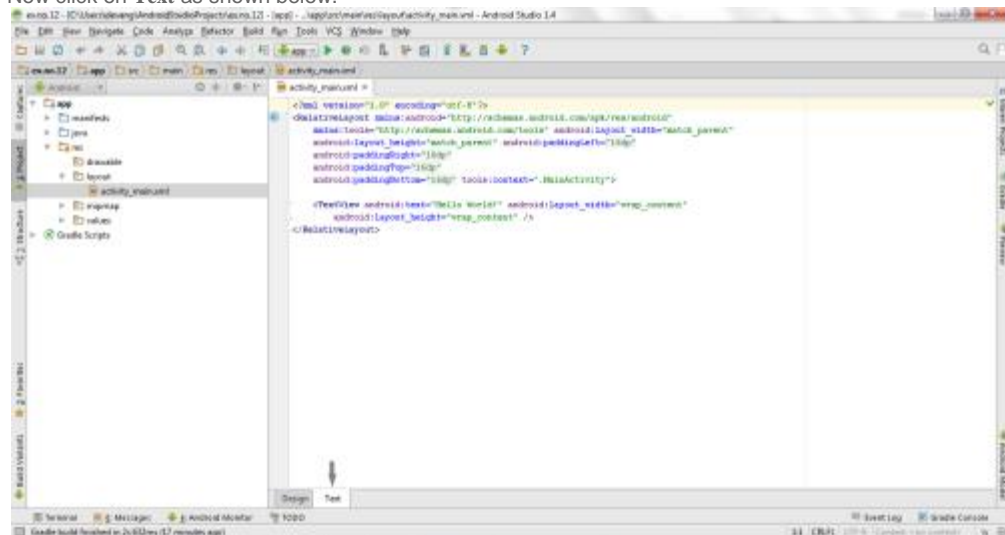


Designing layout for the Android Application:

- Click on app -> res -> layout -> activity_main.xml



Now click on Text as shown below.



Then delete the code which is there and type the code as given below.

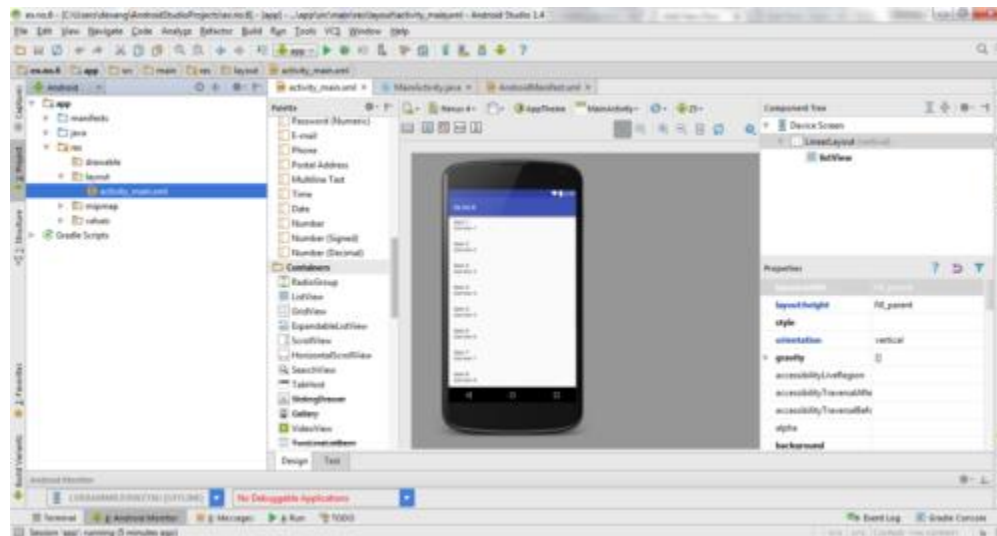
Code for Activity_main.xml:

```

2
1 <?xml version="1.0" encoding="utf-8"?>
2 <LinearLayout
3   xmlns:android="http://schemas.android.com/apk/res/android"
4     android:layout_width="fill_parent"
5     android:layout_height="fill_parent"
6     android:orientation="vertical" >
7
8     <ListView
9       android:id="@+id/listView"
10      android:layout_width="match_parent"
11      android:layout_height="wrap_content" />
12 </LinearLayout>

```

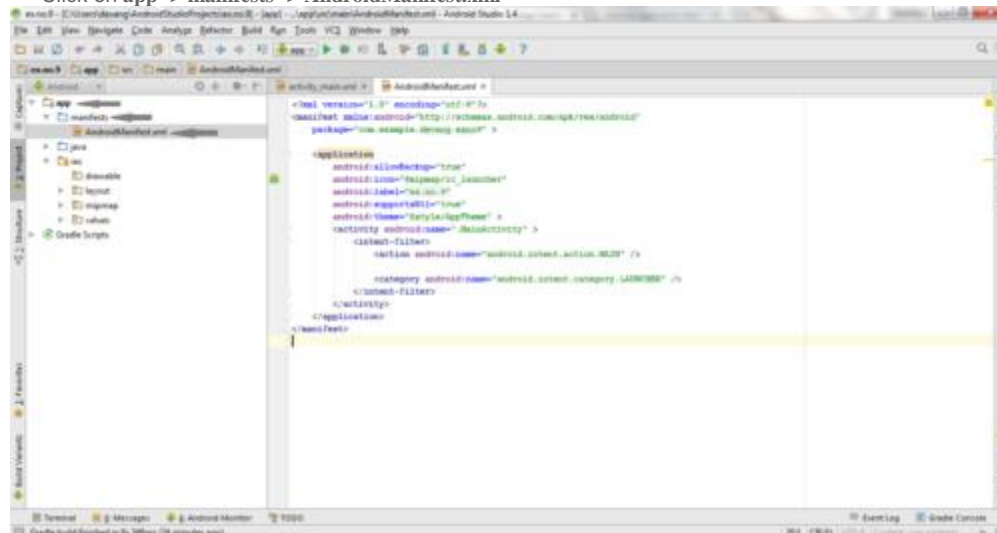
Now click on Design and your application will look as given below.



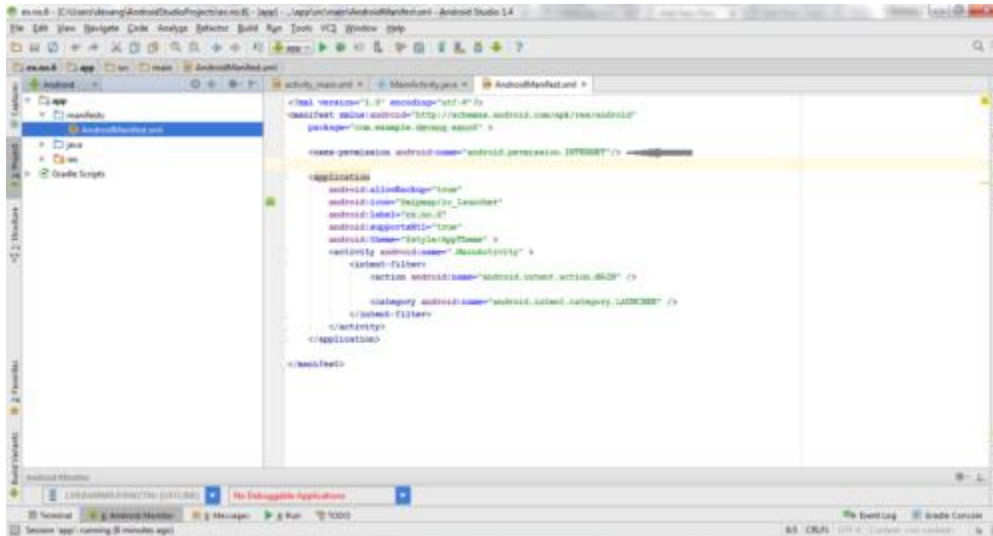
- So now the designing part is completed.

Adding permissions in Manifest for the Android Application:

- Click on app -> manifests -> AndroidManifest.xml



- Now include the **INTERNET** permissions in the AndroidManifest.xml file as shown below



Code for AndroidManifest.xml:

```

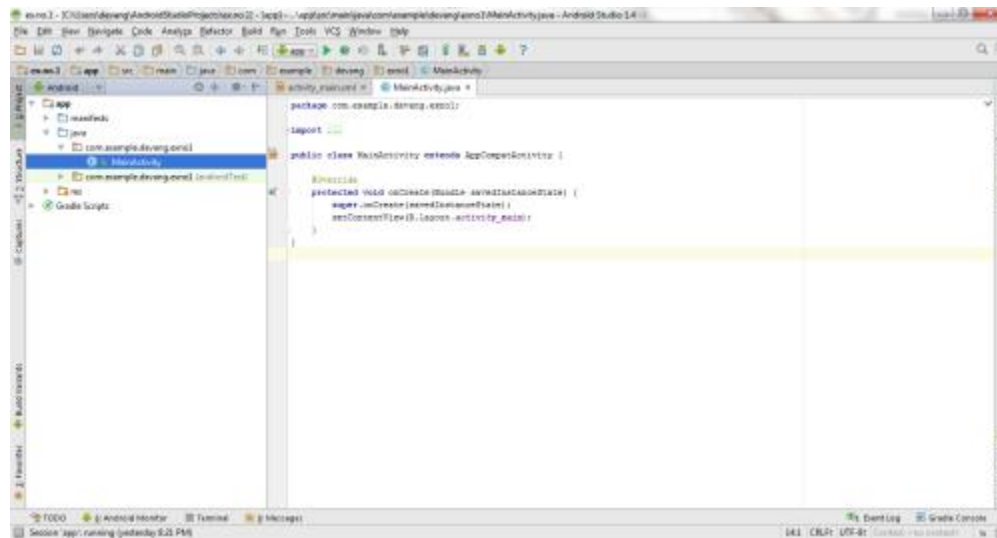
1  <?xml version="1.0" encoding="utf-8"?>
2  <manifest
3      xmlns:android="http://schemas.android.com/apk/res/android"
4      package="com.example.exno6" >
5
6      <uses-permission
7          android:name="android.permission.INTERNET"/>
8
9      <application
10         android:allowBackup="true"
11         android:icon="@mipmap/ic_launcher"
12         android:label="@string/app_name"
13         android:supportsRtl="true"
14         android:theme="@style/AppTheme" >
15         <activity android:name=".MainActivity" >
16             <intent-filter>
17                 <action
18                     android:name="android.intent.action.MAIN" />
19
20                 <category
21                     android:name="android.intent.category.LAUNCHER" />
22             </intent-filter>
23         </activity>
24     </application>
25 </manifest>

```

- So now the Permissions are added in the Manifest.

Java Coding for the Android Application:

- Click on **app** -> **java** -> **com.example.exno6** -> **MainActivity**.



- Then delete the code which is there and type the code as given below.

Code for MainActivity.java:

```
package com.example.week7;

import android.app.ListActivity;
import android.content.Intent;
import android.net.Uri;
import android.os.AsyncTask;
import android.os.Bundle;
import android.view.View;
import android.widget.AdapterView;
import android.widget.AdapterView.OnItemClickListener;
import android.widget.ListView;
import org.xmlpull.v1.XmlPullParser;
import org.xmlpull.v1.XmlPullParserException;
import org.xmlpull.v1.XmlPullParserFactory;
import java.io.IOException;
import java.io.InputStream;
import java.net.MalformedURLException;
import java.net.URL;
import java.util.ArrayList;
import java.util.List;

public class MainActivity extends ListActivity
{
    List headlines;
    List links;

    @Override
    protected void onCreate(Bundle savedInstanceState)
    {
        super.onCreate(savedInstanceState);
        new MyAsyncTask().execute();
    }

    class MyAsyncTask extends AsyncTask<Object, Void, ArrayAdapter>
    {
        @Override
        protected ArrayAdapter doInBackground(Object[] params)
        {

```

```

headlines = new ArrayList();
links = new ArrayList();
try
{
    URL url = new URL("https://codingconnect.net/feed");
    XmlPullParserFactory factory =
XmlPullParserFactory.newInstance();
    factory.setNamespaceAware(false);
    XmlPullParser xpp = factory.newPullParser();

    // We will get the XML from an input stream
    xpp.setInput(getInputStream(url), "UTF_8");
    boolean insideItem = false;

    // Returns the type of current event: START_TAG, END_TAG,
etc..

    int eventType = xpp.getEventType();
    while (eventType != XmlPullParser.END_DOCUMENT)
    {
        if (eventType == XmlPullParser.START_TAG)
        {
            if (xpp.getName().equalsIgnoreCase("item"))
            {
                insideItem = true;
            }
            else if (xpp.getName().equalsIgnoreCase("title"))
            {
                if (insideItem)
                    headlines.add(xpp.nextText()); //extract the
headline

            }
            else if (xpp.getName().equalsIgnoreCase("link"))
            {
                if (insideItem)
                    links.add(xpp.nextText()); //extract the link
of article

            }
        }
        else if (eventType==XmlPullParser.END_TAG &&
xpp.getName().equalsIgnoreCase("item"))
        {
            insideItem=false;
        }
        eventType = xpp.next(); //move to next element
    }

}
catch (MalformedURLException e)
{
    e.printStackTrace();
}
catch (XmlPullParserException e)
{
    e.printStackTrace();
}
catch (IOException e)
{

```



```

        e.printStackTrace();
    }
    return null;
}
protected void onPostExecute(ArrayAdapter adapter)
{
    adapter = new ArrayAdapter(MainActivity.this,
android.R.layout.simple_list_item_1, headlines);
    setListAdapter(adapter);
}

}

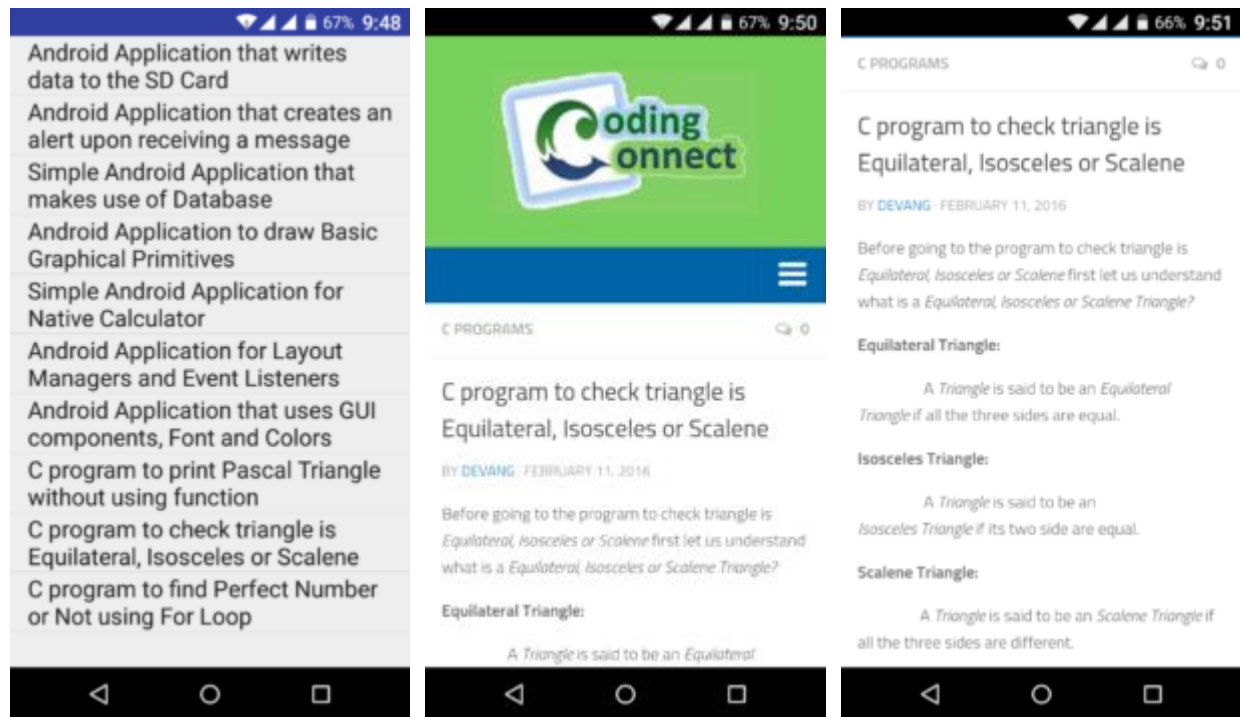
@Override
protected void onItemClick(ListView l, View v, int position, long id)
{
    Uri uri = Uri.parse((links.get(position)).toString());
    Intent intent = new Intent(Intent.ACTION_VIEW, uri);
    startActivity(intent);
}

public InputStream getInputStream(URL url)
{
    try
    {
        return url.openConnection().getInputStream();
    }
    catch (IOException e)
    {
        return null;
    }
}
}

```

- 2
- So now the Coding part is also completed.
- Now run the application to see the output.

Output:



Result:

Thus Android Application that makes use of RSS Feed is developed and executed successfully.