dependencies:

firebase\_core: latest\_version

firebase\_auth: latest\_version

provider: latest\_version

import 'package:firebase\_auth/firebase\_auth.dart';

class AuthService {

final FirebaseAuth \_auth = FirebaseAuth.instance;

Future<void> sendOTP(String phoneNumber) async {

await \_auth.verifyPhoneNumber(

phoneNumber: phoneNumber,

verificationCompleted: (PhoneAuthCredential credential) async {

await \_auth.signInWithCredential(credential);

},

verificationFailed: (FirebaseAuthException e) {

print("Verification failed: ${e.message}");

},

codeSent: (String verificationId, int? resendToken) {

print("OTP sent to $phoneNumber");

},

codeAutoRetrievalTimeout: (String verificationId) {},

);

}

Future<User?> verifyOTP(String verificationId, String otp) async {

try {

PhoneAuthCredential credential = PhoneAuthProvider.credential(

verificationId: verificationId,

smsCode: otp,

);

UserCredential userCredential = await \_auth.signInWithCredential(credential);

return userCredential.user;

} catch (e) {

print("Error verifying OTP: $e");

return null;

}

}

}

dependencies:

firebase\_core: latest\_version

firebase\_auth: latest\_version

cloud\_firestore: latest\_version

provider: latest\_version

google\_maps\_flutter: latest\_version # For location-based services

import 'package:firebase\_auth/firebase\_auth.dart';

import 'package:cloud\_firestore/cloud\_firestore.dart';

class AuthService {

final FirebaseAuth \_auth = FirebaseAuth.instance;

final FirebaseFirestore \_firestore = FirebaseFirestore.instance;

Future<void> sendOTP(String phoneNumber, Function(String) codeSentCallback) async {

await \_auth.verifyPhoneNumber(

phoneNumber: phoneNumber,

verificationCompleted: (PhoneAuthCredential credential) async {

await \_auth.signInWithCredential(credential);

},

verificationFailed: (FirebaseAuthException e) {

print("Verification failed: ${e.message}");

},

codeSent: (String verificationId, int? resendToken) {

codeSentCallback(verificationId);

},

codeAutoRetrievalTimeout: (String verificationId) {},

);

}

Future<User?> verifyOTP(String verificationId, String otp) async {

try {

PhoneAuthCredential credential = PhoneAuthProvider.credential(

verificationId: verificationId,

smsCode: otp,

);

UserCredential userCredential = await \_auth.signInWithCredential(credential);

return userCredential.user;

} catch (e) {

print("Error verifying OTP: $e");

return null;

}

}

Future<void> createUserProfile(String uid, String role, String phone) async {

await \_firestore.collection('users').doc(uid).set({

'phone': phone,

'role': role,

'createdAt': DateTime.now(),

});

}

}

import 'package:flutter/material.dart';

import 'auth\_service.dart';

class OTPLoginScreen extends StatefulWidget {

@override

\_OTPLoginScreenState createState() => \_OTPLoginScreenState();

}

class \_OTPLoginScreenState extends State<OTPLoginScreen> {

final TextEditingController phoneController = TextEditingController();

final TextEditingController otpController = TextEditingController();

String? verificationId;

bool otpSent = false;

final AuthService authService = AuthService();

void sendOTP() {

authService.sendOTP(phoneController.text, (String verId) {

setState(() {

verificationId = verId;

otpSent = true;

});

});

}

void verifyOTP() async {

if (verificationId != null) {

var user = await authService.verifyOTP(verificationId!, otpController.text);

if (user != null) {

print("User logged in: ${user.uid}");

} else {

print("OTP verification failed.");

}

}

}

@override

Widget build(BuildContext context) {

return Scaffold(

appBar: AppBar(title: Text("OTP Login")),

body: Padding(

padding: const EdgeInsets.all(20.0),

child: Column(

children: [

TextField(

controller: phoneController,

keyboardType: TextInputType.phone,

decoration: InputDecoration(labelText: "Phone Number"),

),

SizedBox(height: 20),

otpSent

? TextField(

controller: otpController,

keyboardType: TextInputType.number,

decoration: InputDecoration(labelText: "Enter OTP"),

)

: Container(),

SizedBox(height: 20),

ElevatedButton(

onPressed: otpSent ? verifyOTP : sendOTP,

child: Text(otpSent ? "Verify OTP" : "Send OTP"),

),

],

),

),

);

}

}