class RoleService {

static const String customer = "customer";

static const String worker = "worker";

static const String admin = "admin";

}

void checkUserRole(User user) async {

var userData = await FirebaseFirestore.instance.collection('users').doc(user.uid).get();

if (userData.exists) {

String role = userData.data()?['role'];

if (role == RoleService.customer) {

Navigator.pushReplacementNamed(context, '/customer\_home');

} else if (role == RoleService.worker) {

Navigator.pushReplacementNamed(context, '/worker\_home');

} else if (role == RoleService.admin) {

Navigator.pushReplacementNamed(context, '/admin\_panel');

}

}

}

dependencies:

firebase\_core: latest\_version

firebase\_auth: latest\_version

cloud\_firestore: latest\_version

firebase\_storage: latest\_version # For uploading profile pictures

image\_picker: latest\_version # For picking images from gallery/camera

class UserModel {

String uid;

String name;

String email;

String phone;

String role; // customer, worker, admin

String profilePic;

List<String> serviceHistory;

UserModel({

required this.uid,

required this.name,

required this.email,

required this.phone,

required this.role,

required this.profilePic,

required this.serviceHistory,

});

factory UserModel.fromMap(Map<String, dynamic> data) {

return UserModel(

uid: data['uid'] ?? '',

name: data['name'] ?? '',

email: data['email'] ?? '',

phone: data['phone'] ?? '',

role: data['role'] ?? 'customer',

profilePic: data['profilePic'] ?? '',

serviceHistory: List<String>.from(data['serviceHistory'] ?? []),

);

}

Map<String, dynamic> toMap() {

return {

'uid': uid,

'name': name,

'email': email,

'phone': phone,

'role': role,

'profilePic': profilePic,

'serviceHistory': serviceHistory,

};

}

}

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

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

import 'user\_model.dart';

class UserService {

final FirebaseFirestore \_firestore = FirebaseFirestore.instance;

final FirebaseAuth \_auth = FirebaseAuth.instance;

// Fetch user profile

Future<UserModel?> getUserProfile(String uid) async {

try {

var userDoc = await \_firestore.collection('users').doc(uid).get();

if (userDoc.exists) {

return UserModel.fromMap(userDoc.data()!);

}

} catch (e) {

print("Error fetching user profile: $e");

}

return null;

}

// Update user profile

Future<void> updateUserProfile(UserModel user) async {

try {

await \_firestore.collection('users').doc(user.uid).update(user.toMap());

} catch (e) {

print("Error updating profile: $e");

}

}

}