//DBhelper

import 'package:sqflite/sqflite.dart';

import 'package:path/path.dart';

class User {

final String email;

final String password;

final String? name;

final String? phone;

final String? jobRole;

User({

required this.email,

required this.password,

this.name,

this.phone,

this.jobRole,

});

Map<String, dynamic> toMap() {

return {

'email': email,

'password': password,

'name': name ?? '',

'phone': phone ?? '',

'job\_role': jobRole ?? '',

};

}

static User fromMap(Map<String, dynamic> map) {

return User(

email: map['email'],

password: map['password'],

name: map['name'],

phone: map['phone'],

jobRole: map['job\_role'],

);

}

}

class DatabaseHelper {

static final DatabaseHelper instance = DatabaseHelper.\_init();

static Database? \_database;

DatabaseHelper.\_init();

Future<Database> get database async {

if (\_database != null) return \_database!;

\_database = await \_initDB('immodrops.db');

return \_database!;

}

Future<Database> \_initDB(String filePath) async {

final dbPath = await getDatabasesPath();

final path = join(dbPath, filePath);

return await openDatabase(

path,

version: 1,

onCreate: \_createDB,

);

}

Future \_createDB(Database db, int version) async {

await db.execute('''

CREATE TABLE users (

email TEXT PRIMARY KEY,

password TEXT NOT NULL,

name TEXT,

phone TEXT,

job\_role TEXT

)

''');

}

Future<void> insertUser(User user) async {

final db = await database;

await db.insert(

'users',

user.toMap(),

conflictAlgorithm: ConflictAlgorithm.replace,

);

}

Future<User?> getUser(String email) async {

final db = await database;

final List<Map<String, Object?>> maps = await db.query(

'users',

where: 'email = ?',

whereArgs: [email],

);

if (maps.isEmpty) {

return null;

}

return User.fromMap(maps.first);

}

Future<bool> validateUser(String email, String password) async {

final db = await database;

final List<Map<String, Object?>> result = await db.query(

'users',

where: 'email = ? AND password = ?',

whereArgs: [email, password],

);

return result.isNotEmpty;

}

Future<void> updateUser(User user) async {

final db = await database;

await db.update(

'users',

user.toMap(),

where: 'email = ?',

whereArgs: [user.email],

);

}

Future close() async {

final db = await database;

db.close();

}

}

//profilescreen

import 'package:flutter/material.dart';

import 'database\_helper.dart';

class ProfileScreen extends StatefulWidget {

final String email;

ProfileScreen({required this.email});

@override

\_ProfileScreenState createState() => \_ProfileScreenState();

}

class \_ProfileScreenState extends State<ProfileScreen> {

User? \_user;

bool \_isLoading = true;

final \_formKey = GlobalKey<FormState>();

final TextEditingController \_nameController = TextEditingController();

final TextEditingController \_phoneController = TextEditingController();

final TextEditingController \_jobRoleController = TextEditingController();

@override

void initState() {

super.initState();

\_loadUserData();

}

Future<void> \_loadUserData() async {

final user = await DatabaseHelper.instance.getUser(widget.email);

setState(() {

\_user = user;

if (user != null) {

\_nameController.text = user.name ?? '';

\_phoneController.text = user.phone ?? '';

\_jobRoleController.text = user.jobRole ?? '';

}

\_isLoading = false;

});

}

Future<void> \_updateProfile() async {

if (\_formKey.currentState!.validate() && \_user != null) {

final updatedUser = User(

email: \_user!.email,

password: \_user!.password,

name: \_nameController.text.trim(),

phone: \_phoneController.text.trim(),

jobRole: \_jobRoleController.text.trim(),

);

await DatabaseHelper.instance.updateUser(updatedUser);

ScaffoldMessenger.of(context).showSnackBar(

SnackBar(content: Text('Profile updated successfully')),

);

setState(() {

\_user = updatedUser;

});

}

}

@override

Widget build(BuildContext context) {

return Scaffold(

appBar: AppBar(

title: Text('My Profile'),

backgroundColor: Colors.pinkAccent,

),

body: \_isLoading

? Center(child: CircularProgressIndicator())

: SingleChildScrollView(

padding: EdgeInsets.all(16),

child: Column(

crossAxisAlignment: CrossAxisAlignment.start,

children: [

*// User info card*

Card(

elevation: 4,

shape: RoundedRectangleBorder(

borderRadius: BorderRadius.circular(12),

),

child: Padding(

padding: const EdgeInsets.all(16),

child: Column(

crossAxisAlignment: CrossAxisAlignment.start,

children: [

Center(

child: CircleAvatar(

radius: 50,

backgroundColor: Colors.pink[100],

child: Icon(Icons.person, size: 60, color: Colors.pink),

),

),

SizedBox(height: 16),

Center(

child: Text(

\_user?.name ?? 'Update your profile',

style: TextStyle(

fontSize: 24,

fontWeight: FontWeight.bold,

),

),

),

SizedBox(height: 8),

Center(

child: Text(

\_user?.email ?? '',

style: TextStyle(

fontSize: 16,

color: Colors.grey[600],

),

),

),

SizedBox(height: 16),

Divider(),

SizedBox(height: 16),

*// User details table*

Table(

columnWidths: {

0: FlexColumnWidth(1),

1: FlexColumnWidth(2),

},

children: [

TableRow(

children: [

Padding(

padding: const EdgeInsets.symmetric(vertical: 8),

child: Text(

'Email:',

style: TextStyle(fontWeight: FontWeight.bold),

),

),

Padding(

padding: const EdgeInsets.symmetric(vertical: 8),

child: Text(\_user?.email ?? ''),

),

],

),

TableRow(

children: [

Padding(

padding: const EdgeInsets.symmetric(vertical: 8),

child: Text(

'Name:',

style: TextStyle(fontWeight: FontWeight.bold),

),

),

Padding(

padding: const EdgeInsets.symmetric(vertical: 8),

child: Text(\_user?.name ?? 'Not provided'),

),

],

),

TableRow(

children: [

Padding(

padding: const EdgeInsets.symmetric(vertical: 8),

child: Text(

'Phone:',

style: TextStyle(fontWeight: FontWeight.bold),

),

),

Padding(

padding: const EdgeInsets.symmetric(vertical: 8),

child: Text(\_user?.phone ?? 'Not provided'),

),

],

),

TableRow(

children: [

Padding(

padding: const EdgeInsets.symmetric(vertical: 8),

child: Text(

'Job Role:',

style: TextStyle(fontWeight: FontWeight.bold),

),

),

Padding(

padding: const EdgeInsets.symmetric(vertical: 8),

child: Text(\_user?.jobRole ?? 'Not provided'),

),

],

),

],

),

],

),

),

),

SizedBox(height: 24),

*// Edit profile form*

Card(

elevation: 4,

shape: RoundedRectangleBorder(

borderRadius: BorderRadius.circular(12),

),

child: Padding(

padding: const EdgeInsets.all(16),

child: Form(

key: \_formKey,

child: Column(

crossAxisAlignment: CrossAxisAlignment.start,

children: [

Text(

'Edit Profile',

style: TextStyle(

fontSize: 20,

fontWeight: FontWeight.bold,

),

),

SizedBox(height: 16),

TextFormField(

controller: \_nameController,

decoration: InputDecoration(

labelText: 'Name',

border: OutlineInputBorder(),

prefixIcon: Icon(Icons.person),

),

validator: (value) {

if (value == null || value.isEmpty) {

return 'Please enter your name';

}

return null;

},

),

SizedBox(height: 16),

TextFormField(

controller: \_phoneController,

decoration: InputDecoration(

labelText: 'Phone Number',

border: OutlineInputBorder(),

prefixIcon: Icon(Icons.phone),

),

keyboardType: TextInputType.phone,

validator: (value) {

if (value == null || value.isEmpty) {

return 'Please enter your phone number';

}

return null;

},

),

SizedBox(height: 16),

TextFormField(

controller: \_jobRoleController,

decoration: InputDecoration(

labelText: 'Job Role',

border: OutlineInputBorder(),

prefixIcon: Icon(Icons.work),

),

validator: (value) {

if (value == null || value.isEmpty) {

return 'Please enter your job role';

}

return null;

},

),

SizedBox(height: 24),

Center(

child: ElevatedButton(

onPressed: \_updateProfile,

style: ElevatedButton.styleFrom(

backgroundColor: Colors.pinkAccent,

padding: EdgeInsets.symmetric(horizontal: 60, vertical: 14),

shape: RoundedRectangleBorder(borderRadius: BorderRadius.circular(8)),

),

child: Text(

'Update Profile',

style: TextStyle(fontSize: 16, fontWeight: FontWeight.bold, color: Colors.white),

),

),

),

],

),

),

),

),

],

),

),

);

}

}

//homescreen

import 'package:flutter/material.dart';

import 'package:shared\_preferences/shared\_preferences.dart';

import 'database\_helper.dart';

import 'profile\_screen.dart';

class HomeScreen extends StatefulWidget {

@override

\_HomeScreenState createState() => \_HomeScreenState();

}

class \_HomeScreenState extends State<HomeScreen> {

String? \_loggedInEmail;

User? \_user;

bool \_isLoading = true;

@override

void initState() {

super.initState();

\_loadUserData();

}

Future<void> \_loadUserData() async {

final prefs = await SharedPreferences.getInstance();

final email = prefs.getString('loggedInEmail');

if (email != null) {

final user = await DatabaseHelper.instance.getUser(email);

setState(() {

\_loggedInEmail = email;

\_user = user;

\_isLoading = false;

});

} else {

Navigator.pushReplacementNamed(context, '/login');

}

}

Future<void> \_logout(BuildContext context) async {

final prefs = await SharedPreferences.getInstance();

await prefs.remove('loggedInEmail');

Navigator.pushReplacementNamed(context, '/login');

}

@override

Widget build(BuildContext context) {

return Scaffold(

backgroundColor: Colors.pink[50],

appBar: AppBar(

title: Text('ImmoDrops'),

backgroundColor: Colors.pinkAccent,

actions: [

IconButton(

icon: Icon(Icons.logout),

onPressed: () => \_logout(context),

tooltip: 'Logout',

),

],

),

drawer: Drawer(

child: \_isLoading

? Center(child: CircularProgressIndicator())

: ListView(

padding: EdgeInsets.zero,

children: [

UserAccountsDrawerHeader(

decoration: BoxDecoration(

color: Colors.pinkAccent,

),

accountName: Text(\_user?.name ?? 'User'),

accountEmail: Text(\_loggedInEmail ?? ''),

currentAccountPicture: CircleAvatar(

backgroundColor: Colors.white,

child: Text(

(\_user?.name?.isNotEmpty == true)

? \_user!.name![0].toUpperCase()

: 'U',

style: TextStyle(fontSize: 40.0, color: Colors.pinkAccent),

),

),

),

ListTile(

leading: Icon(Icons.home),

title: Text('Home'),

onTap: () {

Navigator.pop(context);

},

),

ListTile(

leading: Icon(Icons.person),

title: Text('Profile'),

onTap: () {

Navigator.pop(context);

if (\_loggedInEmail != null) {

Navigator.push(

context,

MaterialPageRoute(

builder: (context) => ProfileScreen(email: \_loggedInEmail!),

),

);

}

},

),

ListTile(

leading: Icon(Icons.settings),

title: Text('Settings'),

onTap: () {

Navigator.pop(context);

*// Navigate to settings page*

},

),

Divider(),

ListTile(

leading: Icon(Icons.exit\_to\_app),

title: Text('Logout'),

onTap: () => \_logout(context),

),

],

),

),

body: \_isLoading

? Center(child: CircularProgressIndicator())

: Column(

children: [

*// Welcome card*

Padding(

padding: const EdgeInsets.all(16.0),

child: Card(

elevation: 4,

shape: RoundedRectangleBorder(

borderRadius: BorderRadius.circular(15),

),

child: Padding(

padding: const EdgeInsets.all(16.0),

child: Column(

crossAxisAlignment: CrossAxisAlignment.start,

children: [

Text(

'Welcome, ${\_user?.name ?? 'User'}!',

style: TextStyle(

fontSize: 22,

fontWeight: FontWeight.bold,

),

),

SizedBox(height: 10),

Text(

'Tap on the menu to access your profile.',

style: TextStyle(

fontSize: 16,

color: Colors.grey[600],

),

),

],

),

),

),

),

*// User info card*

if (\_user != null)

Padding(

padding: const EdgeInsets.symmetric(horizontal: 16.0),

child: InkWell(

onTap: () {

if (\_loggedInEmail != null) {

Navigator.push(

context,

MaterialPageRoute(

builder: (context) => ProfileScreen(email: \_loggedInEmail!),

),

);

}

},

child: Card(

elevation: 4,

shape: RoundedRectangleBorder(

borderRadius: BorderRadius.circular(15),

),

child: Padding(

padding: const EdgeInsets.all(16.0),

child: Column(

crossAxisAlignment: CrossAxisAlignment.start,

children: [

Text(

'Your Details',

style: TextStyle(

fontSize: 18,

fontWeight: FontWeight.bold,

),

),

SizedBox(height: 16),

Table(

columnWidths: {

0: FlexColumnWidth(1),

1: FlexColumnWidth(2),

},

children: [

TableRow(

children: [

Padding(

padding: const EdgeInsets.symmetric(vertical: 8),

child: Text(

'Email:',

style: TextStyle(fontWeight: FontWeight.bold),

),

),

Padding(

padding: const EdgeInsets.symmetric(vertical: 8),

child: Text(\_user?.email ?? ''),

),

],

),

TableRow(

children: [

Padding(

padding: const EdgeInsets.symmetric(vertical: 8),

child: Text(

'Name:',

style: TextStyle(fontWeight: FontWeight.bold),

),

),

Padding(

padding: const EdgeInsets.symmetric(vertical: 8),

child: Text(\_user?.name ?? 'Not provided'),

),

],

),

TableRow(

children: [

Padding(

padding: const EdgeInsets.symmetric(vertical: 8),

child: Text(

'Phone:',

style: TextStyle(fontWeight: FontWeight.bold),

),

),

Padding(

padding: const EdgeInsets.symmetric(vertical: 8),

child: Text(\_user?.phone ?? 'Not provided'),

),

],

),

TableRow(

children: [

Padding(

padding: const EdgeInsets.symmetric(vertical: 8),

child: Text(

'Job Role:',

style: TextStyle(fontWeight: FontWeight.bold),

),

),

Padding(

padding: const EdgeInsets.symmetric(vertical: 8),

child: Text(\_user?.jobRole ?? 'Not provided'),

),

],

),

],

),

SizedBox(height: 16),

Center(

child: TextButton.icon(

icon: Icon(Icons.edit, color: Colors.pinkAccent),

label: Text(

'Edit Profile',

style: TextStyle(

color: Colors.pinkAccent,

fontWeight: FontWeight.bold,

),

),

onPressed: () {

if (\_loggedInEmail != null) {

Navigator.push(

context,

MaterialPageRoute(

builder: (context) => ProfileScreen(email: \_loggedInEmail!),

),

);

}

},

),

),

],

),

),

),

),

),

],

),

);

}

}

//loginscreen

import 'package:flutter/material.dart';

import 'package:shared\_preferences/shared\_preferences.dart';

import 'database\_helper.dart';

class LoginScreen extends StatelessWidget {

final TextEditingController \_emailController = TextEditingController();

final TextEditingController \_passwordController = TextEditingController();

bool isValidEmail(String email) {

final emailRegex = RegExp(r'^[^@]+@[^@]+\.[^@]+');

return emailRegex.hasMatch(email);

}

bool isValidPassword(String password) {

return password.length >= 6 && password.length <= 10;

}

Future<void> \_login(BuildContext context) async {

final email = \_emailController.text.trim();

final password = \_passwordController.text;

if (!isValidEmail(email)) {

ScaffoldMessenger.of(context).showSnackBar(

SnackBar(content: Text('Please enter a valid email address')),

);

return;

}

if (!isValidPassword(password)) {

ScaffoldMessenger.of(context).showSnackBar(

SnackBar(content: Text('Password must be 6-10 characters long')),

);

return;

}

final isValid = await DatabaseHelper.instance.validateUser(email, password);

if (isValid) {

final prefs = await SharedPreferences.getInstance();

await prefs.setString('loggedInEmail', email);

Navigator.pushReplacementNamed(context, '/home');

} else {

ScaffoldMessenger.of(context).showSnackBar(

SnackBar(content: Text('Invalid email or password')),

);

}

}

@override

Widget build(BuildContext context) {

return Scaffold(

backgroundColor: Colors.pinkAccent,

body: SingleChildScrollView(

child: Column(

children: [

SizedBox(height: 80),

Center(

child: Text(

'ImmoDrops',

style: TextStyle(

color: Colors.white,

fontSize: 32,

fontWeight: FontWeight.bold,

letterSpacing: 1.2,

),

),

),

SizedBox(height: 30),

Container(

width: double.infinity,

padding: EdgeInsets.symmetric(horizontal: 24, vertical: 32),

margin: EdgeInsets.symmetric(horizontal: 20),

decoration: BoxDecoration(

color: Colors.white,

borderRadius: BorderRadius.vertical(top: Radius.circular(30)),

),

child: Column(

crossAxisAlignment: CrossAxisAlignment.start,

children: [

Text("Sign In", style: TextStyle(fontSize: 22, fontWeight: FontWeight.bold)),

SizedBox(height: 20),

Text("Email", style: TextStyle(fontWeight: FontWeight.bold)),

TextField(

controller: \_emailController,

decoration: InputDecoration(

hintText: "Enter your email",

border: UnderlineInputBorder(),

),

),

SizedBox(height: 16),

Text("Password", style: TextStyle(fontWeight: FontWeight.bold)),

TextField(

controller: \_passwordController,

obscureText: true,

decoration: InputDecoration(

hintText: "Enter password",

border: UnderlineInputBorder(),

),

),

SizedBox(height: 10),

Align(

alignment: Alignment.centerRight,

child: TextButton(

onPressed: () {},

child: Text("Forgot password?", style: TextStyle(color: Colors.black54)),

),

),

SizedBox(height: 10),

Center(

child: ElevatedButton(

onPressed: () => \_login(context),

style: ElevatedButton.styleFrom(

backgroundColor: Colors.pinkAccent,

padding: EdgeInsets.symmetric(horizontal: 60, vertical: 14),

shape: RoundedRectangleBorder(borderRadius: BorderRadius.circular(8)),

),

child: Text("Sign In", style: TextStyle(fontSize: 16, fontWeight: FontWeight.bold, color: Colors.white)),

),

),

SizedBox(height: 20),

Center(

child: Row(

mainAxisAlignment: MainAxisAlignment.center,

children: [

Text("Don't have an account? "),

GestureDetector(

onTap: () => Navigator.pushReplacementNamed(context, '/signup'),

child: Text("Sign Up", style: TextStyle(color: Colors.pinkAccent, fontWeight: FontWeight.bold)),

),

],

),

),

SizedBox(height: 20),

Divider(),

SizedBox(height: 10),

Center(child: Text("OR")),

SizedBox(height: 10),

Row(

mainAxisAlignment: MainAxisAlignment.spaceEvenly,

children: [

Icon(Icons.facebook, size: 30, color: Colors.black),

Icon(Icons.g\_mobiledata, size: 30, color: Colors.black),

Icon(Icons.alternate\_email, size: 30, color: Colors.black),

],

),

],

),

),

],

),

),

);

}

}

//signup

import 'package:flutter/material.dart';

import 'package:sqflite/sqflite.dart';

import 'splash\_screen.dart';

import 'login\_screen.dart';

import 'signup\_screen.dart';

import 'home\_screen.dart';

import 'profile\_screen.dart';

void main() async {

*// Ensure Flutter is initialized before accessing platform-specific features*

WidgetsFlutterBinding.ensureInitialized();

runApp(const MyApp());

}

class MyApp extends StatelessWidget {

const MyApp({super.key});

@override

Widget build(BuildContext context) {

return MaterialApp(

title: 'ImmoDrops',

debugShowCheckedModeBanner: false,

theme: ThemeData(

primarySwatch: Colors.pink,

primaryColor: Colors.pinkAccent,

appBarTheme: AppBarTheme(

backgroundColor: Colors.pinkAccent,

foregroundColor: Colors.white,

elevation: 4,

),

elevatedButtonTheme: ElevatedButtonThemeData(

style: ElevatedButton.styleFrom(

backgroundColor: Colors.pinkAccent,

foregroundColor: Colors.white,

),

),

),

initialRoute: '/splash',

routes: {

'/splash': (\_) => SplashScreen(),

'/login': (\_) => LoginScreen(),

'/signup': (\_) => SignupScreen(),

'/home': (\_) => HomeScreen(),

},

*// For profile screen, we need to pass email parameter so we'll handle it in onGenerateRoute*

onGenerateRoute: (settings) {

if (settings.name == '/profile') {

final args = settings.arguments as String;

return MaterialPageRoute(

builder: (context) => ProfileScreen(email: args),

);

}

return null;

},

);

}

}

//splash

import 'package:flutter/material.dart';

import 'package:shared\_preferences/shared\_preferences.dart';

import 'dart:async';

import 'database\_helper.dart';

class SplashScreen extends StatefulWidget {

@override

\_SplashScreenState createState() => \_SplashScreenState();

}

class \_SplashScreenState extends State<SplashScreen> {

@override

void initState() {

super.initState();

\_setupDatabase();

}

Future<void> \_setupDatabase() async {

*// Initialize the database*

await DatabaseHelper.instance.database;

Timer(Duration(seconds: 2), () async {

final prefs = await SharedPreferences.getInstance();

final loggedInEmail = prefs.getString('loggedInEmail');

if (loggedInEmail != null) {

Navigator.pushReplacementNamed(context, '/home');

} else {

Navigator.pushReplacementNamed(context, '/login');

}

});

}

@override

Widget build(BuildContext context) {

return Scaffold(

backgroundColor: Colors.pinkAccent,

body: Center(

child: Column(

mainAxisAlignment: MainAxisAlignment.center,

children: [

Text(

'ImmoDrops',

style: TextStyle(

color: Colors.white,

fontSize: 36,

fontWeight: FontWeight.bold,

),

),

SizedBox(height: 24),

CircularProgressIndicator(

color: Colors.white,

),

],

),

),

);

}

}

//main

import 'package:flutter/material.dart';

import 'package:sqflite/sqflite.dart';

import 'splash\_screen.dart';

import 'login\_screen.dart';

import 'signup\_screen.dart';

import 'home\_screen.dart';

import 'profile\_screen.dart';

void main() async {

*// Ensure Flutter is initialized before accessing platform-specific features*

WidgetsFlutterBinding.ensureInitialized();

runApp(const MyApp());

}

class MyApp extends StatelessWidget {

const MyApp({super.key});

@override

Widget build(BuildContext context) {

return MaterialApp(

title: 'ImmoDrops',

debugShowCheckedModeBanner: false,

theme: ThemeData(

primarySwatch: Colors.pink,

primaryColor: Colors.pinkAccent,

appBarTheme: AppBarTheme(

backgroundColor: Colors.pinkAccent,

foregroundColor: Colors.white,

elevation: 4,

),

elevatedButtonTheme: ElevatedButtonThemeData(

style: ElevatedButton.styleFrom(

backgroundColor: Colors.pinkAccent,

foregroundColor: Colors.white,

),

),

),

initialRoute: '/splash',

routes: {

'/splash': (\_) => SplashScreen(),

'/login': (\_) => LoginScreen(),

'/signup': (\_) => SignupScreen(),

'/home': (\_) => HomeScreen(),

},

*// For profile screen, we need to pass email parameter so we'll handle it in onGenerateRoute*

onGenerateRoute: (settings) {

if (settings.name == '/profile') {

final args = settings.arguments as String;

return MaterialPageRoute(

builder: (context) => ProfileScreen(email: args),

);

}

return null;

},

);

}

}

//pubspec.yaml

name: immodrops

description: A Flutter application for ImmoDrops.

publish\_to: 'none'

version: 1.0.0+1

environment:

sdk: '>=3.0.0 <4.0.0'

dependencies:

flutter:

sdk: flutter

cupertino\_icons: ^1.0.2

shared\_preferences: ^2.2.0

sqflite: ^2.3.0

path: ^1.8.3

dev\_dependencies:

flutter\_test:

sdk: flutter

flutter\_lints: ^2.0.0

flutter:

uses-material-design: true