Practical No: - 01

Flutter program to work with SQLite Database

DatabaseHelper.dart:

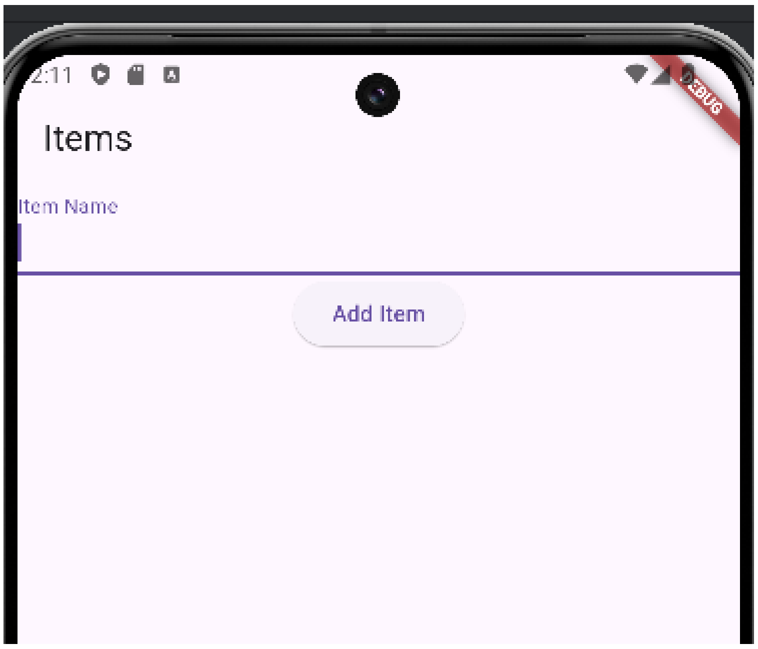
```
import 'package:flutter/material.dart';
import 'package:path/path.dart';
import 'package:sqflite/sqflite.dart';
class DatabaseHelper {
  static final DatabaseHelper _instance = DatabaseHelper._internal();
  factory DatabaseHelper() => _instance;
  static Database? _database;
  DatabaseHelper._internal();
  Future<Database> get database async {
    _database ??= await _initDatabase();
    return _database!;
  }
  Future<Database> _initDatabase() async {
    String path = join(await getDatabasesPath(), 'simple_database.db');
    return await openDatabase(
      path,
      onCreate: (db, version) {
        return db.execute(
          'CREATE TABLE items(id INTEGER PRIMARY KEY
AUTOINCREMENT, name TEXT)',
        );
      },
      version: 1,
    );
  }
  Future<void> insertItem(String name) async {
    final db = await database;
    await db.insert(
      'items',
      {'name': name},
      conflictAlgorithm: ConflictAlgorithm.replace,
    );
  }
  Future<List<Map<String, dynamic>>> fetchItems() async {
    final db = await database;
    return await db.query('items');
  }
  Future<void> deleteItem(int id) async {
    final db = await database;
    await db.delete(
      'items',
      where: 'id = ?',
      whereArgs: [id],
    );
  }
}
```

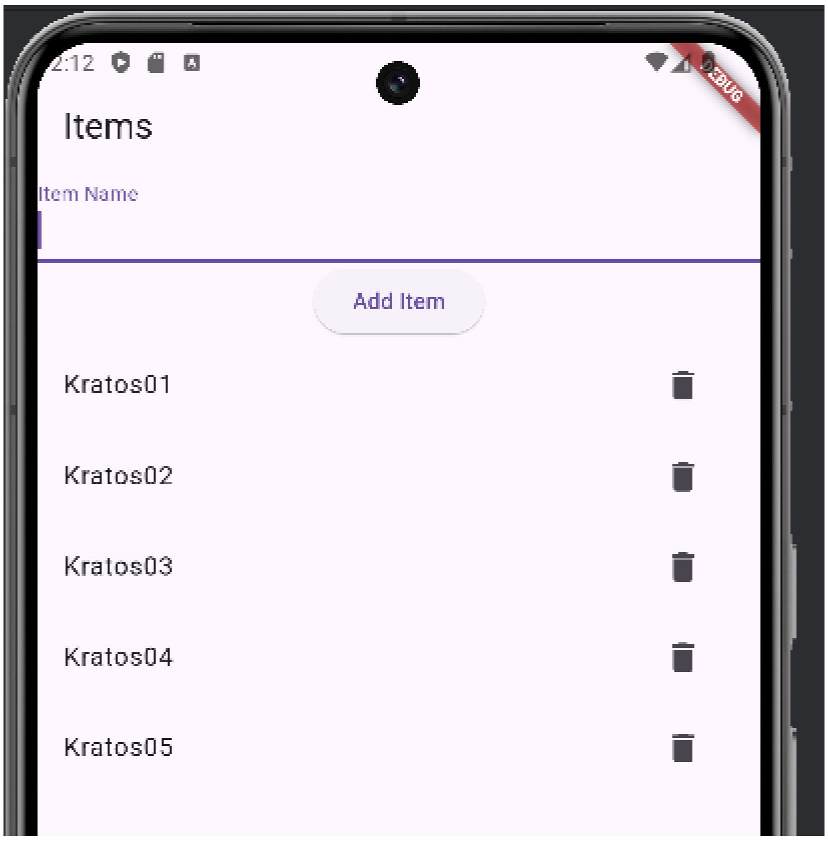
Main.dart:

```
import 'package:flutter/material.dart';
import 'package:flutter/material.dart';
import 'DatabaseHelper.dart';
void main() {
  runApp(MyApp());
}
class MyApp extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      title: 'Simple Database App',
      home: ItemListScreen(),
    );
  }
}
class ItemListScreen extends StatefulWidget {
  @override
  _ItemListScreenState createState() => _ItemListScreenState();
}
class _ItemListScreenState extends State<ItemListScreen> {
  final DatabaseHelper _databaseHelper = DatabaseHelper();
  final TextEditingController _controller = TextEditingController();
  List<Map<String, dynamic>> _items = [];
```

```
@override
void initState() {
  super.initState();
  _loadItems();
}
Future<void> _loadItems() async {
  final items = await _databaseHelper.fetchItems();
  setState(() {
    _items = items;
  });
}
Future<void> _addItem() async {
  if (_controller.text.isNotEmpty) {
    await _databaseHelper.insertItem(_controller.text);
    _controller.clear();
    _loadItems();
  }
}
Future<void> _deleteItem(int id) async {
  await _databaseHelper.deleteItem(id);
  _loadItems();
}
@override
Widget build(BuildContext context) {
  return Scaffold(
    appBar: AppBar(title: Text('Items')),
    body: Column(
      children: [
        TextField(
          controller: _controller,
          decoration: InputDecoration(labelText: 'Item Name'),
        ),
        ElevatedButton(
          onPressed: _addItem,
          child: Text('Add Item'),
        ),
        Expanded(
            child: ListView.builder(
              itemCount: _items.length,
              itemBuilder: (context, index) {
                return ListTile(
                  title: Text(_items[index]['name']),
                  trailing: IconButton(
                    icon: Icon(Icons.delete),
                    onPressed: () => _deleteItem(_items[index]['id']),
                  ),
                );
              },
            ),
          ),
        ],
      ),
    );
  }
}
```

Output:





Items

Item Name

Add Item

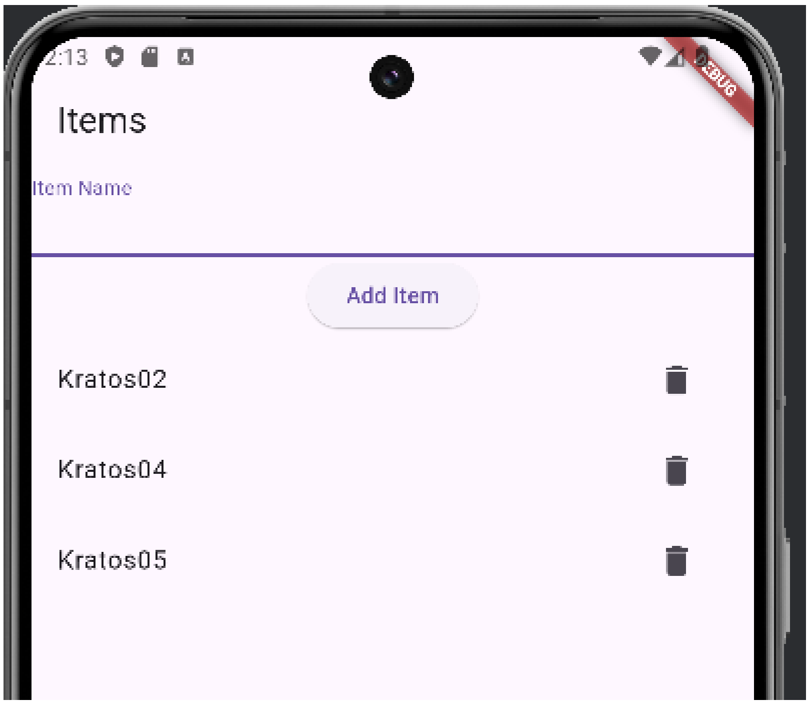
Kratos01

Kratos02

Kratos03

Kratos04

Kratos05



Items

Item Name

Add Item

Kratos02

Kratos04

Kratos05

Practical No: - 02

Flutter program based on RestAPI (Json format)

Main.dart: -

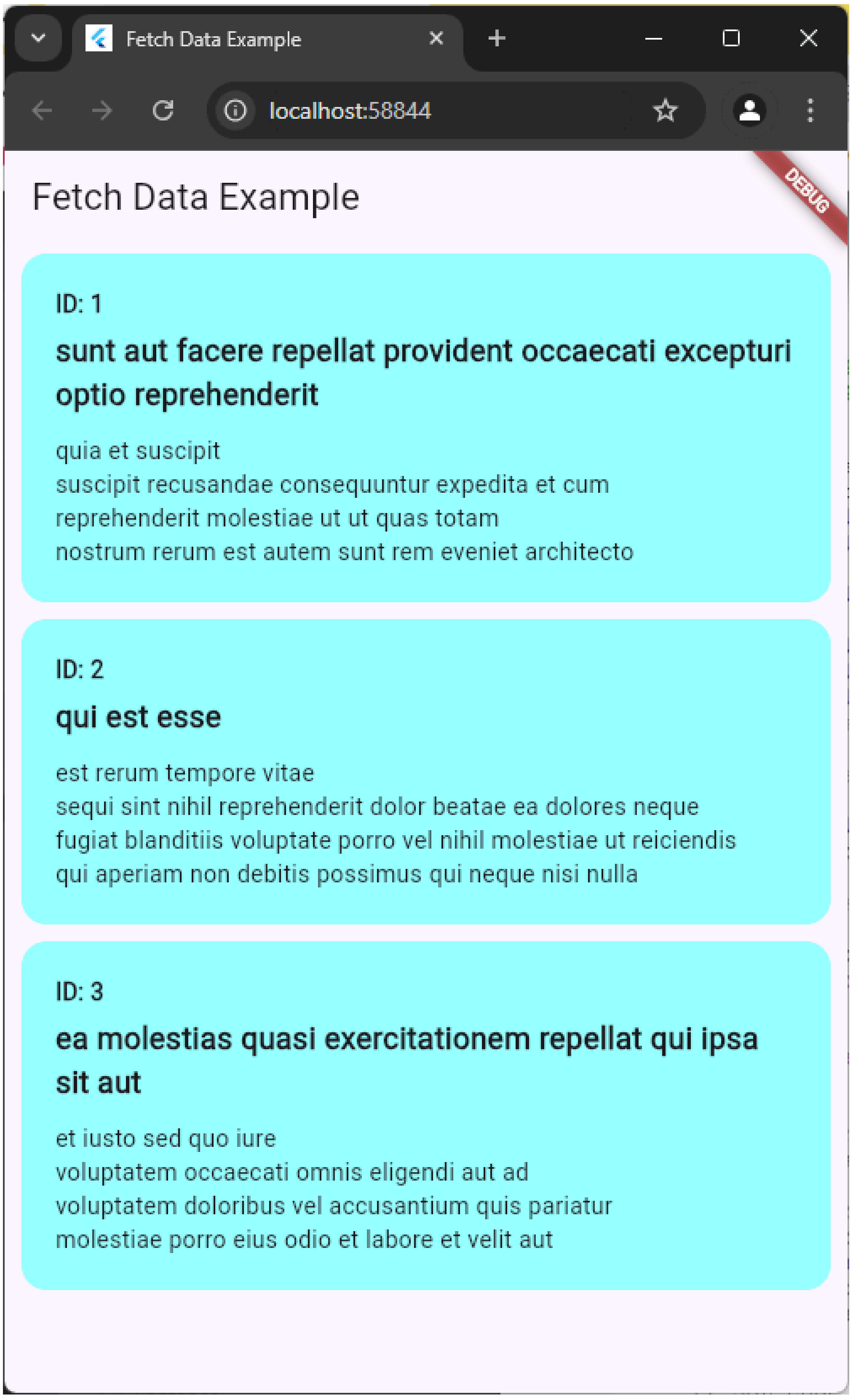
```
import 'dart:convert';
import 'package:flutter/material.dart';
import 'package:http/http.dart' as http; // Import the http package
import 'package:json3/post.dart';
Future<List<Post>> fetchPost() async {
  final response = await
http.get(Uri.parse('https://jsonplaceholder.typicode.com/posts'));
  if (response.statusCode == 200) {
    final parsed = json.decode(response.body).cast<Map<String, dynamic>>();
    // Limit to the first 3 posts
    return parsed.take(3).map<Post>((json) => Post.fromMap(json)).toList();
  } else {
    throw Exception('Failed to load posts');
  }
}
void main() => runApp(MyApp());
class MyApp extends StatefulWidget {
  @override
  _MyAppState createState() => _MyAppState();
}
class _MyAppState extends State<MyApp> {
  late Future<List<Post>> futurePost;
  @override
  void initState() {
    super.initState();
    futurePost = fetchPost();
  }

  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      title: 'Fetch Data Example',
      theme: ThemeData(
        primaryColor: Colors.lightBlueAccent,
),
home: Scaffold(
  appBar: AppBar(
    title: Text('Fetch Data Example'),
  ),
  body: FutureBuilder<List<Post>>(
    future: futurePost,
    builder: (context, snapshot) {
      if (snapshot.hasData) {
        return ListView.builder(
          itemCount: snapshot.data!.length,
          itemBuilder: (_, index) => Container(
            margin: EdgeInsets.symmetric(horizontal: 10, vertical: 5),
            padding: EdgeInsets.all(20.0),
            decoration: BoxDecoration(
              color: Color(0xff97FFFF),
              borderRadius: BorderRadius.circular(15.0),
            ),
            child: Column(
              mainAxisAlignment: MainAxisAlignment.start,
              crossAxisAlignment: CrossAxisAlignment.start,
              children: [
                Text(
                  "ID: ${snapshot.data![index].id}",
                  style: TextStyle(
                    fontSize: 14.0,
                    fontWeight: FontWeight.bold,
                  ),
                ),
                SizedBox(height: 5),
                Text(
                  "${snapshot.data![index].title}",
                  style: TextStyle(
                    fontSize: 18.0,
                    fontWeight: FontWeight.bold,
                  ),
                ),
                    SizedBox(height: 10),
                    Text("${snapshot.data![index].body}"),
                  ],
                ),
              ),
            );
          } else {
            return Center(child: CircularProgressIndicator());
          }
        },
      ),
    ),
  );
 }
}
```

Post.dart: -

```
import 'dart:convert';
List<Post> postFromJson(String str) =>
  List<Post>.from(json.decode(str).map((x) => Post.fromMap(x)));
class Post {
 Post({
  required this.userId,
  required this.id,
  required this.title,
  required this.body,
 });
 int userId;
 int id;
 String title;
 String body;
 factory Post.fromMap(Map<String, dynamic> json) => Post(
  userId: json["userId"],
  id: json["id"],
  title: json["title"],
  body: json["body"],
 );
}
```

Output: -



```
Fetch Data Example

ID: 1
sunt aut facere repellat provident occaecati excepturi optio reprehenderit
quia et suscipit
suscipit recusandae consequuntur expedita et cum
reprehenderit molestiae ut ut quas totam
nostrum rerum est autem sunt rem eveniet architecto

ID: 2
qui est esse
est rerum tempore vitae
sequi sint nihil reprehenderit dolor beatae ea dolores neque
fugiat blanditiis voluptate porro vel nihil molestiae ut reiciendis
qui aperiam non debitis possimus qui neque nisi nulla

ID: 3
ea molestias quasi exercitationem repellat qui ipsa sit aut
et iusto sed quo iure
voluptatem occaecati omnis eligendi aut ad
voluptatem doloribus vel accusantium quis pariatur
molestiae porro eius odio et labore et velit aut
```