package com.appdroid.tourstravelsproject.User;  
  
import androidx.annotation.NonNull;  
import androidx.appcompat.app.AppCompatActivity;  
  
import android.app.ProgressDialog;  
import android.content.Intent;  
import android.os.Bundle;  
import android.util.Log;  
import android.view.View;  
import android.widget.Button;  
import android.widget.EditText;  
  
import com.appdroid.tourstravelsproject.Admin.AddPackage;  
import com.appdroid.tourstravelsproject.Holder.UserInfo;  
import com.appdroid.tourstravelsproject.R;  
import com.google.firebase.FirebaseException;  
import com.google.firebase.FirebaseTooManyRequestsException;  
import com.google.firebase.auth.FirebaseAuth;  
import com.google.firebase.auth.FirebaseAuthInvalidCredentialsException;  
import com.google.firebase.auth.FirebaseUser;  
import com.google.firebase.auth.PhoneAuthCredential;  
import com.google.firebase.auth.PhoneAuthOptions;  
import com.google.firebase.auth.PhoneAuthProvider;  
  
import java.util.concurrent.TimeUnit;  
  
public class UserLogin extends AppCompatActivity {  
  
 EditText contact;  
 Button sendOtp;  
 UserInfo userInfo;  
 FirebaseAuth mAuth = FirebaseAuth.*getInstance*();  
 private String mVerificationId;  
 private PhoneAuthProvider.ForceResendingToken mResendToken;  
  
 ProgressDialog progressDialog;  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*activity\_user\_login*);  
  
 progressDialog = new ProgressDialog(UserLogin.this);  
 progressDialog.setCanceledOnTouchOutside(false );  
  
 contact = findViewById(R.id.*contact*);  
 sendOtp = findViewById(R.id.*sendOTP*);  
  
 userInfo = (UserInfo) getIntent().getSerializableExtra("userData");  
  
 contact.setText(userInfo.getMobileNo());  
  
 sendOtp.setOnClickListener(new View.OnClickListener() {  
 @Override  
 public void onClick(View v) {  
 getVerificationCode("+91"+userInfo.getMobileNo());  
 }  
 });  
 }  
  
 public void getVerificationCode(String mobileNo){  
 progressDialog.show();  
 PhoneAuthOptions options =  
 PhoneAuthOptions.*newBuilder*(mAuth)  
 .setPhoneNumber(mobileNo) *// Phone number to verify* .setTimeout(60L, TimeUnit.*SECONDS*) *// Timeout and unit* .setActivity(this) *// Activity (for callback binding)* .setCallbacks(mCallbacks) *// OnVerificationStateChangedCallbacks* .build();  
 PhoneAuthProvider.*verifyPhoneNumber*(options);  
 }  
 PhoneAuthProvider.OnVerificationStateChangedCallbacks mCallbacks = new PhoneAuthProvider.OnVerificationStateChangedCallbacks() {  
  
 @Override  
 public void onVerificationCompleted(PhoneAuthCredential credential) {  
 *// This callback will be invoked in two situations:  
 // 1 - Instant verification. In some cases the phone number can be instantly  
 // verified without needing to send or enter a verification code.  
 // 2 - Auto-retrieval. On some devices Google Play services can automatically  
 // detect the incoming verification SMS and perform verification without  
 // user action.* Log.*d*("TAG", "onVerificationCompleted:" + credential);  
  
 *//signInWithPhoneAuthCredential(credential);* }  
  
 @Override  
 public void onVerificationFailed(FirebaseException e) {  
 *// This callback is invoked in an invalid request for verification is made,  
 // for instance if the the phone number format is not valid.* Log.*w*("TAG", "onVerificationFailed", e);  
  
 if (e instanceof FirebaseAuthInvalidCredentialsException) {  
 *// Invalid request* } else if (e instanceof FirebaseTooManyRequestsException) {  
 *// The SMS quota for the project has been exceeded* }  
  
 *// Show a message and update the UI* }  
  
 @Override  
 public void onCodeSent(@NonNull String verificationId,  
 @NonNull PhoneAuthProvider.ForceResendingToken token) {  
 *// The SMS verification code has been sent to the provided phone number, we  
 // now need to ask the user to enter the code and then construct a credential  
 // by combining the code with a verification ID.* Log.*d*("TAG", "onCodeSent:" + verificationId);  
  
 *// Save verification ID and resending token so we can use them later* mVerificationId = verificationId;  
 mResendToken = token;  
  
 Intent i = new Intent(getApplicationContext(), VerifyOTP.class);  
 i.putExtra("code",verificationId);  
 i.putExtra("userInfo",userInfo);  
 startActivity(i);  
 finish();  
 progressDialog.dismiss();  
 }  
 };  
  
 @Override  
 public void onStart() {  
 super.onStart();  
 *// Check if user is signed in (non-null) and update UI accordingly.* FirebaseUser currentUser = mAuth.getCurrentUser();  
 if(currentUser != null){  
 startActivity(new Intent(getApplicationContext(), UserDashboard.class));  
 finish();  
 }  
 }  
  
}