**Dipen Delvadiya**

**ITMD – 455**

**Lab – 4**

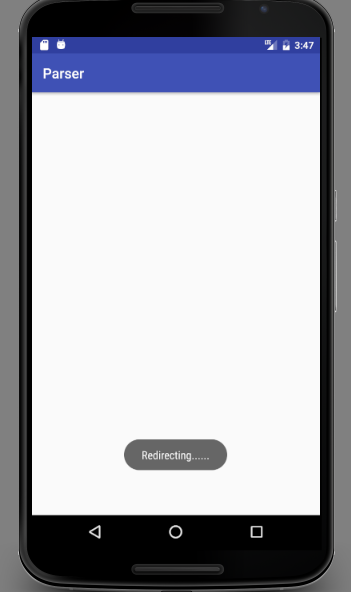
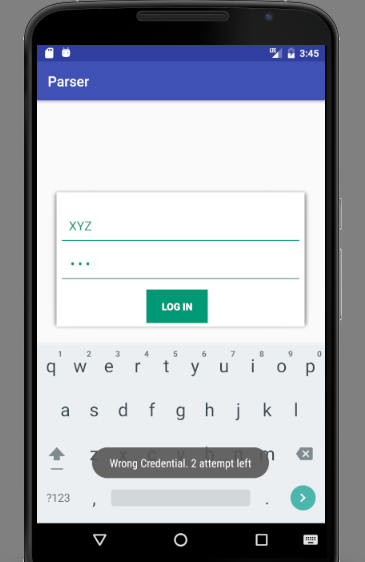
**James Papademas**

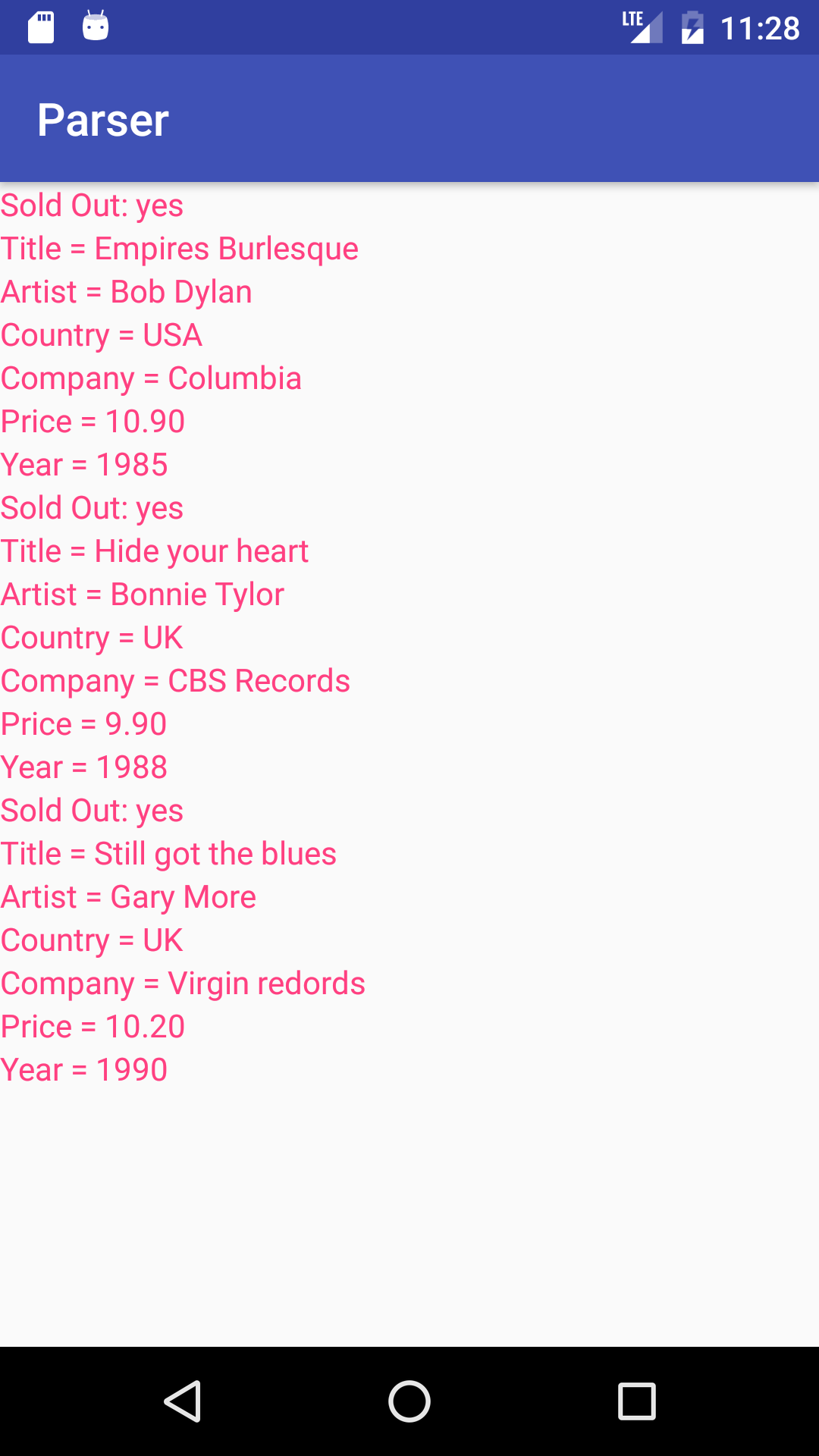
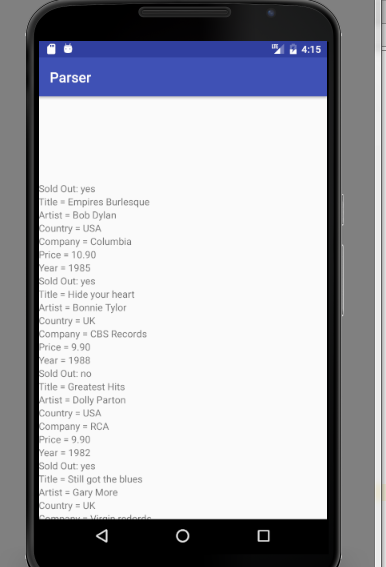
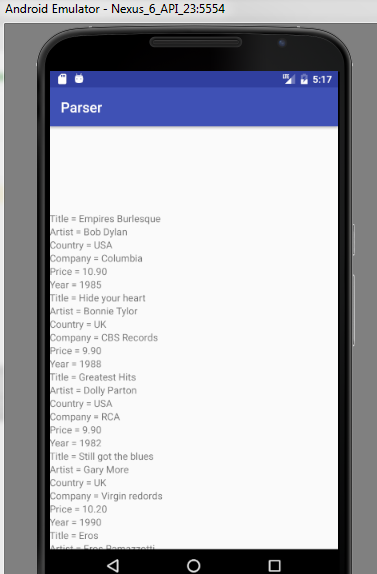
**Snapshots:**

**Note: username:Mary**

**Password: 123**

1. **Wrong Credentials 2. Successful LogIn**

****

****

**Source Code:**

**AndroidManifest.xml:**

*<?***xml version="1.0" encoding="utf-8"***?>*<**manifest xmlns:android="http://schemas.android.com/apk/res/android"  
 package="com.example.parser"**>  
  
 <**uses-permission android:name="android.permission.INTERNET"**></**uses-permission**>  
 <**application  
 android:allowBackup="true"  
 android:icon="@mipmap/ic\_launcher"  
 android:label="@string/app\_name"  
 android:supportsRtl="true"  
 android:theme="@style/AppTheme"**>  
 <**activity android:name=".Login"**>  
 <**intent-filter**>  
 <**action android:name="android.intent.action.MAIN"** />  
  
 <**category android:name="android.intent.category.LAUNCHER"** />  
 </**intent-filter**>  
 </**activity**>  
 <**activity android:name=".MainActivity"**></**activity**>  
 </**application**>  
  
</**manifest**>

**Login.java**

**package** com.example.parser;  
  
**import** android.content.Context;  
**import** android.content.Intent;  
**import** android.support.v7.app.AppCompatActivity;  
**import** android.os.Bundle;  
**import** android.view.View;  
**import** android.widget.Button;  
**import** android.widget.EditText;  
**import** android.widget.Toast;  
  
**public class** Login **extends** AppCompatActivity {  
  
 **int attmpt**=3;  
 @Override  
 **protected void** onCreate(Bundle savedInstanceState) {  
 **super**.onCreate(savedInstanceState);  
 setContentView(R.layout.***activity\_login***);  
 **final** Context context = getApplicationContext();  
 **final** String name=**"Mary"**;  
 **final** String passwd=**"123"**;  
 **final** EditText usrnm,pass;  
 Button btn;  
 usrnm=(EditText)findViewById(R.id.***loginusrnm***);  
 pass=(EditText)findViewById(R.id.***loginpasswd***);  
 btn=(Button)findViewById(R.id.***btnLogin***);  
  
  
 btn.setOnClickListener(**new** View.OnClickListener() {  
 @Override  
 **public void** onClick(View v) {  
 **if**(usrnm.getText().toString().equals(name) && pass.getText().toString().equals(passwd))  
 {  
 Toast.*makeText*(context,**"Redirecting......"**, Toast.***LENGTH\_LONG***).show();  
 Intent i=**new** Intent(Login.**this**,MainActivity.**class**);  
 startActivity(i);  
 }  
 **else** {  
 **attmpt**--;  
 **if**(**attmpt**<0)  
 {  
 Toast.*makeText*(context,**"No Attempt is left"**, Toast.***LENGTH\_LONG***).show();  
 }  
 **else** Toast.*makeText*(context,**"Wrong Credential. "**+**attmpt**+**" attempt left"**, Toast.***LENGTH\_LONG***).show();  
 }  
 }  
 });  
 }  
}

**MainActivity.java:**

**package** com.example.parser;  
  
**import** android.app.ProgressDialog;  
**import** android.os.AsyncTask;  
**import** android.support.v7.app.AppCompatActivity;  
**import** android.os.Bundle;  
**import** android.view.View;  
**import** android.view.ViewGroup;  
**import** android.widget.TextView;  
**import** android.widget.Toast;  
  
**import** org.xml.sax.InputSource;  
**import** org.xml.sax.XMLReader;  
**import** java.net.URL;  
**import** javax.xml.parsers.SAXParser;  
**import** javax.xml.parsers.SAXParserFactory;  
  
  
**public class** MainActivity **extends** AppCompatActivity {  
  
XMLGettersSetters **data**;  
ProgressDialog **waitProgress**;  
  
*/\*\*  
 \* Called when the activity is first created.  
 \*/*@Override  
**public void** onCreate(Bundle savedInstanceState) {  
 **super**.onCreate(savedInstanceState);  
 setContentView(R.layout.***activity\_main***);  
 **new** BackgroundTask().execute();  
}  
  
**public class** BackgroundTask **extends** AsyncTask<Void, Integer, Void> {  
  
 @Override  
 **protected void** onPreExecute() {  
 **super**.onPreExecute();  
 }  
  
 @Override  
 **protected void** onProgressUpdate(Integer... values) {  
 **super**.onProgressUpdate(values);  
 **if** (**waitProgress** != **null**) {  
 **waitProgress**.dismiss();  
 }  
 }  
  
 @Override  
 **protected void** onPostExecute(Void aVoid) {  
 **super**.onPostExecute(aVoid);  
  
 **if** (**waitProgress** != **null**) {  
 **waitProgress**.dismiss();  
 }  
 }  
  
 @Override  
 **protected** Void doInBackground(Void... params) {  
  
 **try** {  
 **synchronized** (**this**) {  
 saxParser();  
 }  
 } **catch** (Exception e) {  
 e.printStackTrace();  
 }  
  
 **return null**;  
 }  
}  
  
**private void** saxParser() {  
 */\*\*  
 \* Get the view of the layout within the main layout, so that we can  
 add TextViews.  
 \*\*/* View layout = findViewById(R.id.***layout***);  
  
 */\*\*  
 \* Create TextView Arrays to add the retrieved data to.  
 \*\*/* TextView sold[];  
 TextView title[];  
  
 TextView artist[];  
  
 TextView country[];  
  
 TextView company[];  
  
 TextView price[];  
  
 TextView year[];  
  
 **try** {  
  
  
 */\*\*  
 \* Create a new instance of the SAX parser  
 \*\*/* SAXParserFactory saxPF = SAXParserFactory.*newInstance*();  
 SAXParser saxP = saxPF.newSAXParser();  
 XMLReader xmlR = saxP.getXMLReader();  
  
 *// URL of the XML* URL url = **new** URL(**"http://www.papademas.net/cd\_catalog3.xml"**);  
 */\*\*  
 \* Create the Handler to handle each of the XML tags.  
 \*\*/* XMLHandler myXMLHandler = **new** XMLHandler();  
 xmlR.setContentHandler(myXMLHandler);  
 xmlR.parse(**new** InputSource(url.openStream()));  
  
 } **catch** (Exception e) {  
 System.***out***.println(e);  
 }  
  
 **data** = XMLHandler.*data*;  
  
 */\*\*  
 \* Makes the TextView length the size of the TextView arrays by  
 getting the size of the  
 \*\*/* sold=**new** TextView[**data**.getAttribute().size()];  
 title = **new** TextView[**data**.getTitle().size()];  
 artist = **new** TextView[**data**.getArtist().size()];  
 country = **new** TextView[**data**.getCountry().size()];  
 company = **new** TextView[**data**.getCompany().size()];  
 price = **new** TextView[**data**.getPrice().size()];  
 year = **new** TextView[**data**.getYear().size()];  
  
 */\*\*  
 \* Run a for loop to set All the TextViews with text until  
 \* the size of the array is reached.  
 \*  
 \*\*/* **for** (**int** i = 0; i < **data**.getTitle().size(); i++) {  
  
 **if**(**data**.getAttribute().get(i).equals(**"yes"**)) {  
 sold[i] = **new** TextView(**this**);  
 sold[i].setText(**"Sold Out: "** + **data**.getAttribute().get(i));  
  
 title[i] = **new** TextView(**this**);  
 title[i].setText(**"Title = "** + **data**.getTitle().get(i));  
  
 artist[i] = **new** TextView(**this**);  
 artist[i].setText(**"Artist = "** + **data**.getArtist().get(i));  
  
 country[i] = **new** TextView(**this**);  
 country[i].setText(**"Country = "** + **data**.getCountry().get(i));  
  
 company[i] = **new** TextView(**this**);  
 company[i].setText(**"Company = "** + **data**.getCompany().get(i));  
  
 price[i] = **new** TextView(**this**);  
 price[i].setText(**"Price = "** + **data**.getPrice().get(i));  
  
 year[i] = **new** TextView(**this**);  
 year[i].setText(**"Year = "** + **data**.getYear().get(i));  
 ((ViewGroup) layout).addView(sold[i]);  
 ((ViewGroup) layout).addView(title[i]);  
 ((ViewGroup) layout).addView(artist[i]);  
 ((ViewGroup) layout).addView(country[i]);  
 ((ViewGroup) layout).addView(company[i]);  
 ((ViewGroup) layout).addView(price[i]);  
 ((ViewGroup) layout).addView(year[i]);  
 }  
 }  
  
 setContentView(layout);  
  
}  
}

**XMLGettersSetters.java:**

**package** com.example.parser;  
  
**import** java.util.ArrayList;  
**import** android.util.Log;  
  
*/\*\*  
 \* This class contains all getter and setter methods to set and retrieve data.  
 \*  
 \*\*/***public class** XMLGettersSetters {  
  
 **private** ArrayList<String> **title** = **new** ArrayList<String>();  
 **private** ArrayList<String> **artist** = **new** ArrayList<String>();  
 **private** ArrayList<String> **country** = **new** ArrayList<String>();  
 **private** ArrayList<String> **company** = **new** ArrayList<String>();  
 **private** ArrayList<String> **price** = **new** ArrayList<String>();  
 **private** ArrayList<String> **year** = **new** ArrayList<String>();  
 **private** ArrayList<String> **CD**=**new** ArrayList<String>();  
  
  
 **public** ArrayList<String> getCompany() {  
 **return company**;  
 }  
  
 **public void** setCompany(String company) {  
 **this**.**company**.add(company);  
 Log.*i*(**"This is the company:"**, company);  
 }  
  
 **public** ArrayList<String> getPrice() {  
 **return price**;  
 }  
  
 **public void** setPrice(String price) {  
 **this**.**price**.add(price);  
 Log.*i*(**"This is the price:"**, price);  
 }  
  
 **public** ArrayList<String> getYear() {  
 **return year**;  
 }  
  
 **public void** setYear(String year) {  
 **this**.**year**.add(year);  
 Log.*i*(**"This is the year:"**, year);  
 }  
  
 **public** ArrayList<String> getTitle() {  
 **return title**;  
 }  
  
 **public void** setTitle(String title) {  
 **this**.**title**.add(title);  
 Log.*i*(**"This is the title:"**, title);  
 }  
  
 **public** ArrayList<String> getArtist() {  
 **return artist**;  
 }  
  
 **public void** setArtist(String artist) {  
 **this**.**artist**.add(artist);  
 Log.*i*(**"This is the artist:"**, artist);  
 }  
  
 **public** ArrayList<String> getCountry() {  
 **return country**;  
 }  
  
 **public void** setCountry(String country) {  
 **this**.**country**.add(country);  
 Log.*i*(**"This is the country:"**, country);  
 }  
  
 **public** ArrayList<String> getAttribute()  
 {  
 **return CD**;  
 }  
 **public void** setAttribute(String attr) {  
 **this**.**CD**.add(attr);  
 Log.*i*(**"Sold out?: "**, attr);  
 }  
  
  
}

**XMLHandler.java:**

**package** com.example.parser;  
  
**import** android.util.Log;  
  
**import** org.xml.sax.Attributes;  
**import** org.xml.sax.SAXException;  
**import** org.xml.sax.helpers.DefaultHandler;  
  
**public class** XMLHandler **extends** DefaultHandler {  
 String **elementValue** = **null**;  
 Boolean **elementOn** = **false**;  
 **public static** XMLGettersSetters *data* = **null**;  
  
 **public static** XMLGettersSetters getXMLData() {  
 **return** *data*;  
 }  
  
  
 **public static void** setXMLData(XMLGettersSetters data) {  
 XMLHandler.*data* = data;  
 }  
  
 */\*\*  
 \* This will be called when the tags of the XML starts.  
 \*\*/* @Override  
 **public void** startElement(String uri, String localName, String qName,  
 Attributes attributes) **throws** SAXException {  
  
 **elementOn** = **true**;  
  
 **if** (localName.equals(**"CATALOG"**))  
 {  
 *data* = **new** XMLGettersSetters();  
 }  
  
 **if** (localName.equals(**"CATALOG"**)) {  
 *data* = **new** XMLGettersSetters();  
 }  
 **else if** (localName.equals(**"CD"**)) {  
 **try**{  
 String attributeValue = attributes.getValue(**"attr"**);  
 *data*.setAttribute(attributeValue);  
 }*//end try  
 //catch error if thrown to avoid possible null pointer exception* **catch** (Exception e) {  
 Log.*i*(**"err on handler "** , e.getMessage());}  
 }  
  
 }  
  
 */\*\*  
 \* This will be called when the tags of the XML end.  
 \*\*/* @Override  
 **public void** endElement(String uri, String localName, String qName)  
 **throws** SAXException {  
  
 **elementOn** = **false**;  
  
 */\*\*  
 \* Sets the  
 \* values after retrieving the values from the XML tags  
 \* \*/* **if** (localName.equalsIgnoreCase(**"title"**))  
 *data*.setTitle(**elementValue**);  
 **else if** (localName.equalsIgnoreCase(**"artist"**))  
 *data*.setArtist(**elementValue**);  
 **else if** (localName.equalsIgnoreCase(**"country"**))  
 *data*.setCountry(**elementValue**);  
 **else if** (localName.equalsIgnoreCase(**"company"**))  
 *data*.setCompany(**elementValue**);  
 **else if** (localName.equalsIgnoreCase(**"price"**))  
 *data*.setPrice(**elementValue**);  
 **else if** (localName.equalsIgnoreCase(**"year"**))  
 *data*.setYear(**elementValue**);  
  
  
*// else if(localName.equals("attr"))  
// data.setattr(elementValue);* }  
 */\*\*  
 \* This is called to get the tags value  
 \*\*/* @Override  
 **public void** characters(**char**[] ch, **int** start, **int** length)  
 **throws** SAXException {  
  
 **if** (**elementOn**) {  
 **elementValue** = **new** String(ch, start, length);  
 **elementOn** = **false**;  
 }  
  
 }  
}

**activity\_login.xml:**

*<?***xml version="1.0" encoding="utf-8"***?>*<**RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:tools="http://schemas.android.com/tools"  
 android:id="@+id/activity\_login"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:paddingBottom="@dimen/activity\_vertical\_margin"  
 android:paddingLeft="@dimen/activity\_horizontal\_margin"  
 android:paddingRight="@dimen/activity\_horizontal\_margin"  
 android:paddingTop="@dimen/activity\_vertical\_margin"  
 tools:context="com.example.parser.Login"**>  
  
 <**ScrollView  
 android:id="@+id/s1"  
 android:layout\_width="fill\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginTop="105dp"  
 android:layout\_alignParentTop="true"  
 android:layout\_alignParentStart="true"**>  
  
 <**LinearLayout  
 android:layout\_width="fill\_parent"  
 android:layout\_height="wrap\_content"  
 android:background="@drawable/bg\_shadow\_button"  
 android:orientation="vertical"  
 android:padding="20dp"** >  
  
 <**LinearLayout  
 android:layout\_width="fill\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginTop="15dp"  
 android:orientation="vertical"** >  
  
 <**EditText  
 android:id="@+id/loginusrnm"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:background="@drawable/bottom\_shadow"  
 android:hint="Enter User Name"  
 android:inputType="textPersonName"  
 android:padding="10dp"  
 android:textColor="@color/theme\_color"  
 android:textColorHint="@color/theme\_color"  
 android:typeface="normal"** />  
  
 <**EditText  
 android:id="@+id/loginpasswd"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginTop="10dp"  
 android:background="@drawable/bottom\_shadow"  
 android:hint="Enter Password"  
 android:inputType="textPassword"  
 android:padding="10dp"  
 android:textColor="@color/theme\_color"  
 android:textColorHint="@color/theme\_color"  
 android:typeface="normal"** />  
  
 <**LinearLayout  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_gravity="center"  
 android:layout\_marginTop="15dp"  
 android:gravity="center"  
 android:orientation="horizontal"** >  
  
 <**Button  
 android:id="@+id/btnLogin"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_gravity="center"  
 android:layout\_marginRight="10dp"  
 android:background="@color/theme\_color"  
 android:gravity="center"  
 android:paddingLeft="20dp"  
 android:paddingRight="20dp"  
 android:text="Log In"  
 android:textColor="@color/white"  
 android:textStyle="bold"** />  
  
 </**LinearLayout**>  
 </**LinearLayout**>  
  
 </**LinearLayout**>  
 </**ScrollView**>  
</**RelativeLayout**>

**activity\_main.xml:**

*<?***xml version="1.0" encoding="utf-8"***?>*<**LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 android:layout\_width="fill\_parent"  
 android:layout\_height="fill\_parent"  
 android:orientation="vertical"  
 android:id="@+id/layout"**>  
  
  
</**LinearLayout**>

**Colors.xml:**

*<?***xml version="1.0" encoding="utf-8"***?>*<**resources**>  
 <**color name="colorPrimary"**>#3F51B5</**color**>  
 <**color name="colorPrimaryDark"**>#303F9F</**color**>  
 <**color name="colorAccent"**>#FF4081</**color**>  
 <**color name="list\_item\_title"**>#fff</**color**>  
 <**color name="white"**>#fff</**color**>  
 <**color name="black"**>#000</**color**>  
 <**color name="theme\_color"**>#009976</**color**>  
 <**color name="list\_background"**>#303030</**color**>  
 <**color name="list\_background\_pressed"**>#984238</**color**>  
 <**color name="list\_divider"**>#272727</**color**>  
 <**color name="counter\_text\_bg"**>#626262</**color**>  
 <**color name="counter\_text\_color"**>#c5c4c4</**color**>  
 <**color name="line\_bg"**>#FFB4B4B4</**color**>  
 <**color name="line\_bg\_top"**>#FFF0F0F0</**color**>  
</**resources**>

**Strings.xml:**

<**resources**>  
 <**string name="app\_name"**>Parser</**string**>  
</**resources**>

**Styles.xml:**

<**resources**>  
  
 *<!-- Base application theme. -->* <**style name="AppTheme" parent="Theme.AppCompat.Light.DarkActionBar"**>  
 *<!-- Customize your theme here. -->* <**item name="colorPrimary"**>@color/colorPrimary</**item**>  
 <**item name="colorPrimaryDark"**>@color/colorPrimaryDark</**item**>  
 <**item name="colorAccent"**>@color/colorAccent</**item**>  
 <**item name="android:editTextColor"**>@color/colorAccent</**item**>  
 <**item name="android:textColorPrimary"**>@color/colorAccent</**item**>  
 <**item name="android:textColorSecondary"**>@color/colorAccent</**item**>  
 <**item name="android:textColorTertiary"**>@color/colorAccent</**item**>  
 <**item name="android:textColorPrimaryInverse"**>@color/colorAccent</**item**>  
 <**item name="android:textColorSecondaryInverse"**>@color/colorAccent</**item**>  
 <**item name="android:textColorTertiaryInverse"**>@color/colorAccent</**item**>  
 </**style**>  
  
  
</**resources**>