import 'package:flutter/material.dart';

import 'package:sqflite/sqflite.dart';

import 'package:path/path.dart';

// --- Modelo Planeta ---

class Planeta {

int? id;

String nome;

double distanciaSol;

double tamanho;

String? apelido;

Planeta({

this.id,

required this.nome,

required this.distanciaSol,

required this.tamanho,

this.apelido,

});

Map<String, dynamic> toMap() {

return {

'id': id,

'nome': nome,

'distancia\_sol': distanciaSol,

'tamanho': tamanho,

'apelido': apelido,

};

}

factory Planeta.fromMap(Map<String, dynamic> map) {

return Planeta(

id: map['id'],

nome: map['nome'],

distanciaSol: map['distancia\_sol'],

tamanho: map['tamanho'],

apelido: map['apelido'],

);

}

}

// --- Banco de Dados ---

class DatabaseHelper {

static final DatabaseHelper \_instance = DatabaseHelper.\_internal();

factory DatabaseHelper() => \_instance;

DatabaseHelper.\_internal();

late Database \_database;

Future<Database> get database async {

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

\_database = await \_initDatabase();

return \_database;

}

Future<Database> \_initDatabase() async {

String path = join(await getDatabasesPath(), 'planetas.db');

return await openDatabase(path, onCreate: (db, version) {

return db.execute(

'CREATE TABLE planetas(id INTEGER PRIMARY KEY, nome TEXT, distancia\_sol REAL, tamanho REAL, apelido TEXT)',

);

}, version: 1);

}

Future<int> insertPlaneta(Planeta planeta) async {

final db = await database;

return await db.insert('planetas', planeta.toMap());

}

Future<List<Planeta>> getPlanetas() async {

final db = await database;

final List<Map<String, dynamic>> maps = await db.query('planetas');

return List.generate(maps.length, (i) {

return Planeta.fromMap(maps[i]);

});

}

Future<int> updatePlaneta(Planeta planeta) async {

final db = await database;

return await db.update(

'planetas',

planeta.toMap(),

where: 'id = ?',

whereArgs: [planeta.id],

);

}

Future<int> deletePlaneta(int id) async {

final db = await database;

return await db.delete(

'planetas',

where: 'id = ?',

whereArgs: [id],

);

}

}

// --- Tela de Listagem de Planetas ---

class ListPlanetasPage extends StatelessWidget {

final DatabaseHelper dbHelper = DatabaseHelper();

@override

Widget build(BuildContext context) {

return Scaffold(

appBar: AppBar(title: Text('Planetas')),

body: FutureBuilder<List<Planeta>>(

future: dbHelper.getPlanetas(),

builder: (context, snapshot) {

if (!snapshot.hasData) {

return Center(child: CircularProgressIndicator());

}

List<Planeta> planetas = snapshot.data!;

return ListView.builder(

itemCount: planetas.length,

itemBuilder: (context, index) {

final planeta = planetas[index];

return ListTile(

title: Text(planeta.nome),

subtitle: Text('Distância: ${planeta.distanciaSol} UA'),

onTap: () {

Navigator.push(

context,

MaterialPageRoute(

builder: (context) => DetalhesPlanetaPage(planeta),

),

);

},

trailing: IconButton(

icon: Icon(Icons.delete),

onPressed: () async {

await dbHelper.deletePlaneta(planeta.id!);

ScaffoldMessenger.of(context).showSnackBar(

SnackBar(content: Text('Planeta excluído!')),

);

(context as Element).markNeedsBuild();

},

),

);

},

);

},

),

floatingActionButton: FloatingActionButton(

onPressed: () {

Navigator.push(

context,

MaterialPageRoute(

builder: (context) => CadastroPlanetaPage(),

),

);

},

child: Icon(Icons.add),

),

);

}

}

// --- Tela de Cadastro de Planetas ---

class CadastroPlanetaPage extends StatefulWidget {

@override

\_CadastroPlanetaPageState createState() => \_CadastroPlanetaPageState();

}

class \_CadastroPlanetaPageState extends State<CadastroPlanetaPage> {

final DatabaseHelper dbHelper = DatabaseHelper();

final \_formKey = GlobalKey<FormState>();

final \_nomeController = TextEditingController();

final \_distanciaController = TextEditingController();

final \_tamanhoController = TextEditingController();

final \_apelidoController = TextEditingController();

@override

Widget build(BuildContext context) {

return Scaffold(

appBar: AppBar(title: Text('Cadastro de Planeta')),

body: Padding(

padding: const EdgeInsets.all(16.0),

child: Form(

key: \_formKey,

child: Column(

children: [

TextFormField(

controller: \_nomeController,

decoration: InputDecoration(labelText: 'Nome do Planeta'),

validator: (value) {

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

return 'Nome é obrigatório';

}

return null;

},

),

TextFormField(

controller: \_distanciaController,

decoration: InputDecoration(labelText: 'Distância do Sol (UA)'),

keyboardType: TextInputType.number,

validator: (value) {

if (value == null || value.isEmpty || double.tryParse(value) == null || double.parse(value) <= 0) {

return 'Distância deve ser um número positivo';

}

return null;

},

),

TextFormField(

controller: \_tamanhoController,

decoration: InputDecoration(labelText: 'Tamanho (Km)'),

keyboardType: TextInputType.number,

validator: (value) {

if (value == null || value.isEmpty || double.tryParse(value) == null || double.parse(value) <= 0) {

return 'Tamanho deve ser um número positivo';

}

return null;

},

),

TextFormField(

controller: \_apelidoController,

decoration: InputDecoration(labelText: 'Apelido (Opcional)'),

),

ElevatedButton(

onPressed: () {

if (\_formKey.currentState!.validate()) {

final planeta = Planeta(

nome: \_nomeController.text,

distanciaSol: double.parse(\_distanciaController.text),

tamanho: double.parse(\_tamanhoController.text),

apelido: \_apelidoController.text,

);

dbHelper.insertPlaneta(planeta);

Navigator.pop(context);

}

},

child: Text('Salvar'),

),

],

),

),

),

);

}

}

// --- Tela de Detalhes do Planeta ---

class DetalhesPlanetaPage extends StatelessWidget {

final Planeta planeta;

DetalhesPlanetaPage(this.planeta);

@override

Widget build(BuildContext context) {

return Scaffold(

appBar: AppBar(title: Text(planeta.nome)),

body: Padding(

padding: const EdgeInsets.all(16.0),

child: Column(

crossAxisAlignment: CrossAxisAlignment.start,

children: [

Text('Nome: ${planeta.nome}'),

Text('Distância do Sol: ${planeta.distanciaSol} UA'),

Text('Tamanho: ${planeta.tamanho} Km'),

Text('Apelido: ${planeta.apelido ?? "Não Informado"}'),

],

),

),

);

}

}

// --- Função principal ---

void main() {

runApp(MaterialApp(

debugShowCheckedModeBanner: false,

home: ListPlanetasPage(),

));

}