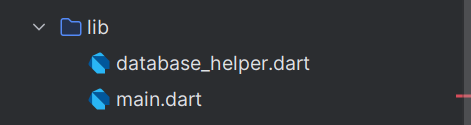
# **Source code**



|  |
| --- |
| database\_helper.dart |
| import 'package:sqflite/sqflite.dart'; import 'package:path/path.dart';  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('appnotes\_database.db');  return *\_database*!;  }   Future<Database> \_initDB(String filePath) async{  final dbPath = await getDatabasesPath();  final path = join(dbPath, filePath);  return openDatabase(path, version: 1, onCreate: \_onCreate);  }   Future \_onCreate(Database db, int version) async {  await db.execute('''  CREATE TABLE notes(  id INTEGER PRIMARY KEY AUTOINCREMENT,  title TEXT,  description TEXT  )  ''');  }   Future<int> insertNote(Map<String, dynamic> note) async{  final db = await *instance*.database;  return await db.insert('notes', note);  }   Future<List<Map<String, dynamic>>> fetchNotes() async {  final db = await *instance*.database;  return await db.query('notes');  }   Future<int> deleteNote(int id) async {  final db = await *instance*.database;  return await db.delete('notes', where: 'id = ?', whereArgs: [id]);  } } |

|  |
| --- |
| main.dart |
| import 'package:flutter/material.dart'; import 'database\_helper.dart';  void main() {  runApp(MyApp()); }  class MyApp extends StatelessWidget {  @override  Widget build(BuildContext context) {  return MaterialApp(  title: 'MY Notes app',  theme: ThemeData(  primarySwatch: Colors.*red*,  colorScheme:  ColorScheme.fromSwatch(primarySwatch: Colors.*cyan*).copyWith(  secondary: Colors.*green*, // Accent color for FAB, switches, etc.  ),  ),  home: MyHomePage(),  debugShowCheckedModeBanner: false,  );  } }  class MyHomePage extends StatefulWidget {  @override  State<MyHomePage> createState() => \_MyHomePageState(); }  class \_MyHomePageState extends State<MyHomePage> {  final \_titleController = TextEditingController();  final \_descriptionController = TextEditingController();  List<Map<String, dynamic>> \_notes = [];   @override  void initState() {  super.initState();  \_fetchNotes();  }   Future<void> \_fetchNotes() async {  final notes = await DatabaseHelper.*instance*.fetchNotes();  setState(() {  \_notes = notes;  });  }   Future<void> \_addNote() async {  final title = \_titleController.text;  final description = \_descriptionController.text;  if (title.isNotEmpty && description.isNotEmpty) {  await DatabaseHelper.*instance* .insertNote({'title': title, 'description': description});  \_titleController.clear();  \_descriptionController.clear();  \_fetchNotes();  }  }   Future<void> \_deleteNote(int id) async {  await DatabaseHelper.*instance*.deleteNote(id);  \_fetchNotes();  }   \_showNoteDialog() {  showDialog(  context: context,  builder: (context) {  return AlertDialog(  backgroundColor: Color(0xFF1565C0),  // Set the background color (e.g., deep blue)  shape: RoundedRectangleBorder(  borderRadius:  BorderRadius.circular(12), // Optional: Rounded corners  ),  title: Text(  'Add New Note',  style: TextStyle(color: Colors.*white*),  ),  content: Column(  children: [  TextField(  controller: \_titleController,  decoration: InputDecoration(  labelText: 'Title',  labelStyle: TextStyle(  color: Colors  .*white*, // Change label (title) text color to white  ),  ),  style: TextStyle(  color: Colors.*white*, // Change entered text color to white  ),  ),  TextField(  controller: \_descriptionController,  decoration: InputDecoration(  labelText: 'Description',  labelStyle: TextStyle(  color: Colors  .*white*, // Change label (description) text color to white  ),  ),  style: TextStyle(  color: Colors.*white*, // Change entered text color to white  ),  ),  SizedBox(height: 20),  ElevatedButton(  onPressed: () { // Insert new note to the database  \_addNote();  Navigator.*pop*(context);  \_fetchNotes();  },  child: Text(  'Save',  style: TextStyle(  color: Color(  0xFF424242), // Make sure this color is explicitly set  ),  ),  ),  ],  ),  );  },  );  }   @override  Widget build(BuildContext context) {  return Scaffold(  appBar: AppBar(  title: Center(  child: Text(  'Personal Notes',  style: TextStyle(  color: Colors.*white*, fontSize: 30, fontWeight: FontWeight.*bold*),  ),  ),  backgroundColor:  Color(0xFF1565C0), // Replace 'Colors.blue' with your desired color  ),  body: Container(  color: Color(0xFFF5F5F5),  child: Padding(  padding: const EdgeInsets.all(16.0),  child: Column(  children: [  SizedBox(height: 100),  ElevatedButton(  onPressed: \_showNoteDialog,  child: Text(  'Add Note',  style: TextStyle(color: Colors.*white*, fontSize: 20),  ),  style: ElevatedButton.*styleFrom*(  backgroundColor: Color(0xFF1565C0),  side: BorderSide(  color: Colors.*lightBlueAccent*,  // Border color (same as button background)  width: 4, // Border thickness  ), // Vibrant Green  shape: RoundedRectangleBorder(  borderRadius:  BorderRadius.circular(12), // Sleek rounded corners  ),  ),  ),  SizedBox(height: 20),  Expanded(  child: ListView.builder(  itemCount: \_notes.length,  itemBuilder: (context, index) {  final note = \_notes[index];  return Column(  children: [  Container(  color: Color(0xFF1565C0),  child: ListTile(  title: Text(  note['title'],  style: TextStyle(color: Colors.*greenAccent*),  ),  subtitle: Text(  note['description'],  style: TextStyle(color: Colors.*white*),  ),  trailing: IconButton(  icon: Icon(  Icons.*delete*,  color: Colors.*red*,  ),  onPressed: () => \_deleteNote(note['id']),  ),  selectedTileColor: Color(0xFF001F3D),  ),  ),  Divider(  color: Colors.*white*, // Divider color (white line between tiles)  thickness: 1, // Line thickness  ),  ],  );  },  ),  ),  ],  ),  ),  ),  );  } } |