|  |  |
| --- | --- |
| **Ex: 1 A** | Sending SMS with toast notification from android application |
| **Date:** |

**AIM:** To Develop a mobile application for Sending SMS with toast notification.

###### PROCEDURE:

|  |  |
| --- | --- |
| 1 | Use Android Studio IDE to create an Android application and name it. |
| 2 | Modify *src/MainActivity.java* file and add required code to take care of sending sms. |
| 3 | Modify layout XML file *res/layout/activity\_main.xml* add any GUI component if required. I'm adding a simple GUI to take mobile number and SMS text to be sent and a simple button to send SMS. |
| 4 | No need to define default string constants at res/values/strings.xml. Android studio takes care of default constants. |
| 5 | Modify *AndroidManifest.xml* as shown in the code. |
| 6 | Run the application to launch Android emulator and verify the result of the changes done in the application. |

**AndroidManifest.xml:**

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

<LinearLayout xmlns:android=["htt](http://schemas.android.com/apk/res/android)p[://schemas.android.com/apk/res/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/fstTxt" android:layout\_width="wrap\_content" android:layout\_height="wrap\_content" android:layout\_marginLeft="100dp" android:layout\_marginTop="150dp" android:text="Mobile No" />

<EditText

android:id="@+id/mblTxt" android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content" android:layout\_marginLeft="100dp" android:ems="10"/>

<TextView

android:id="@+id/secTxt" android:layout\_width="wrap\_content" android:layout\_height="wrap\_content" android:text="Message" android:layout\_marginLeft="100dp" />

<EditText

android:id="@+id/msgTxt" android:layout\_width="wrap\_content" android:layout\_height="wrap\_content" android:layout\_marginLeft="100dp" android:ems="10" />

<Button

android:id="@+id/btnSend" android:layout\_width="wrap\_content" android:layout\_height="wrap\_content" android:layout\_marginLeft="100dp" android:text="Send SMS" />

</LinearLayout>

###### MainActivity.java

import android.content.Intent; import android.net.Uri;

import android.provider.Telephony;

import android.support.v7.app.AppCompatActivity; import android.os.Bundle;

import android.telephony.SmsManager; import android.view.View;

import android.widget.Button; import android.widget.EditText; import android.widget.Toast;

public class MainActivity extends AppCompatActivity { private EditText txtMobile;

private EditText txtMessage; private Button btnSms;

@Override

protected void onCreate(Bundle savedInstanceState) { super.onCreate(savedInstanceState); setContentView(R.layout.activity\_main); txtMobile = (EditText)findViewById(R.id.mblTxt); txtMessage = (EditText)findViewById(R.id.msgTxt); btnSms = (Button)findViewById(R.id.btnSend);

btnSms.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View v) {

try{

SmsManager smgr = SmsManager.getDefault();

smgr.sendTextMessage(txtMobile.getText().toString(),null,txtMessage

.getText().toString(),null,null);

Toast.makeText(MainActivity.this, "SMS Sent Successfully", Toast.LENGTH\_SHORT).show();

}

catch (Exception e){

Toast.makeText(MainActivity.this, "SMS Failed to Send, Please try again", Toast.LENGTH\_SHORT).show();

}

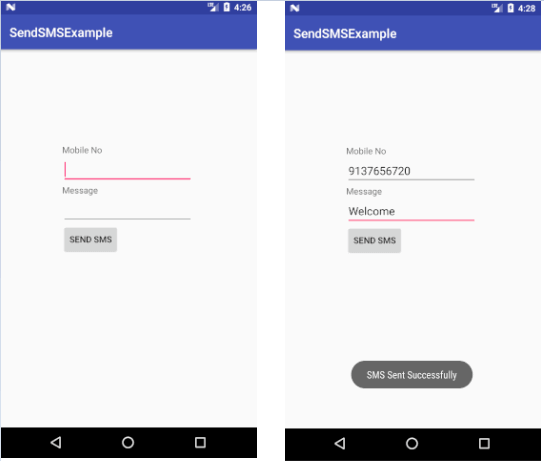
}

});

}

}

###### OUTPUT:



**RESULT:**

Thus application for sending SMS with toast notification was successfully completed.

|  |  |
| --- | --- |
| **Ex: 1 B** | Sending an email from android device |
| **Date:** |

###### AIM:

To develop a mobile application for sending an email from android application

###### PROCEDURE:

|  |  |
| --- | --- |
| 1 | Use Android studio to create an Android application |
| 2 | Modify *src/MainActivity.java* file and add required code to take care of sending email. |
| 3 | Modify layout XML file *res/layout/activity\_main.xml* add any GUI component if required. I'm adding a simple button to launch Email Client. |
| 4 | Modify *res/values/strings.xml* to define required constant values |
| 5 | Modify *AndroidManifest.xml* as shown below |
| 6 | Run the application to launch Android emulator and verify the result of the changes done in the application. |

**activity\_main.xml**

<LinearLayout xmlns:android="<http://schemas.android.com/apk/res/android>"

android:layout\_width="fill\_parent" android:layout\_height="fill\_parent" android:orientation="vertical" >

<TextView

android:id="@+id/textView1"

android:layout\_width="wrap\_content" android:layout\_height="wrap\_content" android:text="Sending Mail Example" android:layout\_alignParentTop="true" android:layout\_centerHorizontal="true" android:textSize="30dp" />

<TextView

android:id="@+id/textView2" android:layout\_width="wrap\_content" android:layout\_height="wrap\_content" android:text="Tutorials point " android:textColor="#ff87ff09" android:textSize="30dp" android:layout\_above="@+id/imageButton" android:layout\_alignRight="@+id/imageButton" android:layout\_alignEnd="@+id/imageButton" />

<ImageButton android:layout\_width="wrap\_content" android:layout\_height="wrap\_content" android:id="@+id/imageButton" android:src="@drawable/abc" android:layout\_centerVertical="true" android:layout\_centerHorizontal="true" />

<Button

android:id="@+id/sendEmail" android:layout\_width="fill\_parent" android:layout\_height="wrap\_content" android:text="@string/compose\_email"/>

</LinearLayout>

###### MainActivity.java

import android.net.Uri; import android.os.Bundle; import android.app.Activity;

import android.content.Intent; import android.util.Log; import android.view.Menu; import android.view.View; import android.widget.Button; import android.widget.Toast;

public class MainActivity extends Activity {

@Override

protected void onCreate(Bundle savedInstanceState) { super.onCreate(savedInstanceState); setContentView(R.layout.activity\_main);

Button startBtn = (Button) findViewById(R.id.sendEmail); startBtn.setOnClickListener(new View.OnClickListener() {

public void onClick(View view) { sendEmail();

}

});

}

protected void sendEmail() { Log.i("Send email", "");

String[] TO = {""};

String[] CC = {""};

Intent emailIntent = new Intent(Intent.ACTION\_SEND);

emailIntent.setData(Uri.parse("mailto:")); emailIntent.setType("text/plain"); emailIntent.putExtra(Intent.EXTRA\_EMAIL, TO); emailIntent.putExtra(Intent.EXTRA\_CC, CC); emailIntent.putExtra(Intent.EXTRA\_SUBJECT, "Your subject"); emailIntent.putExtra(Intent.EXTRA\_TEXT, "Email message goes

here");

try {

startActivity(Intent.createChooser(emailIntent, "Send mail..."));

finish();

Log.i("Finished sending email...", "");

} catch (android.content.ActivityNotFoundException ex) {

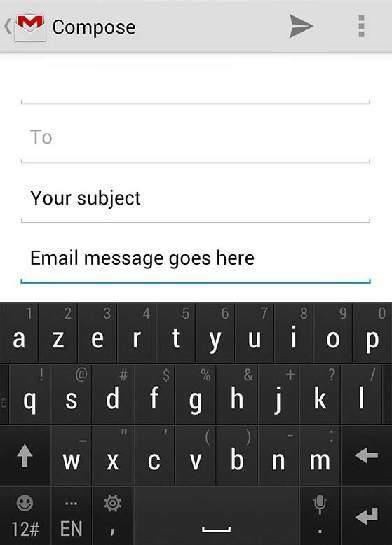
Toast.makeText(MainActivity.this, "There is no email client installed.", Toast.LENGTH\_SHORT).show();

}

}

}

**OUTPUT:**



**RESULT:**

Thus we successfully developed mobile application for sending an email.

|  |  |
| --- | --- |
| **Ex: 2** | Implement an application that implements Multi-threading |
| **Date:** |

###### AIM:

To implement an application that implements Multi-threading

###### PROCEDURE:

|  |  |
| --- | --- |
| 1 | Use Android studio to create an Android application |
| 2 | Modify *src/MainActivity.java* file and add required code to take care of multithread. |
| 3 | Modify layout XML file *res/layout/activity\_main.xml* add any GUI component if required. |
| 4 | Modify *res/values/strings.xml* to define required constant values |
| 5 | Modify *AndroidManifest.xml* as shown below |
| 6 | Run the application to launch Android emulator and verify the result of the changes done in the application. |

**Activity\_main.xml:**

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

<LinearLayout xmlns:android=["ht](http://schemas.android.com/apk/res/android)t[p://schemas.android.com/apk/res/android](http://schemas.android.com/apk/res/android)" android:layout\_width="match\_parent" android:layout\_height="match\_parent"

android:orientation="vertical" >

<ImageView android:id="@+id/imageView" android:layout\_width="250dp" android:layout\_height="250dp" android:layout\_margin="50dp" android:layout\_gravity="center" />

<Button

android:id="@+id/button" android:layout\_width="wrap\_content" android:layout\_height="wrap\_content" android:layout\_margin="10dp" android:layout\_gravity="center" android:text="Load Image 1" />

<Button

android:id="@+id/button2" android:layout\_width="wrap\_content" android:layout\_height="wrap\_content" android:layout\_margin="10dp" android:layout\_gravity="center" android:text="Load image 2" />

</LinearLayout>

**MainActivity.java**

import android.os.Bundle;

import android.support.v7.app.AppCompatActivity; import android.view.View;

import android.widget.Button; import android.widget.ImageView;

public class MainActivity extends AppCompatActivity

{

ImageView img; Button bt1,bt2;

@Override

protected void onCreate(Bundle savedInstanceState)

{

super.onCreate(savedInstanceState); setContentView(R.layout.activity\_main);

bt1 = (Button)findViewById(R.id.button); bt2= (Button) findViewById(R.id.button2);

img = (ImageView)findViewById(R.id.imageView);

bt1.setOnClickListener(new View.OnClickListener()

{

@Override

public void onClick(View v)

{

new Thread(new Runnable()

{

@Override public void run()

{

img.post(new Runnable()

{

@Override public void run()

{

}

});

}

img.setImageResource(R.drawable.india1);

}

});

}).start();

bt2.setOnClickListener(new View.OnClickListener()

{

@Override

public void onClick(View v)

{

new Thread(new Runnable()

{

@Override public void run()

{

img.post(new Runnable()

{

@Override public void run()

{

}

});

}

img.setImageResource(R.drawable.india2);

}

});

}

}

}).start();

###### OUTPUT:



**RESULT:**

Thus Android Application that implements Multi-threading is developed and executed successfully.

|  |  |
| --- | --- |
| **Ex: 3** | Using audio and video functions in android application. |
| **Date:** |

###### AIM:

To develop a android application for audio and video functions in an application.

###### PROCEDURE:

|  |  |
| --- | --- |
| 1 | Use Android studio to create an Android application |
| 2 | Modify *src/MainActivity.java* file and add required code to take care of audio and video functions. |
| 3 | Modify layout XML file *res/layout/activity\_main.xml* add any GUI component if required. |
| 4 | Modify *res/values/strings.xml* to define required constant values |
| 5 | Modify *AndroidManifest.xml* as shown below |
| 6 | Run the application to launch Android emulator and verify the result of the changes done in the application. |

activity\_main.xml

<RelativeLayout xmlns:androclass=["http://schemas.android.com/apk/](http://schemas.android.com/apk/res/android)r[es/android](http://schemas.android.com/apk/res/android)" xmlns:tools="[http://schemas.android.com/tools"](http://schemas.android.com/tools) 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=".MainActivity" >

<TextView android:id="@+id/textView1" android:layout\_width="wrap\_content" android:layout\_height="wrap\_content" android:layout\_alignParentTop="true"

android:layout\_marginTop="30dp" android:text="Audio Controller" />

<Button

android:id="@+id/button1" style="?android:attr/buttonStyleSmall" android:layout\_width="wrap\_content" android:layout\_height="wrap\_content" android:layout\_alignLeft="@+id/textView1" android:layout\_below="@+id/textView1" android:layout\_marginTop="48dp" android:text="start" />

<Button

android:id="@+id/button2" style="?android:attr/buttonStyleSmall" android:layout\_width="wrap\_content" android:layout\_height="wrap\_content" android:layout\_alignTop="@+id/button1" android:layout\_toRightOf="@+id/button1" android:text="pause" />

<Button

android:id="@+id/button3" style="?android:attr/buttonStyleSmall" android:layout\_width="wrap\_content" android:layout\_height="wrap\_content" android:layout\_alignTop="@+id/button2" android:layout\_toRightOf="@+id/button2" android:text="stop" />

</RelativeLayout>

*MainActivity.java*

import android.media.MediaPlayer; import android.os.Bundle;

import android.os.Environment; import android.app.Activity; import android.view.Menu; import android.view.View;

import android.view.View.OnClickListener; import android.widget.Button;

public class MainActivity extends Activity { Button start,pause,stop;

@Override

protected void onCreate(Bundle savedInstanceState) { super.onCreate(savedInstanceState); setContentView(R.layout.activity\_main);

start=(Button)findViewById(R.id.button1); pause=(Button)findViewById(R.id.button2); stop=(Button)findViewById(R.id.button3);

//creating media player

final MediaPlayer mp=new MediaPlayer(); try{

//you can change the path, here path is external directory(e.g. sdcard) /Music/maine.mp3 mp.setDataSource(Environment.getExternalStorageDirectory().getPath()+"/Music/maine.mp3");

mp.prepare();

}catch(Exception e){e.printStackTrace();} start.setOnClickListener(new OnClickListener() {

@Override

public void onClick(View v) { mp.start();

}

});

pause.setOnClickListener(new OnClickListener() {

@Override

public void onClick(View v) { mp.pause();

}

});

stop.setOnClickListener(new OnClickListener() {

@Override

public void onClick(View v) { mp.stop();

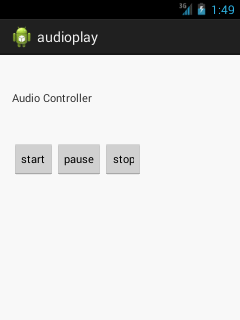
}

});

}

}

**OUTPUT:**



###### RESULT:

Thus the Android Application that implements Multi-threading was developed and executed successfully.

|  |  |
| --- | --- |
| **Ex: 4** | Develop an application that makes use of RSS Feed |
| **Date:** |

###### AIM:

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

###### PROCEDURE:

|  |  |
| --- | --- |
| 1 | Use Android studio to create an Android application |
| 2 | Modify *src/MainActivity.java* file and add required code to take care of makes use of RSS Feed. |
| 3 | Modify layout XML file *res/layout/activity\_main.xml* add any GUI component if required. |
| 4 | Modify *res/values/strings.xml* to define required constant values |
| 5 | Modify *AndroidManifest.xml* as shown below |
| 6 | Run the application to launch Android emulator and verify the result of the changes done in the application. |

**Activity\_main.xml**

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

<LinearLayout xmlns:android=["http://schemas.android.com/](http://schemas.android.com/apk/res/android)a[pk/res/android"](http://schemas.android.com/apk/res/android) android:layout\_width="fill\_parent"

android:layout\_height="fill\_parent" android:orientation="vertical" >

<ListView android:id="@+id/listView"

android:layout\_width="match\_parent" android:layout\_height="wrap\_content" />

</LinearLayout>

### MainActivity.java:

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.ArrayAdapter; 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 onListItemClick(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)

{

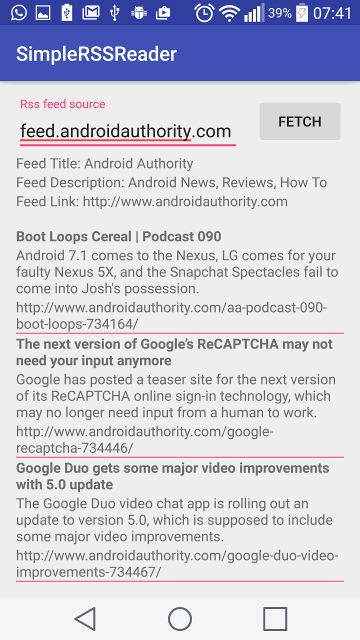
return null;

}

}

}

###### OUTPUT:



**RESULT:**

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

|  |  |
| --- | --- |
| **Ex: 5** | Application development using web service in android |
| **Date:** |

###### AIM:

To develop an Application development using web service in android

###### PROCEDURE:

|  |  |
| --- | --- |
| 1 | Use Android studio to create an Android application |
| 2 | Modify *src/MainActivity.java* file and add required code. |
| 3 | Modify layout XML file *res/layout/activity\_main.xml* add any GUI component if required. |
| 4 | Modify *res/values/strings.xml* to define required constant values |
| 5 | Modify *AndroidManifest.xml* as shown below |
| 6 | Run the application to launch Android emulator and verify the result of the changes done in the application. |

*activity\_main.xml:*

<RelativeLayout xmlns:android=["http://schemas.android.com/](http://schemas.android.com/apk/res/android)a[pk/res/android"](http://schemas.android.com/apk/res/android) xmlns:t[ools="ht](http://schemas.android.com/tools)tp:/[/sc](http://schemas.android.com/tools)he[mas.android.com/tools"](http://schemas.android.com/tools) 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=".MainActivity" >

<EditText android:id="@+id/editText1"

android:layout\_width="wrap\_content" android:layout\_height="wrap\_content" android:layout\_alignParentTop="true"

android:layout\_centerHorizontal="true" android:hint="Username" android:ems="10" >

<requestFocus />

</EditText>

<EditText android:id="@+id/editText2"

android:layout\_width="wrap\_content" android:layout\_height="wrap\_content" android:layout\_alignLeft="@+id/editText1" android:layout\_below="@+id/editText1" android:layout\_marginTop="67dp" android:ems="10"

android:hint="Password" android:inputType="textPassword" />

<Button

android:id="@+id/button2" android:layout\_width="wrap\_content" android:layout\_height="wrap\_content" android:layout\_alignParentBottom="true" android:layout\_marginBottom="24dp" android:layout\_toRightOf="@+id/button1" android:text="New User" />

<ProgressBar android:id="@+id/progressBar1" style="?android:attr/progressBarStyleLarge" android:layout\_width="wrap\_content" android:layout\_height="wrap\_content" android:layout\_alignLeft="@+id/button1" android:layout\_below="@+id/editText2" android:layout\_marginTop="22dp" />

<Button

android:id="@+id/button1" android:layout\_width="wrap\_content" android:layout\_height="wrap\_content" android:layout\_alignLeft="@+id/editText2" android:layout\_below="@+id/progressBar1" android:layout\_marginLeft="22dp" android:text="Login" />

</RelativeLayout>

*MainActivity.java*

import java.io.BufferedReader;

import java.io.InputStream; import java.io.InputStreamReader; import java.util.ArrayList;

import java.util.List;

import org.apache.http.HttpEntity; import org.apache.http.HttpResponse; import org.apache.http.NameValuePair; import org.apache.http.client.HttpClient;

import org.apache.http.client.entity.UrlEncodedFormEntity; import org.apache.http.client.methods.HttpPost;

import org.apache.http.impl.client.DefaultHttpClient; import org.apache.http.message.BasicNameValuePair; import android.os.AsyncTask;

import android.os.Bundle; import android.app.Activity; import android.content.Intent; import android.view.View;

import android.view.View.OnClickListener; import android.widget.Button;

import android.widget.EditText; import android.widget.ProgressBar; import android.widget.Toast;

public class MainActivity extends Activity { EditText password,userName;

Button login,resister; ProgressBar progressBar;

protected void onCreate(Bundle savedInstanceState) { super.onCreate(savedInstanceState); setContentView(R.layout.activity\_main); password=(EditText) findViewById(R.id.editText2); userName=(EditText) findViewById(R.id.editText1); login=(Button) findViewById(R.id.button1); resister=(Button) findViewById(R.id.button2);

//progess\_msz.setVisibility(View.GONE); progressBar=(ProgressBar) findViewById(R.id.progressBar1); progressBar.setVisibility(View.GONE);

resister.setOnClickListener(new OnClickListener() {

@Override

public void onClick(View arg0) {

// TODO Auto-generated method stub

Intent intent=new Intent(MainActivity.this,ResisterUser.class);

}

});

startActivity(intent);

login.setOnClickListener(new OnClickListener() {

public void onClick(View v) { progressBar.setVisibility(View.VISIBLE);

String s1=userName.getText().toString(); String s2=password.getText().toString(); new ExecuteTask().execute(s1,s2);

}

});

}

class ExecuteTask extends AsyncTask<String, Integer, String>

{

@Override

protected String doInBackground(String... params) { String res=PostData(params);

return res;

}

@Override

protected void onPostExecute(String result) { progressBar.setVisibility(View.GONE);

//progess\_msz.setVisibility(View.GONE); Toast.makeText(getApplicationContext(), result, 3000).show();

}

}

public String PostData(String[] valuse) { String s="";

try

{

HttpClient httpClient=new DefaultHttpClient();

HttpPost httpPost=new HttpPost("http://10.0.0.8:7777/HttpPostServlet/servlet/Login");

List<NameValuePair> list=new ArrayList<NameValuePair>(); list.add(new BasicNameValuePair("name", valuse[0])); list.add(new BasicNameValuePair("pass",valuse[1])); httpPost.setEntity(new UrlEncodedFormEntity(list));

HttpResponse httpResponse= httpClient.execute(httpPost);

HttpEntity httpEntity=httpResponse.getEntity(); s= readResponse(httpResponse);

}

catch(Exception exception) {} return s;

}

public String readResponse(HttpResponse res) { InputStream is=null;

String return\_text=""; try {

is=res.getEntity().getContent();

BufferedReader bufferedReader=new BufferedReader(new InputStreamReader(is)); String line="";

StringBuffer sb=new StringBuffer();

while ((line=bufferedReader.readLine())!=null)

{

sb.append(line);

}

return\_text=sb.toString();

} catch (Exception e)

{

}

return return\_text;

}

}

*RegisterUser.java*

import java.util.ArrayList; import java.util.List;

import org.apache.http.NameValuePair; import org.apache.http.client.HttpClient;

import org.apache.http.client.entity.UrlEncodedFormEntity; import org.apache.http.client.methods.HttpPost;

import org.apache.http.impl.client.DefaultHttpClient; import org.apache.http.message.BasicNameValuePair; import android.os.AsyncTask;

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

import android.view.View.OnClickListener; import android.widget.Button;

import android.widget.EditText; import android.widget.ProgressBar;

public class ResisterUser extends Activity { EditText userName,passwprd;

Button resister,login; ProgressBar progressBar;

protected void onCreate(Bundle savedInstanceState) { super.onCreate(savedInstanceState); setContentView(R.layout.activity\_resister\_user); userName=(EditText) findViewById(R.id.editText1);; passwprd=(EditText) findViewById(R.id.editText2); resister=(Button) findViewById(R.id.button1);

progressBar=(ProgressBar) findViewById(R.id.progressBar1); progressBar.setVisibility(View.GONE);

resister.setOnClickListener(new OnClickListener() {

@Override

public void onClick(View v) { progressBar.setVisibility(View.VISIBLE);

}

});

String s1=userName.getText().toString(); String s2=passwprd.getText().toString(); new ExecuteTask().execute(s1,s2);

}

class ExecuteTask extends AsyncTask<String, Integer, String>

{

@Override

protected String doInBackground(String... params) {

PostData(params); return null;

}

@Override

protected void onPostExecute(String result) {

progressBar.setVisibility(View.GONE);

}

}

public void PostData(String[] valuse) { try

{

HttpClient httpClient=new DefaultHttpClient(); HttpPost httpPost=new HttpPost(

"http://10.0.0.8:7777/HttpPostServlet/servlet/httpPostServlet"); List<NameValuePair> list=new ArrayList<NameValuePair>();

list.add(new BasicNameValuePair("name", valuse[0])); list.add(new BasicNameValuePair("pass",valuse[1])); httpPost.setEntity(new UrlEncodedFormEntity(list)); httpClient.execute(httpPost);

}

catch(Exception e)

{

System.out.println(e);

}

}

}

AndroidManifest.xml

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

<manifest xmlns:android=["http://schemas.android.com/](http://schemas.android.com/apk/res/android)a[pk/res/android"](http://schemas.android.com/apk/res/android) package="com.example.newrestapi"

android:versionCode="1" android:versionName="1.0" >

<uses-sdk android:minSdkVersion="8" android:targetSdkVersion="17" />

<uses-permission android:name="android.permission.INTERNET" />

<application android:allowBackup="true"

android:icon="@drawable/ic\_launcher" android:label="@string/app\_name" android:theme="@style/AppTheme" >

<activity

android:name="com.example.newrestapi.MainActivity" android:label="@string/app\_name" >

<intent-filter>

<action android:name="android.intent.action.MAIN" />

<category android:name="android.intent.category.LAUNCHER" />

</intent-filter>

</activity>

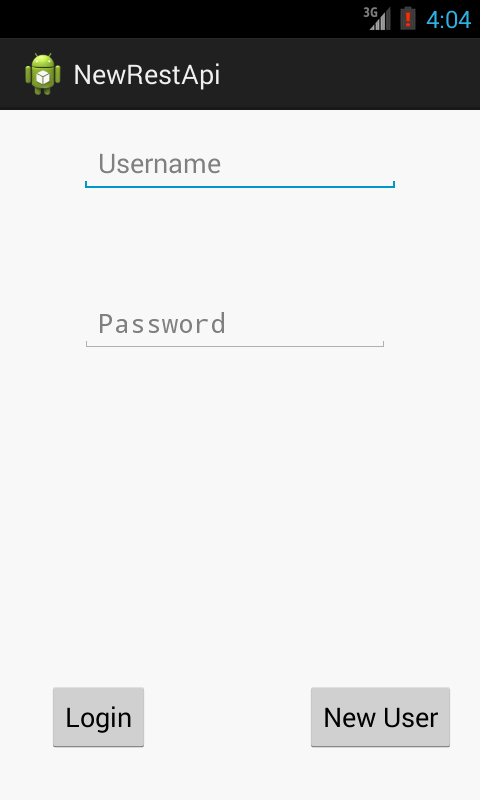
<activity android:name="com.example.newrestapi.ResisterUser" android:label="@string/title\_activity\_resister\_user" >

</activity>

</application>

</manifest>

###### OUTPUT:

 Graphical user interface, text, application

Description automatically generated

**RESULT:**

Thus a web service in android was successfully developed.

|  |  |
| --- | --- |
| **Ex: 6** | Android application for obtaining user location using GPS |
| **Date:** |

###### AIM:

To create Android application for obtaining user location using GPS

###### PROCEDURE:

|  |  |
| --- | --- |
| 1 | Use Android studio to create an Android application |
| 2 | Modify *src/MainActivity.java* file and add required code. |
| 3 | Modify layout XML file *res/layout/activity\_main.xml* add any GUI component if required. |
| 4 | Modify *res/values/strings.xml* to define required constant values |
| 5 | Modify *AndroidManifest.xml* as shown below |
| 6 | Run the application to launch Android emulator and verify the result of the changes done in the application. |

activity\_main.xml:

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

<LinearLayout xmlns:android="<http://schemas.android.com/apk/res/android>" android:layout\_width="match\_parent" android:layout\_height="match\_parent"

android:layout\_margin="16dp" android:gravity="center\_horizontal" android:orientation="vertical">

<TextView android:id="@+id/location"

android:layout\_width="wrap\_content" android:layout\_height="wrap\_content" android:text="Current GPS Loaction" android:textSize="24sp" android:textStyle="bold" />

<Button android:id="@+id/getLocation" android:text="Get location"

android:layout\_width="wrap\_content" android:layout\_height="wrap\_content" />

</LinearLayout> MainActivity.java:

import android.content.pm.PackageManager; import android.location.Location;

import android.os.Bundle;

import android.support.v4.app.ActivityCompat; import android.support.v7.app.AppCompatActivity; import android.view.View;

import android.widget.Button; import android.widget.TextView;

import com.google.android.gms.location.FusedLocationProviderClient; import com.google.android.gms.location.LocationServices;

import com.google.android.gms.tasks.OnSuccessListener;

import static android.Manifest.permission.ACCESS\_FINE\_LOCATION; public class MainActivity extends AppCompatActivity {

private FusedLocationProviderClient client;

@Override

protected void onCreate(Bundle savedInstanceState) { super.onCreate(savedInstanceState); setContentView(R.layout.activity\_main); requestPermission();

client = LocationServices.getFusedLocationProviderClient(this); Button button = findViewById(R.id.getLocation); button.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View v) {

if (ActivityCompat.checkSelfPermission(MainActivity.this, ACCESS\_FINE\_LOCATION) != PackageManager.PERMISSION\_GRANTED ) {

return;

}

client.getLastLocation().addOnSuccessListener(MainActivity.this, new OnSuccessListener() {

@Override

public void onSuccess(Location location) { if (location != null) {

TextView textView = findViewById(R.id.location); textView.setText(location.toString());

}

});

}

}

}

});

private void requestPermission(){

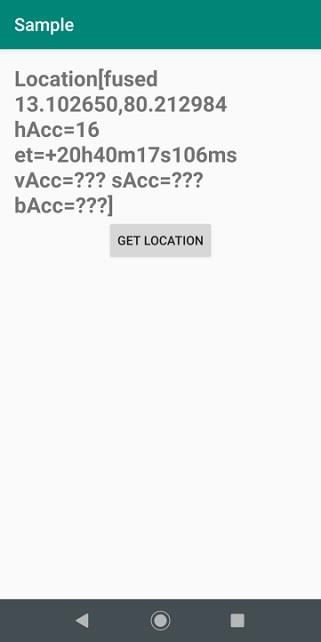
1);

}

}

ActivityCompat.requestPermissions(this, new String[]{ACCESS\_FINE\_LOCATION

###### OUTPUT :



**RESULT:**

Thus the program to obtaining user location using GPS was executed successfully.

|  |  |
| --- | --- |
| **Ex: 7** | Android application for database connectivity with MySQL |
| **Date:** |

###### AIM:

To create android application for database connectivity with MySQL

###### PROCEDURE:

|  |  |
| --- | --- |
| 1 | Use Android studio to create an Android application |
| 2 | Modify *src/MainActivity.java* file and add required code. |
| 3 | Modify layout XML file *res/layout/activity\_main.xml* add any GUI component if required. |
| 4 | Modify *res/values/strings.xml* to define required constant values |
| 5 | Modify *AndroidManifest.xml* as shown below |
| 6 | Run the application to launch Android emulator and verify the result of the changes done in the application. |

MySQL Database Script : create schema myDB

use myDB

create table tblCountries (

ID int NOT NULL AUTO\_INCREMENT primary key, Country varchar(255) NOT NULL

)

Insert into tblCountries (Country) values ('India') Insert into tblCountries (Country) values ('Australia') Insert into tblCountries (Country) values ('Mauritius') Insert into tblCountries (Country) values ('USA') Insert into tblCountries (Country) values ('England')

Insert into tblCountries (Country) values ('New Zealand') Insert into tblCountries (Country) values ('Spain')

Select \* from tblCountries

select distinct Country from tblCountries

###### activity\_main.xml:

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

<LinearLayout xmlns:android=["http://schemas.android.com/](http://schemas.android.com/apk/res/android)a[pk/res/android"](http://schemas.android.com/apk/res/android) xmlns:app=["http://sche](http://schemas.android.com/apk/res-auto)m[as.android.com/apk/res-auto"](http://schemas.android.com/apk/res-auto) xmlns:t[ools="ht](http://schemas.android.com/tools)tp:/[/schemas.android.com/tool](http://schemas.android.com/tools)s" android:layout\_width="match\_parent" android:layout\_height="match\_parent"

android:fitsSystemWindows="true" android:orientation="vertical" android:padding="5dp">

<TextView android:layout\_width="match\_parent" android:layout\_height="wrap\_content" android:layout\_margin="5dp" android:padding="5dp" android:text="Android MySQL Application" android:textColor="@color/colorAccent" android:textSize="20sp" android:textStyle="bold" />

<TextView android:id="@+id/txtData" android:layout\_width="match\_parent" android:layout\_height="wrap\_content" android:layout\_margin="2dp"

android:text="Your Data will be Displayed here" android:textStyle="bold" />

<Button android:id="@+id/btnFetch"

android:layout\_width="wrap\_content" android:layout\_height="wrap\_content" android:layout\_gravity="center\_horizontal" android:layout\_margin="2dp" android:background="@color/colorPrimaryDark" android:minWidth="250dp"

android:text="Fetch Data" android:textColor="#fff" />

<Button android:id="@+id/btnClear"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content" android:layout\_gravity="center\_horizontal" android:layout\_margin="2dp" android:background="@color/colorAccent" android:minWidth="250dp" android:text="Clear" android:textColor="#fff" />

</LinearLayout>

###### MainActivity.java:

package parallelcodes.mysqlapp;

import android.os.AsyncTask; import android.os.Bundle;

import android.support.v7.app.AppCompatActivity; import android.view.View;

import android.widget.Button; import android.widget.TextView; import android.widget.Toast;

import java.sql.Connection; import java.sql.DriverManager; import java.sql.ResultSet;

import java.sql.ResultSetMetaData; import java.sql.Statement;

public class MainActivity extends AppCompatActivity {

private static final String url = "jdbc:mysql://192.168.0.192:3306/myDB"; private static final String user = "hitesh";

private static final String pass = "1234"; Button btnFetch,btnClear;

TextView txtData;

@Override

protected void onCreate(Bundle savedInstanceState) { super.onCreate(savedInstanceState); setContentView(R.layout.activity\_main);

txtData = (TextView) this.findViewById(R.id.txtData); btnFetch = (Button) findViewById(R.id.btnFetch); btnClear = (Button) findViewById(R.id.btnClear);

btnFetch.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View v) {

// TODO Auto-generated method stub

ConnectMySql connectMySql = new ConnectMySql(); connectMySql.execute("");

}

});

btnClear.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View view) { txtData.setText("");

}

});

}

private class ConnectMySql extends AsyncTask<String, Void, String> { String res = "";

@Override

protected void onPreExecute() { super.onPreExecute();

Toast.makeText(MainActivity.this, "Please wait...", Toast.LENGTH\_SHORT)

.show();

}

@Override

protected String doInBackground(String... params) { try {

Class.forName("com.mysql.jdbc.Driver");

Connection con = DriverManager.getConnection(url, user, pass); System.out.println("Databaseection success");

String result = "Database Connection Successful\n"; Statement st = con.createStatement();

ResultSet rs = st.executeQuery("select distinct Country from tblCountries"); ResultSetMetaData rsmd = rs.getMetaData();

while (rs.next()) {

result += rs.getString(1).toString() + "\n";

}

res = result;

} catch (Exception e) { e.printStackTrace(); res = e.toString();

}

return res;

}

@Override

protected void onPostExecute(String result) {

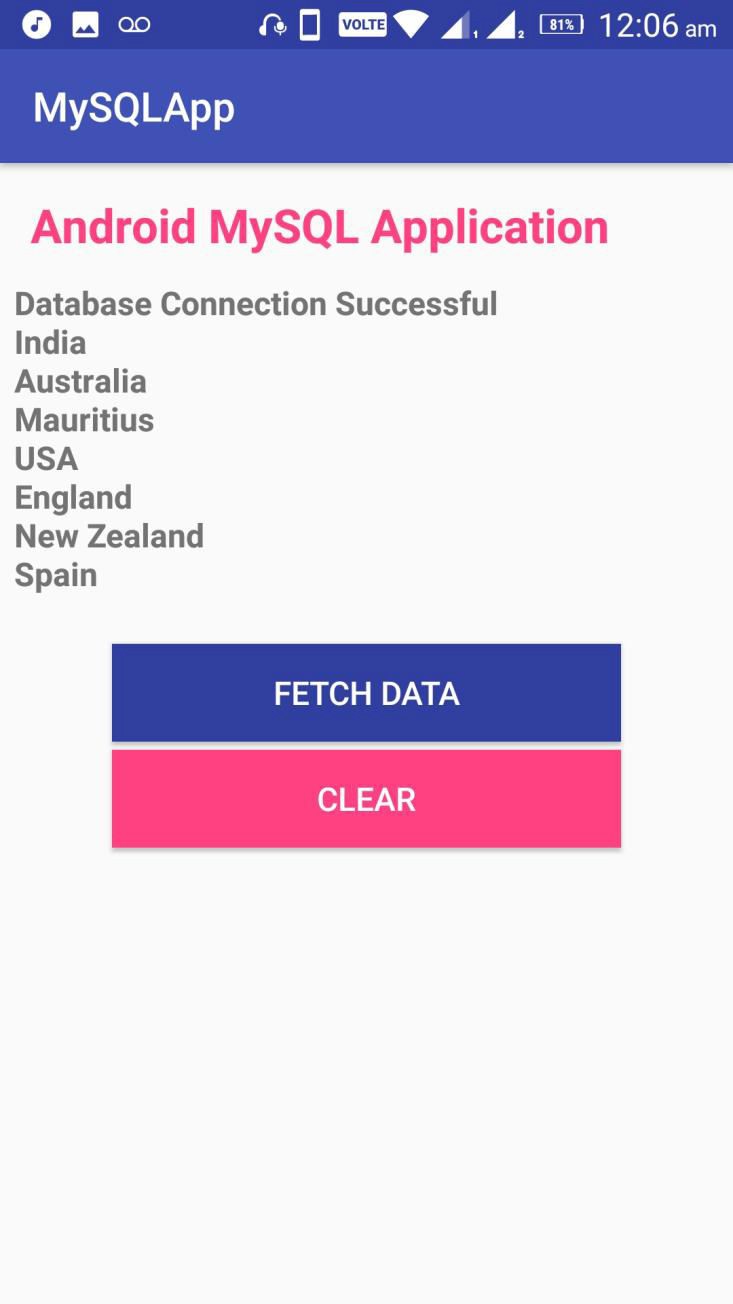
txtData.setText(result);

}

}

}

**OUTPUT:**



###### RESULT:

Thus the application was created successfully.

|  |  |
| --- | --- |
| **Ex: 8** | Implement an application that writes data to the SD card. |
| **Date:** |

###### AIM:

To develop a Android Application that writes data to the SD Card.

###### PROCEDURE:

|  |  |
| --- | --- |
| 1 | Use Android studio to create an Android application |
| 2 | Modify *src/MainActivity.java* file and add required code. |
| 3 | Modify layout XML file *res/layout/activity\_main.xml* add any GUI component if required. |
| 4 | Modify *res/values/strings.xml* to define required constant values |
| 5 | Modify *AndroidManifest.xml* as shown below |
| 6 | Run the application to launch Android emulator and verify the result of the changes done in the application. |

**Activity\_main.xml:**

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

<LinearLayout xmlns:andr[oid="htt](http://schemas.android.com/apk/res/android)p:/[/sc](http://schemas.android.com/apk/res/android)he[mas.android.com/apk/res/android"](http://schemas.android.com/apk/res/android) android:layout\_width="match\_parent" android:layout\_height="match\_parent" android:layout\_margin="20dp" android:orientation="vertical"

<EditText android:id="@+id/editText" android:layout\_width="match\_parent"

android:layout\_height="wrap\_content" android:singleLine="true" android:textSize="30dp" />

<Button android:id="@+id/button"

android:layout\_width="match\_parent" android:layout\_height="wrap\_content" android:layout\_margin="10dp" android:text="Write Data" android:textSize="30dp" />

<Button android:id="@+id/button2"

android:layout\_width="match\_parent" android:layout\_height="wrap\_content" android:layout\_margin="10dp" android:text="Read data" android:textSize="30dp" />

<Button android:id="@+id/button3"

android:layout\_width="match\_parent" android:layout\_height="wrap\_content" android:layout\_margin="10dp" android:text="Clear" android:textSize="30dp" />

</LinearLayout>

###### AndroidManifest.xml:

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

<manifest xmlns:andr[oid="htt](http://schemas.android.com/apk/res/android)p:/[/sc](http://schemas.android.com/apk/res/android)he[mas.android.com/apk/res/android"](http://schemas.android.com/apk/res/android) package="com.example.exno9" >

<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=".MainActivity" >

<intent-filter>

<action android:name="android.intent.action.MAIN" />

<category android:name="android.intent.category.LAUNCHER" />

</intent-filter>

</activity>

</application>

</manifest>

###### MainActivity.java:

import android.os.Bundle;

import android.support.v7.app.AppCompatActivity; import android.view.View;

import android.widget.Button; import android.widget.EditText; import android.widget.Toast; import java.io.BufferedReader; import java.io.File;

import java.io.FileInputStream; import java.io.FileOutputStream; import java.io.InputStreamReader;

public class MainActivity extends AppCompatActivity

{

EditText e1;

Button write,read,clear;

@Override

protected void onCreate(Bundle savedInstanceState){ super.onCreate(savedInstanceState); setContentView(R.layout.activity\_main);

e1= (EditText) findViewById(R.id.editText); write= (Button) findViewById(R.id.button); read= (Button) findViewById(R.id.button2); clear= (Button) findViewById(R.id.button3);

write.setOnClickListener(new View.OnClickListener()

{@Override

public void onClick(View v)

{

String message=e1.getText().toString(); try{

File f=new File("/sdcard/myfile.txt"); f.createNewFile();

FileOutputStream fout=new FileOutputStream(f); fout.write(message.getBytes());

fout.close(); Toast.makeText(getBaseContext(),"Data Written in SDCARD",Toast.LENGTH\_LONG).show();

}

catch (Exception e)

{

Toast.makeText(getBaseContext(),e.getMessage(),Toast. LENGTH\_LONG).show();

}

}

});

read.setOnClickListener(new View.OnClickListener()

{

@Override

public void onClick(View v)

{

String message; String buf = ""; try

{

File f = new File("/sdcard/myfile.txt"); FileInputStream fin = new FileInputStream(f); BufferedReader br = new BufferedReader(new InputStreamReader(fin));

while ((message = br.readLine()) != null)

{

buf += message;

}

e1.setText(buf); br.close();

fin.close();

Toast.makeText(getBaseContext(),"Data Recived from SDCARD",Toast.LENGTH\_LONG).show();

}

catch (Exception e)

{

Toast.makeText(getBaseContext(), e.getMessage(), Toast.LENGTH\_LONG).show();

}

} });

clear.setOnClickListener(new View.OnClickListener() {

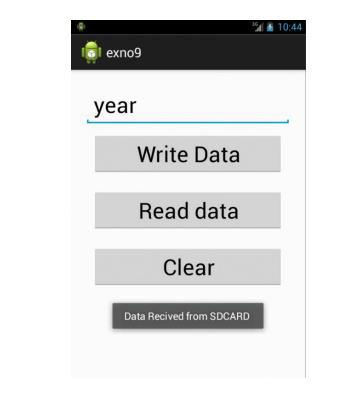
@Override

public void onClick(View v{ e1.setText(""); }

});

}}

###### OUTPUT:



**RESULT:**

Thus Android Application that writes data to the SD Card was developed and executed successfully.

|  |  |
| --- | --- |
| **Ex: 9** | Android application to generate OTP for user verification |
| **Date:** |

###### AIM:

To create an android application to generate OTP for user verification

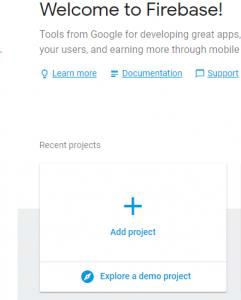
###### PROCEDURE:

STEP1: SET UP A NEW ANDROID PROJECT

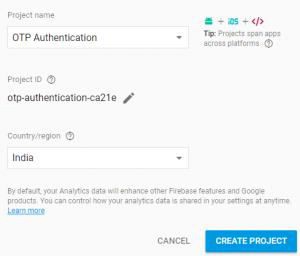
* Start Android Studio and Click on “new project”.
* Give your application a name, mine is “*OTP Authentication*”.
* click next and choose Target android device.
* next, choose an empty activity to keep things simple.
* next, name your Activity and click finish to build the project.

# STEP2: Add Firebase to your app

* Goto Firebase Console
* Login with your Gmail ID
* On the welcome screen of Firebase click on *Add project.*



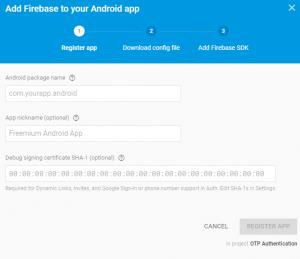
Add project window will open, there enter your project name and select your country, Project id will be automatically created and then click on *CREATE PROJECT* button.



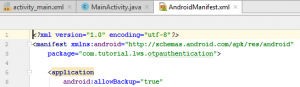
After this, your app’s dashboard will open. click on *Add Firebase to your Android app*



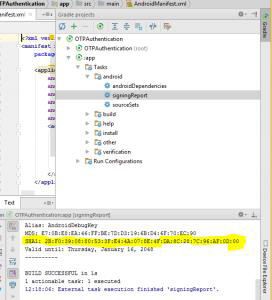
Now you have to register your app by providing the app package name and SHA-1 signing certificate on below screen.



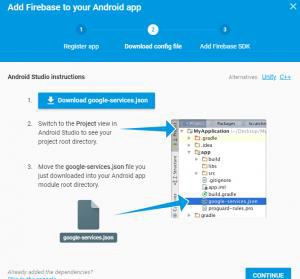
To find the App package name open AndroidManifest.xml and on the top, there is the package name



To get the SHA-1 signing certificate, click on *Gradle* tab on the right end of Android studio to open Gradle properties> then click on :*app> android>* double click on *signingReport.* In the Monitor pane, you will see SHA1.



* Copy and paste the package name and the SHA1 signing certificate in the firebase console register page. Then, click on REGISTER APP button.
* Now you have to download the google-services.json file from the download config page and paste it in app module.



After than in project pane, click on *Gradle Scripts>* open project level build.gradle file and add google service dependency in buildscript.



open app level build.gradle add firebase dependency along with firebase authentication dependency.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | | | |  |
|  | | | | |
|  |  |  |  | |

1. implementation 'com.google.firebase:firebase-core:11.8.0'



1. implementation 'com.google.firebase:firebase-auth:11.8.0'

3

4 note, for android studio below 3.0.1 use compile instead of implementation.

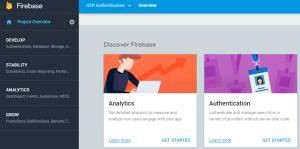
and add the plugin for google service at the bottom of the file.



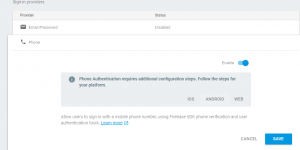
1 apply plugin: 'com.google.gms.google-services'



* Click on *Sync Now* and let the Gradle build.
* Now go back to Firebase console and click on Project Overview. In *Discover Firebase* section there is a card named Authentication, click on GET STARTED on Authentication card.



On the next page click on SETUP SIGN-IN METHOD and enable phone authentication, then SAVE it. At this point, we are done with the initial setup.



# step3: EDIT androidmanifest.XML

Our application needs internet for sending OTP. So, for this reason, we have to, first of all, give permission to our app to use the internet. Add the following line to

your *AndroidManifest.xml* inside manifest tag.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | | | |  |
|  | | | | |
|  |  |  |  | |

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



1. <manifest xmlns:android="<http://schemas.android.com/apk/res/android>"
2. package="com.androindian.fcmotp"> 4

5 <uses-permission android:name="android.permission.INTERNET"></uses-permission> 6

1. <application
2. android:allowBackup="true"
3. android:icon="@mipmap/ic\_launcher"
4. android:label="@string/app\_name"
5. android:usesCleartextTraffic="true"
6. android:roundIcon="@mipmap/ic\_launcher\_round"
7. android:supportsRtl="true"
8. android:theme="@style/AppTheme">
9. <activity android:name=".ProfileActivity"></activity>
10. <activity android:name=".VerifyPhoneActivity" />
11. <activity android:name=".MainActivity">
12. <intent-filter>
13. <action android:name="android.intent.action.MAIN" /> 20
14. <category android:name="android.intent.category.LAUNCHER" />
15. </intent-filter>
16. </activity>
17. </application> 25

26 </manifest>

activity\_main.xml

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | | | |  |
|  | | | | |
|  |  |  |  | |

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



1. <layout
2. xmlns:android=["http://schemas.android.com/apk/res/android](http://schemas.android.com/apk/res/android)"
3. xmlns:app=["http://schemas.android.com/apk/res-auto](http://schemas.android.com/apk/res-auto)"
4. xmlns:tools=["http://schemas.android.com/tools"](http://schemas.android.com/tools)
5. tools:context=".MainActivity">
6. <LinearLayout
7. android:layout\_width="match\_parent"
8. android:layout\_height="match\_parent" 10

11 android:orientation="vertical"> 12

13

14

15

1. <EditText
2. android:id="@+id/mobile"
3. android:layout\_width="match\_parent"
4. android:layout\_height="wrap\_content"
5. android:ems="10"
6. android:inputType="phone"
7. android:hint="Mobile"/> 23

24

1. <Button
2. android:id="@+id/button"
3. android:layout\_width="match\_parent"
4. android:layout\_height="wrap\_content"
5. android:text="Button" /> 30

31

32

33

1. </LinearLayout>
2. </layout>

MainActivity.Java



1 package com.androindian.fcmotp; 2

1. import androidx.appcompat.app.AppCompatActivity;
2. import androidx.databinding.DataBindingUtil; 5
3. import android.content.Intent;
4. import android.os.Bundle;
5. import android.view.View; 9

10 import com.androindian.fcmotp.databinding.ActivityMainBinding;

11

12 public class MainActivity extends AppCompatActivity { 13

14

15

16

17

18

19

20

21

22

23

24

25

26

27

28

29

30

31

32

33

34

35

36

37

38

39

40

41

42

43 }

ActivityMainBinding binding;

@Override

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

binding= DataBindingUtil.setContentView( MainActivity.this,R.layout.activity\_main);

binding.button.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View v) {

String mobile = binding.mobile.getText().toString().trim(); if(mobile.isEmpty() || mobile.length() < 10){

binding.mobile.setError("Enter a valid mobile");

binding.mobile.requestFocus(); return;

}

Intent intent = new Intent(MainActivity.this, VerifyPhoneActivity.class); intent.putExtra("mobile", mobile);

startActivity(intent);

}

});

}

VerifyPhoneActivity.Java

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | | | |  |
|  | | | | |
|  |  |  |  | |

1 package com.androindian.fcmotp; 2



1. import androidx.annotation.NonNull;
2. import androidx.appcompat.app.AppCompatActivity; 5
3. import android.content.Intent;
4. import android.os.Bundle;
5. import android.view.View;
6. import android.widget.EditText;
7. import android.widget.Toast; 11
8. import com.google.android.gms.tasks.OnCompleteListener;
9. import com.google.android.gms.tasks.Task;
10. import com.google.android.gms.tasks.TaskExecutors;
11. import com.google.android.material.snackbar.Snackbar;
12. import com.google.firebase.FirebaseException;
13. import com.google.firebase.auth.AuthResult;
14. import com.google.firebase.auth.FirebaseAuth;
15. import com.google.firebase.auth.FirebaseAuthInvalidCredentialsException;
16. import com.google.firebase.auth.PhoneAuthCredential;
17. import com.google.firebase.auth.PhoneAuthProvider; 22

23 import java.util.concurrent.TimeUnit; 24

25 public class VerifyPhoneActivity extends AppCompatActivity { 26

1. private String mVerificationId;
2. private EditText editTextCode;
3. private FirebaseAuth mAuth; 30
4. @Override
5. protected void onCreate(Bundle savedInstanceState) {
6. super.onCreate(savedInstanceState);
7. setContentView(R.layout.activity\_verify\_phone); 35
8. //initializing objects
9. mAuth = FirebaseAuth.getInstance();
10. editTextCode = findViewById(R.id.editTextCode); 39

40

1. //getting mobile number from the previous activity
2. //and sending the verification code to the number
3. Intent intent = getIntent();
4. String mobile = intent.getStringExtra("mobile");
5. sendVerificationCode(mobile); 46

47

1. //if the automatic sms detection did not work, user can also enter the code manually
2. //so adding a click listener to the button
3. findViewById(R.id.buttonSignIn).setOnClickListener(new View.OnClickListener() {
4. @Override
5. public void onClick(View v) {
6. String code = editTextCode.getText().toString().trim();
7. if (code.isEmpty() || code.length() < 6) {
8. editTextCode.setError("Enter valid code");
9. editTextCode.requestFocus();
10. return;

58 }

59

1. //verifying the code entered manually
2. verifyVerificationCode(code);

62 }

63 });

64

65 }

66

1. //the method is sending verification code
2. //the country id is concatenated
3. //you can take the country id as user input as well
4. private void sendVerificationCode(String mobile) {
5. PhoneAuthProvider.getInstance().verifyPhoneNumber( 72 "+91" + mobile,

73 60,

1. TimeUnit.SECONDS,
2. TaskExecutors.MAIN\_THREAD,
3. mCallbacks); 77 }

78

79

1. //the callback to detect the verification status
2. private PhoneAuthProvider.OnVerificationStateChangedCallbacks mCallbacks = new
3. PhoneAuthProvider.OnVerificationStateChangedCallbacks() {
4. @Override
5. public void onVerificationCompleted(PhoneAuthCredential phoneAuthCredential) { 85
6. //Getting the code sent by SMS
7. String code = phoneAuthCredential.getSmsCode(); 88
8. //sometime the code is not detected automatically
9. //in this case the code will be null
10. //so user has to manually enter the code
11. if (code != null) {

93

94

95

96

97 }

98

editTextCode.setText(code);

//verifying the code verifyVerificationCode(code);

}

99

100

101

102

103

104

105

106

107

108

109

110

111

112

113

114

115

116

117

118

119

120

121

122

123

124

125

126

127

128

129

130

131

132

133

134

135

136

137

138

139

140

141

142

143

144

145

146

147

148

149

150

151

152

153

154

155

}

@Override

public void onVerificationFailed(FirebaseException e) { Toast.makeText(VerifyPhoneActivity.this, e.getMessage(), Toast.LENGTH\_LONG).show();

}

@Override

public void onCodeSent(String s, PhoneAuthProvider.ForceResendingToken forceResendingToken) { super.onCodeSent(s, forceResendingToken);

//storing the verification id that is sent to the user mVerificationId = s;

}

};

private void verifyVerificationCode(String code) {

//creating the credential

PhoneAuthCredential credential = PhoneAuthProvider.getCredential(mVerificationId, code);

//signing the user signInWithPhoneAuthCredential(credential);

}

private void signInWithPhoneAuthCredential(PhoneAuthCredential credential) { mAuth.signInWithCredential(credential)

.addOnCompleteListener(VerifyPhoneActivity.this, new OnCompleteListener<AuthResult>() {

@Override

public void onComplete(@NonNull Task<AuthResult> task) { if (task.isSuccessful()) {

//verification successful we will start the profile activity

Intent intent = new Intent(VerifyPhoneActivity.this, ProfileActivity.class); intent.setFlags(Intent.FLAG\_ACTIVITY\_NEW\_TASK | Intent.FLAG\_ACTIVITY\_CLEAR\_TASK); startActivity(intent);

} else {

//verification unsuccessful.. display an error message

String message = "Somthing is wrong, we will fix it soon...";

if (task.getException() instanceof FirebaseAuthInvalidCredentialsException) { message = "Invalid code entered...";

}

Snackbar snackbar = Snackbar.make(findViewById(R.id.parent), message, Snackbar.LENGTH\_LONG); snackbar.setAction("Dismiss", new View.OnClickListener() {

@Override

public void onClick(View v) {

}

});

snackbar.show();

}

}

});

}

activity\_verify\_phone.xml

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | | | |  |
|  | | | | |
|  |  |  |  | |

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



1. <RelativeLayout xmlns:android=["http://schemas.android.com/apk/res/android](http://schemas.android.com/apk/res/android)"
2. xmlns:app=["http://schemas.android.com/apk/res-auto](http://schemas.android.com/apk/res-auto)"
3. xmlns:tools=["http://schemas.android.com/tools"](http://schemas.android.com/tools)
4. android:layout\_width="match\_parent"
5. android:layout\_height="match\_parent"
6. tools:context=".VerifyPhoneActivity"> 8
7. <RelativeLayout
8. android:id="@+id/relativeLayout"
9. android:layout\_width="match\_parent"
10. android:layout\_height="200dp"
11. android:background="@color/colorPrimary"
12. android:orientation="horizontal"> 15
13. <ImageView
14. android:layout\_width="120dp"
15. android:layout\_height="120dp"
16. android:layout\_centerHorizontal="true"
17. android:layout\_centerVertical="true"
18. android:background="@mipmap/ic\_launcher" /> 22

23 </RelativeLayout> 24

1. <ImageView
2. android:id="@+id/imageView"
3. android:layout\_width="match\_parent"
4. android:layout\_height="120dp"
5. android:layout\_below="@id/relativeLayout"
6. android:layout\_marginTop="-50dp"
7. android:background="@mipmap/ic\_launcher" /> 32
8. <RelativeLayout
9. android:id="@+id/container"
10. android:layout\_width="match\_parent"
11. android:layout\_height="match\_parent"
12. android:layout\_below="@id/imageView"
13. android:orientation="vertical"
14. android:padding="20dp"> 40
15. <TextView
16. android:id="@+id/textView"
17. android:layout\_width="wrap\_content"
18. android:layout\_height="wrap\_content"
19. android:layout\_centerHorizontal="true"
20. android:layout\_marginTop="25dp"
21. android:text="Wait for the code I sent You"
22. android:textAppearance="@style/Base.TextAppearance.AppCompat.Headline"
23. android:textColor="@color/colorPrimary" /> 50
24. <ProgressBar
25. android:id="@+id/progressbar"
26. android:layout\_below="@id/textView"
27. android:layout\_centerHorizontal="true"
28. android:layout\_width="wrap\_content"
29. android:layout\_height="wrap\_content" /> 57
30. <EditText
31. android:id="@+id/editTextCode"
32. android:layout\_width="200dp"
33. android:layout\_height="wrap\_content"
34. android:layout\_below="@id/progressbar"
35. android:layout\_centerHorizontal="true"
36. android:layout\_marginTop="10dp" 65 android:digits="0123456789"
37. android:drawablePadding="10dp"
38. android:hint="enter verification code"
39. android:inputType="phone"
40. android:maxLength="10" /> 70
41. <Button
42. android:id="@+id/buttonSignIn"
43. android:layout\_width="wrap\_content"
44. android:layout\_height="wrap\_content"
45. android:layout\_below="@id/editTextCode"
46. android:layout\_centerHorizontal="true"
47. android:layout\_marginTop="15dp"
48. android:background="@color/colorPrimaryDark"
49. android:text="Sign In"
50. android:textAllCaps="false"
51. android:textColor="#cdd8f1" /> 82

83 </RelativeLayout> 84

85 </RelativeLayout>

ProfileActivity.java

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | | | |  |
|  | | | | |
|  |  |  |  | |

1 package com.androindian.fcmotp; 2



3 import androidx.appcompat.app.AppCompatActivity; 4

5 import android.os.Bundle; 6

7 public class ProfileActivity extends AppCompatActivity { 8

1. @Override
2. protected void onCreate(Bundle savedInstanceState) {
3. super.onCreate(savedInstanceState);
4. setContentView(R.layout.activity\_profile); 13 }

14 }

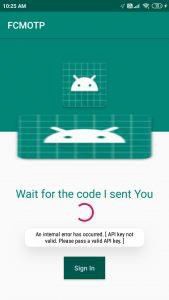
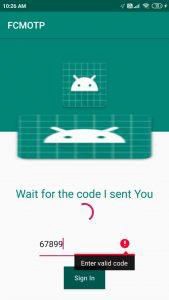
activity\_profile.xml



1. <?xml version="1.0" encoding="utf-8"?>
2. <androidx.constraintlayout.widget.ConstraintLayout xmlns:android=["http://schemas.android.com/apk/res/android](http://schemas.android.com/apk/res/android)"
3. xmlns:app=["http://schemas.android.com/apk/res-auto](http://schemas.android.com/apk/res-auto)"
4. xmlns:tools=["http://schemas.android.com/tools"](http://schemas.android.com/tools)
5. android:layout\_width="match\_parent"
6. android:layout\_height="match\_parent"
7. tools:context=".ProfileActivity"> 8

9 </androidx.constraintlayout.widget.ConstraintLayout>

**OUTPUT:**



###### RESULT:

Thus the application was created and executed successfully.

|  |  |
| --- | --- |
| **Ex: 10** | Implement Biometric authentication in Android |
| **Date:** |

###### AIM:

To Implement Biometric authentication in Android.

###### PROCEDURE:

|  |  |
| --- | --- |
| 1 | Use Android studio to create an Android application |
| 2 | Modify *src/MainActivity.java* file and add required code. |
| 3 | Modify layout XML file *res/layout/activity\_main.xml* add any GUI component if required. |
| 4 | Modify *res/values/strings.xml* to define required constant values |
| 5 | Modify *AndroidManifest.xml* as shown below |
| 6 | Run the application to launch Android emulator and verify the result of the changes done in the application. |

Mainactvity.java

import android.hardware.fingerprint.FingerprintManager; import android.os.Bundle;

import android.view.View;

import android.widget.ImageView; import android.widget.TextView;

import androidx.appcompat.app.AppCompatActivity; public class MainActivity extends AppCompatActivity {

private TextView textView; private ImageView imageView;

private FingerprintManager fingerprintManager; private FingerprintManager.AuthenticationCallback authenticationCallback;

@Override

protected void onCreate(Bundle savedInstanceState) { super.onCreate(savedInstanceState); setContentView(R.layout.activity\_main);

textView = findViewById(R.id.textView); imageView = findViewById(R.id.imageView); imageView.setImageResource(R.drawable.image1); fingerprintManager = (FingerprintManager) getSystemService(FINGERPRINT\_SERVICE);

authenticationCallback = new FingerprintManager.AuthenticationCallback() {

@Override

public void onAuthenticationError(int errorCode, CharSequence errString)

{

textView.setText(“ERROR”); imageView.setImageResource(R.drawable.image2); super.onAuthenticationError(errorCode, errString);

}

@Override

public void onAuthenticationHelp(int helpCode, CharSequence helpString) {

textView.setText(“HELP”); imageView.setImageResource(R.drawable.image1); super.onAuthenticationHelp(helpCode, helpString);

}

@Override public void

onAuthenticationSucceeded(FingerprintManager.AuthenticationResult result) {

textView.setText(“SUCCESS”); imageView.setImageResource(R.drawable.image3); super.onAuthenticationSucceeded(result);

}

@Override

public void onAuthenticationFailed() { textView.setText(“FAILED”); imageView.setImageResource(R.drawable.image2); super.onAuthenticationFailed();

}

};

}

public void scanButton(View view){ fingerprintManager.authenticate(null, null, 0, authenticationCallback, null);

}

}

Gradle File:

apply plugin: 'com.android.application' android {

compileSdkVersion 29

buildToolsVersion "29.0.2" defaultConfig {

applicationId "com.example.myfingerprintauthenticationapp" minSdkVersion 26

targetSdkVersion 29

versionCode 1

versionName "1.0"

testInstrumentationRunner "androidx.test.runner.AndroidJUnitRunner"

}

buildTypes {

release {

minifyEnabled false

proguardFiles getDefaultProguardFile('proguard-android-optimize.txt'), 'proguard-rules.pro'

}

}

}

dependencies {

implementation fileTree(dir: 'libs', include: ['\*.jar']) implementation 'androidx.appcompat:appcompat:1.1.0' implementation 'androidx.constraintlayout:constraintlayout:1.1.3' testImplementation 'junit:junit:4.12'

androidTestImplementation 'androidx.test.ext:junit:1.1.1' androidTestImplementation 'androidx.test.espresso:espresso-core:3.2.0'

}

Resource File:

<resources>

<string name="app\_name">My FingerPrintAuthentication App</string>

<string name="result\_display">Result Display ...</string>

<string name="scan">Scan</string>

<string name="todo">TODO</string>

</resources>

Layout XML File:

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

<androidx.constraintlayout.widget.ConstraintLayout xmlns:android[="http://schemas.android.co](http://schemas.android.com/apk/res/android)m/[apk/res/android](http://schemas.android.com/apk/res/android)"

xmlns:app[="http://schemas.android.com/apk/res-auto](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">

<TextView

android:id="@+id/textView" android:layout\_width="212dp" android:layout\_height="90dp" android:text="@string/result\_display" android:textAlignment="center" android:textSize="24sp" app:layout\_constraintBottom\_toBottomOf="parent" app:layout\_constraintLeft\_toLeftOf="parent" app:layout\_constraintRight\_toRightOf="parent"

app:layout\_constraintTop\_toTopOf="parent" app:layout\_constraintVertical\_bias="0.289" />

<Button

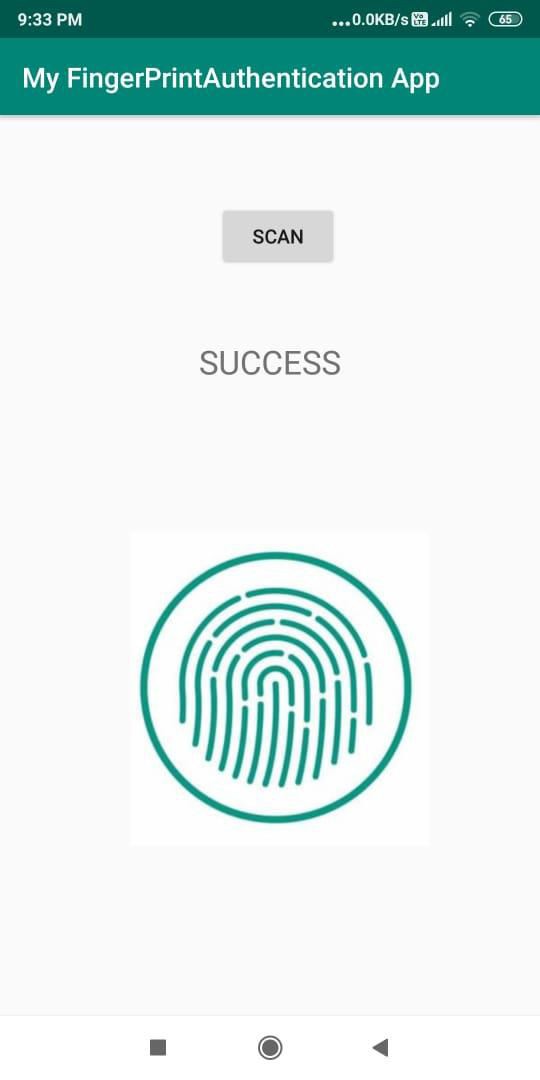
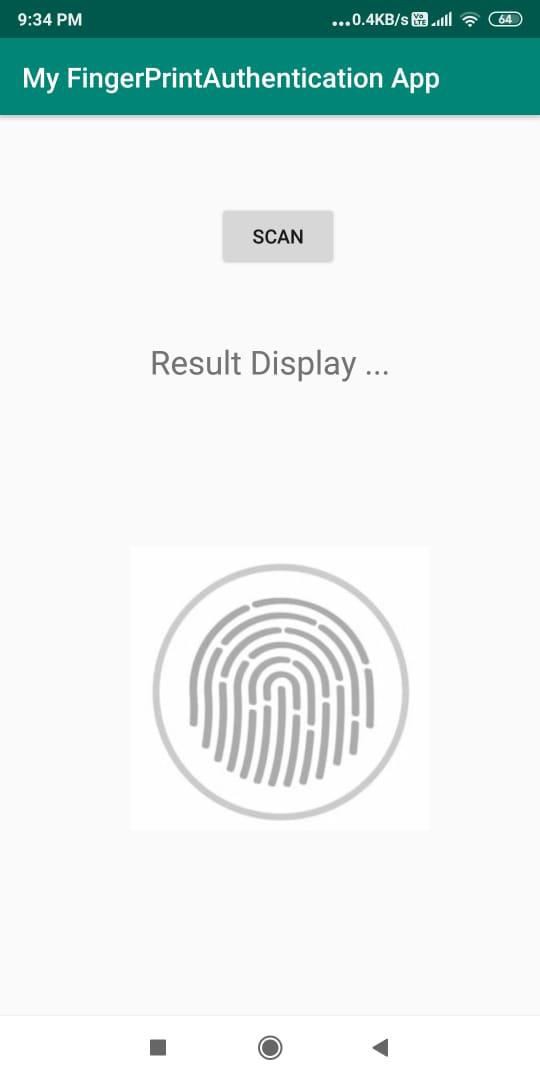
android:id="@+id/button" android:layout\_width="wrap\_content" android:layout\_height="wrap\_content" android:layout\_marginStart="158dp" android:layout\_marginTop="64dp" android:onClick="scanButton" android:text="@string/scan" app:layout\_constraintStart\_toStartOf="parent" app:layout\_constraintTop\_toTopOf="parent" />

<ImageView

android:id="@+id/imageView" android:layout\_width="217dp" android:layout\_height="266dp" android:layout\_marginStart="95dp" android:layout\_marginTop="31dp" android:contentDescription="@string/todo" app:layout\_constraintStart\_toStartOf="parent" app:layout\_constraintTop\_toBottomOf="@+id/textView" tools:srcCompat="@drawable/image1" />

</androidx.constraintlayout.widget.ConstraintLayout>

**OUTPUT:**



###### RESULT:

Thus the application was successfully created.

|  |  |
| --- | --- |
| **Ex: 11** | Develop an iOS application that uses GUI components |
| **Date:** |

###### AIM:

To develop an iOS application that uses GUI components.

###### PROCEDURE:

1. Open Xcode.
2. Go to File>New>Project.
3. Click on Single View Application and click next.
4. Name it Hello World!
5. Decide whether you want it to be an iPhone app, an iPad app, or universal
6. Make sure use storyboards and Use Automatic Reference Counting are checked
7. Click Next
8. Navigate to where you want to save the file and click create

ViewController.swift

import UIKit

class ViewController: UIViewController {

@IBOutlet weak var uiTextFieldResult: UITextField! var varNumber1 = 0

var varNumber2 = 0

var varNumberResult = 0 var varOperator = "+"

override func viewDidLoad() { super.viewDidLoad()

// Do any additional setup after loading the view, typically from a nib.

}

override func didReceiveMemoryWarning() { super.didReceiveMemoryWarning()

// Dispose of any resources that can be recreated.

}

@IBAction func button1(Sender: UIButton){ uiTextFieldResult.text = uiTextFieldResult.text! + "1"

}

@IBAction func button2(Sender: UIButton){ uiTextFieldResult.text = uiTextFieldResult.text! + "2"

}

@IBAction func button3(Sender: UIButton){ uiTextFieldResult.text = uiTextFieldResult.text! + "3"

}

@IBAction func button4(Sender: UIButton){ uiTextFieldResult.text = uiTextFieldResult.text! + "4"

}

@IBAction func button5(Sender: UIButton){ uiTextFieldResult.text = uiTextFieldResult.text! + "5"

}

@IBAction func button6(Sender: UIButton){ uiTextFieldResult.text = uiTextFieldResult.text! + "6"

}

@IBAction func button7(Sender: UIButton){ uiTextFieldResult.text = uiTextFieldResult.text! + "7"

}

@IBAction func button8(Sender: UIButton){ uiTextFieldResult.text = uiTextFieldResult.text! + "8"

@IBAction func button9(Sender: UIButton){ uiTextFieldResult.text = uiTextFieldResult.text! + "9"

}

@IBAction func button0(Sender: UIButton){ uiTextFieldResult.text = uiTextFieldResult.text! + "0"

}

@IBAction func buttonPlus(Sender: UIButton){ varOperator = "+"

varNumber1 = Int(uiTextFieldResult.text!)! clearText()

}

@IBAction func buttonMinus(Sender: UIButton){ varOperator = "-"

varNumber1 = Int(uiTextFieldResult.text!)! clearText()

}

@IBAction func buttonMultiplication(Sender: UIButton){ varOperator = "\*"

varNumber1 = Int(uiTextFieldResult.text!)! clearText()

}

@IBAction func buttonDivision(Sender: UIButton){ varOperator = "/"

varNumber1 = Int(uiTextFieldResult.text!)! clearText()

}

@IBAction func buttonEqual(Sender: UIButton){ varNumber2 = Int(uiTextFieldResult.text!)!

switch varOperator { case "+":

varNumberResult = varNumber1+varNumber2 uiTextFieldResult.text = String(varNumberResult)

case "-":

varNumberResult = varNumber1-varNumber2 uiTextFieldResult.text = String(varNumberResult)

case "\*":

varNumberResult = varNumber1\*varNumber2 uiTextFieldResult.text = String(varNumberResult)

case "/":

varNumberResult = varNumber1/varNumber2 uiTextFieldResult.text = String(varNumberResult)

default:

uiTextFieldResult.text = "ERROR"

}

@IBAction func buttonClear(Sender: UIButton){ clearText()

}

func clearText() { uiTextFieldResult.text = ""

}

}

**OUTPUT:**

A screenshot of a phone

Description automatically generated with low confidence

**RESULT:**

Thus the program was successfully executed.

|  |  |
| --- | --- |
| **Ex: 12** | Develop an iOS application to demonstrate the use of imageview |
| **Date:** |

**AIM**

To develop an iOS application to demonstrate the use of imageview

###### PROCEDURE:

1. Open Xcode.
2. Go to File>New>Project.
3. Click on Single View Application and click next.
4. Name it Hello World!
5. Decide whether you want it to be an iPhone app, an iPad app, or universal
6. Make sure use storyboards and Use Automatic Reference Counting are checked
7. Click Next
8. Navigate to where you want to save the file and click create

###### ViewController.swift

import UIKit

import PlaygroundSupport

class MyViewController : UIViewController { override func viewDidLoad() {

super.viewDidLoad()

let view = UIView() view.backgroundColor = .white

let catImage = UIImage(named: "cat.jpg") let myImageView:UIImageView = UIImageView()

myImageView.contentMode = UIView.ContentMode.scaleAspectFit

myImageView.frame.size.width = 200

myImageView.frame.size.height = 200 myImageView.center = self.view.center

myImageView.image = catImage view.addSubview(myImageView)

self.view = view

}

}

###### OUTPUT:

A black and white photo of a cat

Description automatically generated with medium confidence

**RESULT:**

Thus the program was successfully executed.