1. **XML Layout (activity\_phone\_verification.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:orientation="vertical"

android:padding="16dp"

android:gravity="center">

<!-- EditText for Phone Number -->

<EditText

android:id="@+id/phone\_number\_edit\_text"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:hint="Phone number"

android:inputType="phone"

android:maxLength="15"

android:padding="16dp"

android:textSize="16sp" />

<!-- Button to send verification code -->

<Button

android:id="@+id/send\_verification\_code\_button"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:text="Send Verification Code"

android:textSize="16sp"

android:layout\_marginTop="16dp" />

<!-- EditText for the verification code -->

<EditText

android:id="@+id/verification\_code\_edit\_text"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:hint="Verification code"

android:inputType="number"

android:maxLength="6"

android:padding="16dp"

android:textSize="16sp"

android:layout\_marginTop="16dp" />

<!-- Button to verify the code -->

<Button

android:id="@+id/verify\_code\_button"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:text="Verify Code"

android:textSize="16sp"

android:layout\_marginTop="16dp" />

</LinearLayout>

1. **Java Code (PhoneVerificationActivity.java)**

package com.example.phoneauth;

import android.os.Bundle;

import android.util.Log;

import android.widget.Button;

import android.widget.EditText;

import android.widget.Toast;

import androidx.appcompat.app.AppCompatActivity;

import com.google.firebase.auth.FirebaseAuth;

import com.google.firebase.auth.FirebaseUser;

import com.google.firebase.auth.PhoneAuthCredential;

import com.google.firebase.auth.PhoneAuthProvider;

import java.util.concurrent.TimeUnit;

public class PhoneVerificationActivity extends AppCompatActivity {

private static final String TAG = "PhoneVerification";

private FirebaseAuth mAuth;

private EditText phoneNumberEditText, verificationCodeEditText;

private Button sendVerificationCodeButton, verifyCodeButton;

private String verificationId;

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_phone\_verification);

// Initialize FirebaseAuth instance

mAuth = FirebaseAuth.getInstance();

// Find views

phoneNumberEditText = findViewById(R.id.phone\_number\_edit\_text);

verificationCodeEditText = findViewById(R.id.verification\_code\_edit\_text);

sendVerificationCodeButton = findViewById(R.id.send\_verification\_code\_button);

verifyCodeButton = findViewById(R.id.verify\_code\_button);

// On click listener for Send Verification Code button

sendVerificationCodeButton.setOnClickListener(v -> {

String phoneNumber = phoneNumberEditText.getText().toString();

if (!phoneNumber.isEmpty()) {

sendVerificationCode(phoneNumber);

} else {

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

}

});

// On click listener for Verify Code button

verifyCodeButton.setOnClickListener(v -> {

String code = verificationCodeEditText.getText().toString();

if (!code.isEmpty() && verificationId != null) {

verifyCode(code);

} else {

Toast.makeText(this, "Please enter the verification code", Toast.LENGTH\_SHORT).show();

}

});

}

// Method to send the verification code to the phone number

private void sendVerificationCode(String phoneNumber) {

PhoneAuthProvider.getInstance().verifyPhoneNumber(

phoneNumber, // Phone number to verify

60, // Timeout duration (in seconds)

TimeUnit.SECONDS, // Time unit for the timeout duration

this, // Activity context

mCallbacks); // Callbacks for verification events

}

// Callbacks for phone number verification process

private PhoneAuthProvider.OnVerificationStateChangedCallbacks mCallbacks =

new PhoneAuthProvider.OnVerificationStateChangedCallbacks() {

@Override

public void onVerificationCompleted(PhoneAuthCredential credential) {

// Automatically verify the phone number

signInWithPhoneAuthCredential(credential);

}

@Override

public void onVerificationFailed(com.google.firebase.FirebaseException e) {

// Handle verification failure

Log.e(TAG, "Verification failed", e);

Toast.makeText(PhoneVerificationActivity.this, "Verification failed: " + e.getMessage(), Toast.LENGTH\_SHORT).show();

}

@Override

public void onCodeSent(String verificationId, PhoneAuthProvider.ForceResendingToken token) {

// Save verification ID and resending token for later use

PhoneVerificationActivity.this.verificationId = verificationId;

Toast.makeText(PhoneVerificationActivity.this, "Verification code sent", Toast.LENGTH\_SHORT).show();

}

};

// Method to verify the code entered by the user

private void verifyCode(String code) {

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

signInWithPhoneAuthCredential(credential);

}

// Method to sign in with the phone authentication credential

private void signInWithPhoneAuthCredential(PhoneAuthCredential credential) {

mAuth.signInWithCredential(credential)

.addOnCompleteListener(this, task -> {

if (task.isSuccessful()) {

FirebaseUser user = task.getResult().getUser();

// Successfully signed in

Toast.makeText(PhoneVerificationActivity.this, "Verification successful!", Toast.LENGTH\_SHORT).show();

// Proceed with the next activity

// Intent intent = new Intent(PhoneVerificationActivity.this, NextActivity.class);

// startActivity(intent);

} else {

// Sign-in failed, display a message

Toast.makeText(PhoneVerificationActivity.this, "Verification failed.", Toast.LENGTH\_SHORT).show();

}

});

}

}