**Lab 9: Authentication (Step by step)**

1. **Authentication with Phone number**

Go to this link to create project in Firebase  
<https://console.firebase.google.com/u/0/?pli=1>

A screenshot of a computer

Description automatically generated

Create a project name, then accept all checkboxes and continue in the next steps

A computer screen shot of a person standing in front of a white background

Description automatically generated

Click **Project Overview** to go to your project you created

A screenshot of a computer

Description automatically generated

Scroll down and click **Authentication**

A screenshot of a computer

Description automatically generated

Click button **Get Started**

A screenshot of a computer

Description automatically generated

Choose sign-in method **Phone** in **Native providers**

A screenshot of a computer

Description automatically generated

**Enable** phone sign-in method

A screenshot of a computer

Description automatically generated

Click on optional in the bottom when you want to test (input fake phone number and fake validation code you want to use when you test). If your Firebase’s account is free, sometime it will not smooth or have some troubles, so that you can think about this solution.

A screenshot of a computer

Description automatically generated

After that, the screen will be like this.

A screenshot of a computer

Description automatically generated

Back to the Project Overview. Then click into the Android button to go to next steps

A screenshot of a computer

Description automatically generated

Input the **Android package name** based on the namespace in your project (you can skip 2 options below).

A screenshot of a computer

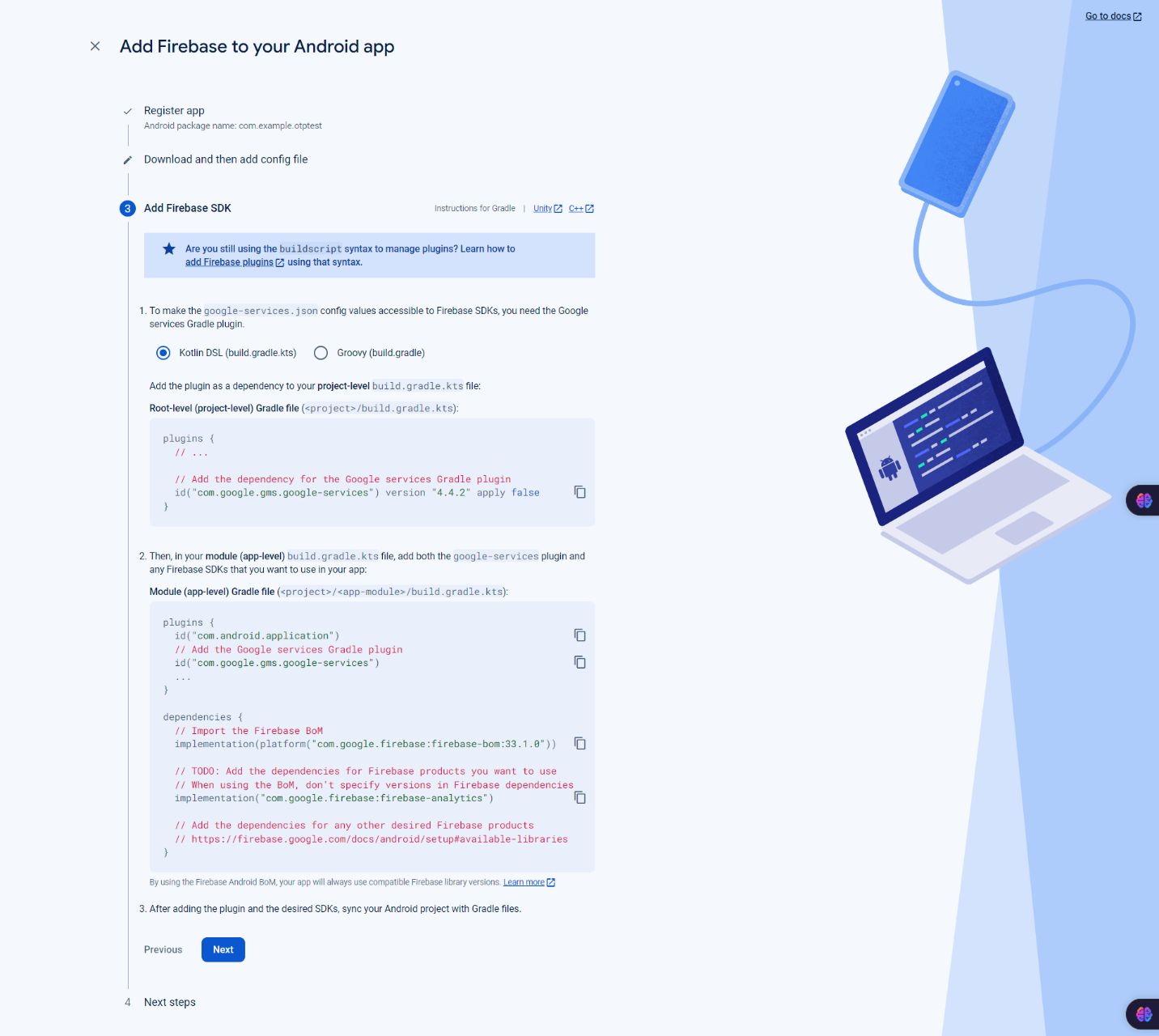
Description automatically generated

A screenshot of a computer program

Description automatically generated

Download **google-services.json** and go to next stepA screenshot of a computer

Description automatically generated



Follow all the instructions in steps 3 and 4

A screenshot of a computer

Description automatically generated

Go to your project and open terminal. Then type ./gradlew signingReport

Back to your console.firebase, click into **Add fingerprint** and paste values of SHA1 and SHA-256

A screenshot of a computer

Description automatically generated

Go to the **activity\_main.xml** file and refer to the following code. Below is the code for the **activity\_main.xml** file.

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

<**RelativeLayout**

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

    xmlns:tools="<http://schemas.android.com/tools>"

    android:layout\_width="match\_parent"

    android:layout\_height="match\_parent"

    tools:context=".MainActivity">

    <!--Edittext for getting users phone number-->

    <**EditText**

        android:id="@+id/idEdtPhoneNumber"

        android:layout\_width="match\_parent"

        android:layout\_height="wrap\_content"

        android:layout\_centerHorizontal="true"

        android:layout\_margin="10dp"

        android:hint="Enter your phone"

        android:importantForAutofill="no"

        android:inputType="phone" />

    <!--Button for getting OTP-->

    <**Button**

        android:id="@+id/idBtnGetOtp"

        android:layout\_width="match\_parent"

        android:layout\_height="wrap\_content"

        android:layout\_below="@id/idEdtPhoneNumber"

        android:layout\_margin="10dp"

        android:text="Get OTP"

        android:textAllCaps="false" />

    <!--Edittext for getting otp from user-->

    <**EditText**

        android:id="@+id/idEdtOtp"

        android:layout\_width="match\_parent"

        android:layout\_height="wrap\_content"

        android:layout\_below="@id/idBtnGetOtp"

        android:layout\_margin="10dp"

        android:hint="Enter OTP"

        android:importantForAutofill="no"

        android:inputType="phone" />

    <!--button for verifying user OTP-->

    <**Button**

        android:id="@+id/idBtnVerify"

        android:layout\_width="match\_parent"

        android:layout\_height="wrap\_content"

        android:layout\_below="@id/idEdtOtp"

        android:layout\_margin="10dp"

        android:text="Verify OTP"

        android:textAllCaps="false" />

</**RelativeLayout**>

Navigate to the **app > AndroidManifest.xml** file and add the below permissions to it.

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

<**uses-permission** android:name="android.permission.ACCESS\_NETWORK\_STATE" />

Navigate to the**app > java > your app’s package name > Right-click on your app’s package name and click on New > Activity > Empty Activity** and name your activity. Here we have given it a name as **HomeActivity**.

Go to the **MainActivity.java** file and refer to the following code. Below is the code for the **MainActivity.java** file. Comments are added inside the code to understand the code in more detail.

**import** android.content.Intent;

**import** android.os.Bundle;

**import** android.text.TextUtils;

**import** android.view.View;

**import** android.widget.Button;

**import** android.widget.EditText;

**import** android.widget.Toast;

**import** androidx.annotation.NonNull;

**import** androidx.appcompat.app.AppCompatActivity;

**import** com.google.android.gms.tasks.OnCompleteListener;

**import** com.google.android.gms.tasks.Task;

**import** com.google.android.gms.tasks.TaskExecutors;

**import** com.google.firebase.FirebaseException;

**import** com.google.firebase.auth.AuthResult;

**import** com.google.firebase.auth.FirebaseAuth;

**import** com.google.firebase.auth.PhoneAuthCredential;

**import** com.google.firebase.auth.PhoneAuthProvider;

**import** java.util.concurrent.TimeUnit;

**public** **class** MainActivity **extends** AppCompatActivity {

    // variable for FirebaseAuth class

**private** FirebaseAuth mAuth;

    // variable for our text input

    // field for phone and OTP.

**private** EditText edtPhone, edtOTP;

    // buttons for generating OTP and verifying OTP

**private** Button verifyOTPBtn, generateOTPBtn;

    // string for storing our verification ID

**private** String verificationId;

    @Override

**protected** **void** onCreate(Bundle savedInstanceState) {

**super**.onCreate(savedInstanceState);

        setContentView(R.layout.activity\_main);

        // below line is for getting instance

        // of our FirebaseAuth.

        mAuth = FirebaseAuth.getInstance();

        // initializing variables for button and Edittext.

        edtPhone = findViewById(R.id.idEdtPhoneNumber);

        edtOTP = findViewById(R.id.idEdtOtp);

        verifyOTPBtn = findViewById(R.id.idBtnVerify);

        generateOTPBtn = findViewById(R.id.idBtnGetOtp);

        // setting onclick listener for generate OTP button.

        generateOTPBtn.setOnClickListener(**new** View.OnClickListener() {

            @Override

**public** **void** onClick(View v) {

                // below line is for checking whether the user

                // has entered his mobile number or not.

**if** (TextUtils.isEmpty(edtPhone.getText().toString())) {

                    // when mobile number text field is empty

                    // displaying a toast message.

                    Toast.makeText(MainActivity.**this**, "Please enter a valid phone number.", Toast.LENGTH\_SHORT).show();

                } **else** {

                    // if the text field is not empty we are calling our

                    // send OTP method for getting OTP from Firebase.

                    String phone = "+91" + edtPhone.getText().toString();

                    sendVerificationCode(phone);

                }

            }

        });

        // initializing on click listener

        // for verify otp button

        verifyOTPBtn.setOnClickListener(**new** View.OnClickListener() {

            @Override

**public** **void** onClick(View v) {

                // validating if the OTP text field is empty or not.

**if** (TextUtils.isEmpty(edtOTP.getText().toString())) {

                    // if the OTP text field is empty display

                    // a message to user to enter OTP

                    Toast.makeText(MainActivity.**this**, "Please enter OTP", Toast.LENGTH\_SHORT).show();

                } **else** {

                    // if OTP field is not empty calling

                    // method to verify the OTP.

                    verifyCode(edtOTP.getText().toString());

                }

            }

        });

    }

**private** **void** signInWithCredential(PhoneAuthCredential credential) {

        // inside this method we are checking if

        // the code entered is correct or not.

        mAuth.signInWithCredential(credential)

                .addOnCompleteListener(**new** OnCompleteListener<AuthResult>() {

                    @Override

**public** **void** onComplete(@NonNull Task<AuthResult> task) {

**if** (task.isSuccessful()) {

                            // if the code is correct and the task is successful

                            // we are sending our user to new activity.

                            Intent i = **new** Intent(MainActivity.**this**, HomeActivity.**class**);

                            startActivity(i);

                            finish();

                        } **else** {

                            // if the code is not correct then we are

                            // displaying an error message to the user.

                            Toast.makeText(MainActivity.**this**, task.getException().getMessage(), Toast.LENGTH\_LONG).show();

                        }

                    }

                });

    }

**private** **void** sendVerificationCode(String number) {

        // this method is used for getting

        // OTP on user phone number.

        PhoneAuthOptions options =

                PhoneAuthOptions.newBuilder(mAuth)

                        .setPhoneNumber(number)            // Phone number to verify

                        .setTimeout(60L, TimeUnit.SECONDS) // Timeout and unit

                        .setActivity(**this**)                 // Activity (for callback binding)

                        .setCallbacks(mCallBack)           // OnVerificationStateChangedCallbacks

                        .build();

        PhoneAuthProvider.verifyPhoneNumber(options);

    }

    // callback method is called on Phone auth provider.

**private** PhoneAuthProvider.OnVerificationStateChangedCallbacks

            // initializing our callbacks for on

            // verification callback method.

            mCallBack = **new** PhoneAuthProvider.OnVerificationStateChangedCallbacks() {

        // below method is used when

        // OTP is sent from Firebase

        @Override

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

**super**.onCodeSent(s, forceResendingToken);

            // when we receive the OTP it

            // contains a unique id which

            // we are storing in our string

            // which we have already created.

            verificationId = s;

        }

        // this method is called when user

        // receive OTP from Firebase.

        @Override

**public** **void** onVerificationCompleted(PhoneAuthCredential phoneAuthCredential) {

            // below line is used for getting OTP code

            // which is sent in phone auth credentials.

**final** String code = phoneAuthCredential.getSmsCode();

            // checking if the code

            // is null or not.

**if** (code != **null**) {

                // if the code is not null then

                // we are setting that code to

                // our OTP edittext field.

                edtOTP.setText(code);

                // after setting this code

                // to OTP edittext field we

                // are calling our verifycode method.

                verifyCode(code);

            }

        }

        // this method is called when firebase doesn't

        // sends our OTP code due to any error or issue.

        @Override

**public** **void** onVerificationFailed(FirebaseException e) {

            // displaying error message with firebase exception.

            Toast.makeText(MainActivity.**this**, e.getMessage(), Toast.LENGTH\_LONG).show();

        }

    };

    // below method is use to verify code from Firebase.

**private** **void** verifyCode(String code) {

        // below line is used for getting

        // credentials from our verification id and code.

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

        // after getting credential we are

        // calling sign in method.

        signInWithCredential(credential);

    }

}

Now we have authenticated our user and move towards our Home Activity. Now we will display a welcome message to our user on successful authentication. For this Navigate to the**app > res > layout > activity\_home.xml** and add the below code to it.

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

<**RelativeLayout**

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

    xmlns:tools="<http://schemas.android.com/tools>"

    android:layout\_width="match\_parent"

    android:layout\_height="match\_parent"

    tools:context=".HomeActivity">

    <**TextView**

        android:layout\_width="match\_parent"

        android:layout\_height="wrap\_content"

        android:layout\_centerInParent="true"

        android:padding="10dp"

        android:text="Geeks for Geeks \n Welcome to Home Screen"

        android:textAlignment="center"

        android:textColor="@color/purple\_500"

        android:textSize="20sp" />

</**RelativeLayout**>

For enabling Phone authentication in the Firebase console go to the [Firebase Console](https://firebase.google.com/). Now click on **Go to Console** option and navigate to your project. After that click on your project. You can get to see the below screen.

A screenshot of a computer

Description automatically generated

Go to **Authentication** 🡪 **Sign-In method** 🡪 Click on **Phone** in Provider

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

Click to **Enable** button and then **Save**

1. **Authentication with Email**

Go to this link to create project in Firebase

<https://console.firebase.google.com/u/0/?pli=1>

A screenshot of a computer

Description automatically generated

Create a project name, then accept all checkboxes and continue in the next steps

A computer screen shot of a person standing in front of a white background

Description automatically generated

Click **Project Overview** to go to your project you created   
A screenshot of a computer

Description automatically generated



Scroll down and click **Authentication**

A screenshot of a computer

Description automatically generated



Click button **Get Started**

A screenshot of a computer

Description automatically generated



Choose sign-in method **Email/Password** in **Native providers**

A screenshot of a computer

Description automatically generated

**Enable** Email/Password sign-in method

A screenshot of a computer

Description automatically generated

After that, the screen will be like this.

A screenshot of a computer

Description automatically generated

Back to the Project Overview. Then click into the Android button to go to next steps

A screenshot of a computer

Description automatically generated

Input the **Android package name** based on the namespace in your project (you can skip 2 options below).

![A screenshot of a computer

Description automatically generated](data:image/png;base64,iVBORw0KGgoAAAANSUhEUgAAAAEAAAABCAYAAAAfFcSJAAAADUlEQVR4XmP4//8vAwAI+AL8ldyzEQAAAABJRU5ErkJggg==)

A screenshot of a computer program

Description automatically generated

Download **google-services.json** and go to next step  
![A screenshot of a computer

Description automatically generated](data:image/png;base64,iVBORw0KGgoAAAANSUhEUgAAAAEAAAABCAYAAAAfFcSJAAAADUlEQVR4XmP4//8PAwAI9gL7zpsyJAAAAABJRU5ErkJggg==)

A screenshot of a computer

Description automatically generated



A screenshot of a computer

Description automatically generated

Follow all the instructions in steps 3 and 4

![A screenshot of a computer

Description automatically generated](data:image/png;base64,iVBORw0KGgoAAAANSUhEUgAAAAEAAAABCAYAAAAfFcSJAAAADUlEQVR4XmP4//8XAwAI8gL5c60pfQAAAABJRU5ErkJggg==)

Go to your project and open terminal. Then type ./gradlew signingReport

Back to your console.firebase, click into **Add fingerprint** and paste values of SHA1 and SHA-256

![A screenshot of a computer

Description automatically generated](data:image/png;base64,iVBORw0KGgoAAAANSUhEUgAAAAEAAAABCAYAAAAfFcSJAAAADUlEQVR4XmP4//87AwAI7AL2/bfpfgAAAABJRU5ErkJggg==)



Add libraries:

Add this in build.gradle.kts(Picture below)

id("com.google.gms.google-services") version "4.4.2" apply **false**

A screenshot of a computer

Description automatically generated





Add this dependencies in build.gradle.kts(Picture below)

id("com.google.gms.google-services")

implementation(platform("com.google.firebase:firebase-bom:33.1.0"))  
implementation("com.google.firebase:firebase-auth")  
implementation("com.google.android.gms:play-services-auth:21.2.0")

A computer screen shot of a program

Description automatically generated







Add resource to drawable from source code provide.

Go to the **activity\_main.xml** file and refer to the following code. Below is the code for the **activity\_main.xml** file.  
<?xml version="1.0" encoding="utf-8"?>  
<**androidx.constraintlayout.widget.ConstraintLayout** 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:id="@+id/main"  
 android:background="@drawable/background\_logo"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 tools:context=".MainActivity">  
  
 <**EditText** android:id="@+id/editTextTextEmailAddress"  
 android:layout\_width="250dp"  
 android:layout\_height="45dp"  
 android:layout\_weight="4"  
 android:background="@drawable/edittext\_background"  
 android:drawablePadding="10dp"  
 android:ems="10"  
 android:hint="Email"  
 android:inputType="text"  
 android:padding="10dp"  
 android:textColor="#FF000000"  
 android:textColorHint="#9AA0B4"  
 android:textSize="15sp"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toTopOf="@+id/guideline30" />  
  
 <**EditText** android:id="@+id/editTextPassword"  
 android:layout\_width="250dp"  
 android:layout\_height="45dp"  
 android:layout\_weight="4"  
 android:background="@drawable/edittext\_background"  
 android:drawablePadding="10dp"  
 android:ems="10"  
 android:layout\_marginTop="30dp"  
 android:hint="Password"  
 android:inputType="textPassword"  
 android:padding="10dp"  
 android:textColor="#FF000000"  
 android:textColorHint="#9AA0B4"  
 android:textSize="15sp"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toBottomOf="@+id/editTextTextEmailAddress" />  
  
 <**androidx.appcompat.widget.AppCompatButton** android:id="@+id/btnRegister"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_gravity="center\_horizontal"  
 android:background="@drawable/button\_login\_background"  
 android:paddingStart="80dp"  
 android:paddingEnd="80dp"  
 android:layout\_marginTop="30dp"  
 android:text="Xác nhận"  
 android:textAllCaps="false"  
 android:textColor="#FFFFFF"  
 android:textSize="20sp"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toBottomOf="@+id/editTextPassword" />  
  
 <**androidx.constraintlayout.widget.Guideline** android:id="@+id/guideline30"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:orientation="horizontal"  
 app:layout\_constraintGuide\_percent="0.35" />  
  
</**androidx.constraintlayout.widget.ConstraintLayout**>

Navigate to the **app > AndroidManifest.xml** file and add the below permissions to it.

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

<**uses-permission** android:name="android.permission.ACCESS\_NETWORK\_STATE" />

Go to the **MainActivity.java** file and refer to the following code. Below is the code for the **MainActivity.java** file. Comments are added inside the code to understand the code in more detail.

**package** com.fu.vfoody;  
  
**import** android.content.Intent;  
**import** android.os.Bundle;  
**import** android.widget.EditText;  
**import** android.widget.Toast;  
  
**import** androidx.activity.EdgeToEdge;  
**import** androidx.appcompat.app.AppCompatActivity;  
**import** androidx.appcompat.widget.AppCompatButton;  
**import** androidx.core.graphics.Insets;  
**import** androidx.core.view.ViewCompat;  
**import** androidx.core.view.WindowInsetsCompat;  
  
**import** com.google.firebase.auth.FirebaseAuth;  
**import** com.google.firebase.auth.FirebaseUser;  
  
**import** java.util.concurrent.ThreadLocalRandom;  
  
**public class** MainActivity **extends** AppCompatActivity {  
  
 **private** AppCompatButton btnRegister;  
 **private** EditText editTextTextEmailAddress, editTextPassword;  
  
 **private** FirebaseAuth mAuth;  
  
 @Override  
 **protected void** onCreate(Bundle savedInstanceState) {  
 **super**.onCreate(savedInstanceState);  
 EdgeToEdge.enable(**this**);  
 setContentView(R.layout.activity\_main);  
 init();  
  
 btnRegister.setOnClickListener(v -> {  
 signUpUser(editTextTextEmailAddress.getText().toString(), editTextPassword.getText().toString());  
 });  
 }  
  
 **private void** signUpUser(String email, String password) {  
 mAuth.createUserWithEmailAndPassword(email, password)  
 .addOnCompleteListener(**this**, task -> {  
 **if** (task.isSuccessful()) {  
 FirebaseUser user = mAuth.getCurrentUser();  
 **if** (user != **null**) {  
 user.sendEmailVerification()  
 .addOnCompleteListener(emailTask -> {  
 **if** (emailTask.isSuccessful()) {  
 //Process with backend  
 Toast.makeText(MainActivity.**this**, "Successfully to send verification email.", Toast.LENGTH\_SHORT).show();  
 } **else** {  
 Toast.makeText(MainActivity.**this**, "Failed to send verification email.", Toast.LENGTH\_SHORT).show();  
 }  
 });  
 }  
 } **else** {  
 Toast.makeText(MainActivity.**this**, "Authentication failed.", Toast.LENGTH\_SHORT).show();  
 }  
 });  
 }  
  
 **private void** init() {  
 mAuth = FirebaseAuth.getInstance();  
 editTextTextEmailAddress = findViewById(R.id.editTextTextEmailAddress);  
 editTextPassword = findViewById(R.id.editTextPassword);  
 btnRegister = findViewById(R.id.btnRegister);  
 }  
}