

```

1 // PRACTICAL-11: Develop a flutter app with following functionalities:
2 // Display two ElevatedButtons "Add Student" and "List Student Record" on first screen.
3 // On tap of "Add Student", ask student name, roll number, and address of the student. Save student data in SQLite
  database using sqflite package on tap of "Save Student Data".
4 // On tap of "List Student Record", retrieve the student records stored in SQLite database and display with edit and
  delete functionalities.
5
6 // Name: Angat Nayanbhai Shah
7 // Enrollment No: 202203103510097
8 // Branch: B.Tech Computer Science and Engineering
9
10 // CODE:
11
12 import 'package:flutter/material.dart';
13 import 'student_database.dart';
14 void main() {
15   runApp(MyApp());
16 }
17 class MyApp extends StatelessWidget {
18   @override
19   Widget build(BuildContext context) {
20     return MaterialApp(
21       title: 'Student Management',
22       home: HomeScreen(),
23       debugShowCheckedModeBanner: false,
24     );
25   }
26 }
27 class HomeScreen extends StatelessWidget {
28   @override
29   Widget build(BuildContext context) {
30     return Scaffold(
31       appBar: AppBar(title: Text('PRACTICAL 11')),
32       body: Center(
33         child: Column(
34           mainAxisAlignment: MainAxisAlignment.center,
35           children: [
36             ElevatedButton(
37               onPressed: () {
38                 Navigator.push(context, MaterialPageRoute(builder: (context) => AddStudentScreen()));
39               },
40               child: Text('Add Student'),
41             ),
42             SizedBox(height: 20),
43             ElevatedButton(
44               onPressed: () {
45                 Navigator.push(context, MaterialPageRoute(builder: (context) => ListStudentScreen()));
46               },
47               child: Text('List Student Record'),
48             ),
49             Spacer(),
50             Padding(
51               padding: const EdgeInsets.all(16.0),
52               child: Text(
53                 '202203103510097', // Footer text
54                 style: TextStyle(
55                   fontSize: 14,
56                   color: Colors.grey,

```

```

57     ),
58     textAlign: TextAlign.center,
59     ),
60     ),
61   ],
62   ),
63   ),
64   );
65 }
66 }
67 class AddStudentScreen extends StatefulWidget {
68   @override
69   _AddStudentScreenState createState() => _AddStudentScreenState();
70 }
71 class _AddStudentScreenState extends State<AddStudentScreen> {
72   final _nameController = TextEditingController();
73   final _rollNumberController = TextEditingController();
74   final _addressController = TextEditingController();
75   Future<void> _saveStudent() async {
76     final name = _nameController.text;
77     final rollNumber = _rollNumberController.text;
78     final address = _addressController.text;
79     await StudentDatabase.instance.addStudent(name, rollNumber, address);
80     Navigator.pop(context);
81   }
82   @override
83   Widget build(BuildContext context) {
84     return Scaffold(
85       appBar: AppBar(title: Text('Add Student')),
86       body: Padding(
87         padding: const EdgeInsets.all(16.0),
88         child: Column(
89           children: [
90             TextField(controller: _nameController, decoration: InputDecoration(labelText: 'Student Name')),
91             TextField(controller: _rollNumberController, decoration: InputDecoration(labelText: 'Roll Number')),
92             TextField(controller: _addressController, decoration: InputDecoration(labelText: 'Address')),
93             SizedBox(height: 20),
94             ElevatedButton(onPressed: _saveStudent, child: Text('Save Student Data')),
95           ],
96         ),
97       ),
98     );
99   }
100 }
101 class ListStudentScreen extends StatefulWidget {
102   @override
103   _ListStudentScreenState createState() => _ListStudentScreenState();
104 }
105 class _ListStudentScreenState extends State<ListStudentScreen> {
106   late Future<List<Map<String, dynamic>>> _studentList;
107   @override
108   void initState() {
109     super.initState();
110     _fetchStudents();
111   }
112   Future<void> _fetchStudents() async {
113     setState() {
114       _studentList = StudentDatabase.instance.getStudents();

```

```

115   });
116   }
117   Future<void> _deleteStudent(int id) async {
118     await StudentDatabase.instance.deleteStudent(id);
119     _fetchStudents();
120   }
121   Future<void> _editStudent(int id, String name, String rollNumber, String address) async {
122     await StudentDatabase.instance.updateStudent(id, name, rollNumber, address);
123     _fetchStudents();
124   }
125   @override
126   Widget build(BuildContext context) {
127     return Scaffold(
128       appBar: AppBar(title: Text('Student Records')),
129       body: FutureBuilder<List<Map<String, dynamic>>>>(
130         future: _studentList,
131         builder: (context, snapshot) {
132           if (!snapshot.hasData) {
133             return Center(child: CircularProgressIndicator());
134           }
135           final students = snapshot.data!;
136           return ListView.builder(
137             itemCount: students.length,
138             itemBuilder: (context, index) {
139               final student = students[index];
140               return ListTile(
141                 title: Text(student['name']),
142                 subtitle: Text('Roll No: ${student['roll_number']}, Address: ${student['address']}'),
143                 trailing: Row(
144                   mainAxisAlignment: MainAxisAlignment.min,
145                   children: [
146                     IconButton(
147                       icon: Icon(Icons.edit),
148                       onPressed: () {
149                         // Add edit functionality
150                         _editStudent(student['id'], student['name'], student['roll_number'], student['address']);
151                       },
152                     ),
153                     IconButton(
154                       icon: Icon(Icons.delete),
155                       onPressed: () {
156                         _deleteStudent(student['id']);
157                       },
158                     ),
159                   ],
160                 ),
161             );
162           },
163         );
164       },
165     ),
166   );
167   }
168   }
169
170 // student_database.dart
171 import 'package:sqflite/sqflite.dart';
172 import 'package:path/path.dart';

```

```

173 class StudentDatabase {
174     static final StudentDatabase instance = StudentDatabase._init();
175     static Database? _database;
176     StudentDatabase._init();
177     Future<Database> get database async {
178         if (_database != null) return _database!;
179
180         _database = await _initDB('students.db');
181         return _database!;
182     }
183     Future<Database> _initDB(String filePath) async {
184         final dbPath = await getDatabasesPath();
185         final path = join(dbPath, filePath);
186         return await openDatabase(
187             path,
188             version: 1,
189             onCreate: _createDB,
190         );
191     }
192     Future _createDB(Database db, int version) async {
193         const idType = 'INTEGER PRIMARY KEY AUTOINCREMENT';
194         const textType = 'TEXT NOT NULL';
195         await db.execute("""
196             CREATE TABLE students (
197                 id $idType,
198                 name $textType,
199                 roll_number $textType,
200                 address $textType
201             )
202             """);
203     }
204     Future<int> addStudent(String name, String rollNumber, String address) async {
205         final db = await instance.database;
206         final data = {'name': name, 'roll_number': rollNumber, 'address': address};
207         return await db.insert('students', data);
208     }
209     Future<List<Map<String, dynamic>>> getStudents() async {
210         final db = await instance.database;
211         return await db.query('students');
212     }
213     Future<int> updateStudent(int id, String name, String rollNumber, String address) async {
214         final db = await instance.database;
215         final data = {'name': name, 'roll_number': rollNumber, 'address': address};
216         return await db.update('students', data, where: 'id = ?', whereArgs: [id]);
217     }
218     Future<int> deleteStudent(int id) async {
219         final db = await instance.database;
220         return await db.delete('students', where: 'id = ?', whereArgs: [id]);
221     }
222     Future close() async {
223         final db = await instance.database;
224         db.close();
225     }
226 }

```