

```

1 // PRACTICAL-12: Develop a flutter app to maintain tasks. Allow the user to create a task and save in SQLite using
  // Drift package. User can remove a task, view completed or all tasks.
2
3 // Name: Angat Nayanbhai Shah
4 // Enrollment No: 202203103510097
5 // Branch: B.Tech Computer Science and Engineering
6
7 // CODE:
8
9 import 'package:flutter/material.dart';
10 import 'database.dart';
11 import 'task_repository.dart';
12 void main() {
13   runApp(MyApp());
14 }
15 class MyApp extends StatelessWidget {
16   @override
17   Widget build(BuildContext context) {
18     return MaterialApp(
19       title: 'Task Manager',
20       theme: ThemeData(primarySwatch: Colors.blue),
21       home: TaskManager(),
22       debugShowCheckedModeBanner: false,
23     );
24   }
25   class TaskManager extends StatefulWidget {
26     @override
27     _TaskManagerState createState() => _TaskManagerState();
28   }
29   class _TaskManagerState extends State<TaskManager> {
30     final AppDatabase _database = AppDatabase();
31     late TaskRepository _repository;
32     final TextEditingController _taskController = TextEditingController();
33     List<Task> _tasks = [];
34     @override
35     void initState() {
36       super.initState();
37       _repository = TaskRepository(_database);
38       _loadTasks();
39     }
40     Future<void> _loadTasks() async {
41       final tasks = await _repository.getAllTasks();
42       setState(() {
43         _tasks = tasks..sort((a, b) => a.isCompleted ? 1 : -1);
44       });
45     }
46     void _addTask() async {
47       if (_taskController.text.isNotEmpty) {
48         await _repository.addTask(_taskController.text);
49         _taskController.clear();
50         _loadTasks();
51       }
52     }
53     void _removeTask(int id) async {
54       await _repository.removeTask(id);
55       _loadTasks();
56     }
57     void _toggleTaskCompletion(int id, bool isCompleted) async {
58       await _repository.markTaskAsCompleted(id, !isCompleted);
59       _loadTasks();
60     }
61     @override
62     Widget build(BuildContext context) {
63       return Scaffold(

```

```

58 appBar: AppBar(title: Text('PRACTICAL 12')),
59 body: Column(
60   children: [
61     Padding(
62       padding: const EdgeInsets.all(16.0),
63       child: Row(
64         children: [
65           Expanded(
66             child: TextField(
67               controller: _taskController,
68               decoration: InputDecoration(labelText: 'TASK TITLE'),
69             ),
70           ),
71           IconButton(
72             icon: Icon(Icons.add),
73             onPressed: _addTask,
74           ),
75         ],
76       ),
77     ),
78     Expanded(
79       child: ListView.builder(
80         itemCount: _tasks.length,
81         itemBuilder: (context, index) {
82           final task = _tasks[index];
83           return ListTile(
84             leading: Checkbox(
85               value: task.isCompleted,
86               onChanged: (value) {
87                 _toggleTaskCompletion(task.id, task.isCompleted);
88               },
89             ),
90             title: Text(
91               task.title,
92               style: TextStyle(
93                 decoration: task.isCompleted
94                   ? TextDecoration.lineThrough
95                   : TextDecoration.none,
96             ),
97           ),
98             trailing: IconButton(
99               icon: Icon(Icons.delete),
100               onPressed: () => _removeTask(task.id),
101             ),
102           );
103         },
104       ),
105     ),
106     Spacer(),
107     Padding(
108       padding: const EdgeInsets.all(16.0),
109       child: Text(
110         '202203103510097',
111         style: TextStyle(
112           fontSize: 14,
113           color: Colors.grey,
114         ),
115         textAlign: TextAlign.center,

```

```

116     ),
117     ),
118   ],
119   ),
120 );
121 }
122 }
123
124 // database.dart
125 import 'package:drift/drift.dart';
126 import 'package:drift/native.dart';
127 part 'database.g.dart';
128 @DriftDatabase(tables: [Tasks])
129 class AppDatabase extends _$AppDatabase {
130   AppDatabase() : super(NativeDatabase.memory());
131   @override
132   int get schemaVersion => 1; }
133 class Tasks extends Table {
134   IntColumn get id => integer().autoIncrement();
135   TextColumn get title => text().withLength(min: 1, max: 100);
136   BoolColumn get isCompleted => boolean().withDefault(const Constant(false));
137 }
138
139 // task_model.dart
140 class Task {
141   final int? id;
142   final String title;
143   final bool isCompleted;
144   Task({this.id, required this.title, this.isCompleted = false}); }
145
146 // task_repository.dart
147 import 'package:drift/drift.dart';
148 import 'database.dart';
149 class TaskRepository {
150   final AppDatabase _database;
151   TaskRepository(this._database);
152   Future<List<Task>> getAllTasks() async {
153     return await (_database.select(_database.tasks).get()).then((rows) {
154       return rows.map((row) => Task(id: row.id, title: row.title, isCompleted: row.isCompleted)).toList();
155     }); }
156   Future<void> addTask(String title) async {
157     await _database.into(_database.tasks).insert(
158       TasksCompanion(
159         title: Value(title),
160         isCompleted: const Value(false),
161       ),
162     ); }
163   Future<void> removeTask(int id) async {
164     await (_database.delete(_database.tasks)..where((tbl) => tbl.id.equals(id))).go();
165   }
166   Future<void> markTaskAsCompleted(int id, bool isCompleted) async {
167     await (_database.update(_database.tasks)
168       ..where((tbl) => tbl.id.equals(id)))
169       .write(TasksCompanion(
170         isCompleted: Value(isCompleted),
171       )); }
172 }

```