

## Practical 10 : Create and application Crud Operation with SQLite in Flutter.

### Main.dart

```
import 'package:flutter/material.dart';
import 'package:resetapi/sqlHelper.dart';

void main() {
  runApp(const MyApp());
}

class MyApp extends StatelessWidget {
  const MyApp({Key? key}) : super(key: key);

  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      // Remove the debug banner
      debugShowCheckedModeBanner: false,
      title: 'SQLITE',
      theme: ThemeData(
        primarySwatch: Colors.orange,
      ),
      home: const HomePage(),
    );
  }
}

class HomePage extends StatefulWidget {
  const HomePage({Key? key}) : super(key: key);

  @override
  _HomePageState createState() => _HomePageState();
}

class _HomePageState extends State<HomePage> {
  // All journals
  List<Map<String, dynamic>> _journals = [];

  bool _isLoading = true;
  // This function is used to fetch all data from the database
  void _refreshJournals() async {
    final data = await SQLHelper.getItems();
    setState(() {
      _journals = data;
      _isLoading = false;
    });
  }

  @override
  void initState() {
```

```
super.initState();_refreshJournals(); // Loading the diary when the app starts
}

final TextEditingController _titleController = TextEditingController();
final TextEditingController _descriptionController = TextEditingController();

// This function will be triggered when the floating button is pressed
// It will also be triggered when you want to update an item
void _showForm(int? id) async {
  if (id != null) {
    // id == null -> create new item
    // id != null -> update an existing item
    final existingJournal =
      _journals.firstWhere((element) => element['id'] == id);
    _titleController.text = existingJournal['title'];
    _descriptionController.text = existingJournal['description'];
  }

  showModalBottomSheet(
    context: context,
    elevation: 5,
    isScrollControlled: true,
    builder: (_) => Container(
      padding: EdgeInsets.only(
        top: 15,
        left: 15,
        right: 15,
        // this will prevent the soft keyboard from covering the text fields
        bottom: MediaQuery.of(context).viewInsets.bottom + 120,
      ),
      child: Column(
        mainAxisAlignment: MainAxisAlignment.min,
        crossAxisAlignment: CrossAxisAlignment.end,
        children: [
          TextField(
            controller: _titleController,
            decoration: const InputDecoration(hintText: 'Title'),
          ),
          const SizedBox(
            height: 10,
          ),
          TextField(
            controller: _descriptionController,
            decoration: const InputDecoration(hintText: 'Description'),
          ),
          const SizedBox(
            height: 20,
          ),
          ElevatedButton(
            onPressed: () async {
              // Save new journal
            }
          )
        ]
      )
    )
  );
}
```

```
        if (id == null) {
          await _addItem();
        }

        if (id != null) {
          await _updateItem(id);
        }

        // Clear the text fields
        _titleController.text = "";
        _descriptionController.text = "";
      )
    ],
  ),
));
} // Close the bottom sheet Navigator.of(context).pop();
},
child: Text(id == null ? 'Create New' : 'Update'),

// Insert a new journal to the database
Future<void> _addItem() async {
  await SQLHelper.createItem(
    _titleController.text, _descriptionController.text);
  _refreshJournals();
}

// Update an existing journal
Future<void> _updateItem(int id) async {
  await SQLHelper.updateItem(
    id, _titleController.text, _descriptionController.text);
  _refreshJournals();
}

// Delete an item
void _deleteItem(int id) async {
  await SQLHelper.deleteItem(id);
  ScaffoldMessenger.of(context).showSnackBar(const SnackBar(
    content: Text('Successfully deleted a journal!'),
  ));
  _refreshJournals();
}

@override
Widget build(BuildContext context) {
  return Scaffold(
    appBar: AppBar(
      title: const Text('SQL'),
    ),
    body: _isLoading
      ? const Center(
        child: CircularProgressIndicator(),
```

```

)
: ListView.builder(
itemCount: _journals.length,
itemBuilder: (context, index) => Card(
color: Colors.orange[200],
margin: const EdgeInsets.all(15),
child: ListTile(
title: Text(_journals[index]['title']),
subtitle: Text(_journals[index]['description']),
trailing: SizedBox(
width: 100,
child: Row(
children: [
IconButton(
icon: const Icon(Icons.edit),
onPressed: () => _showForm(_journals[index]['id']),
),
IconButton(
icon: const Icon(Icons.delete),
onPressed: () =>
_deleteItem(_journals[index]['id']),
),
],
),
)),
),
floatingActionButton: FloatingActionButton(
child: const Icon(Icons.add),
onPressed: () => _showForm(null),
),
);
}
}

```

### sqlHelper.dart

```

import 'package:flutter/foundation.dart';
import 'package:sqflite/sqflite.dart' as sql;

class SQLHelper {
  static Future<void> createTables(sql.Database database) async {
    await database.execute("""CREATE TABLE items(
      id INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,
      title TEXT,
      description TEXT,
      createdAt TIMESTAMP NOT NULL DEFAULT CURRENT_TIMESTAMP
    )

```

```
        """);
    }
    // id: the id of a item
    // title, description: name and description of your activity
    // created_at: the time that the item was created. It will be automatically handled by SQLite

    static Future<sql.Database> db() async {
        return sql.openDatabase(
            'dbtech.db',
            version: 1,
            onCreate: (sql.Database database, int version) async {
                await createTables(database);
            },
        );
    }

    // Create new item (journal)
    static Future<int> createItem(String title, String? description) async {
        final db = await SQLHelper.db();

        final data = {'title': title, 'description': description};
        final id = await db.insert('items', data,
            conflictAlgorithm: sql.ConflictAlgorithm.replace);
        return id;
    }

    // Read all items (journals)
    static Future<List<Map<String, dynamic>>> getItems() async {
        final db = await SQLHelper.db();
        return db.query('items', orderBy: "id");
    }

    // Read a single item by id
    // The app doesn't use this method but I put here in case you want to see it
    static Future<List<Map<String, dynamic>>> getItem(int id) async {
        final db = await SQLHelper.db();
        return db.query('items', where: "id = ?", whereArgs: [id], limit: 1);
    }

    // Update an item by id
    static Future<int> updateItem(
        int id, String title, String? description) async {
        final db = await SQLHelper.db();

        final data = {
            'title': title,
            'description': description,
            'createdAt': DateTime.now().toString()
        };

        final result =
```

```
await db.update('items', data, where: "id = ?", whereArgs: [id]);
return result;
}

// Delete
static Future<void> deleteItem(int id) async {
  final db = await SQLHelper.db();
  try {
    await db.delete("items", where: "id = ?", whereArgs: [id]);
  } catch (err) {
    debugPrint("Something went wrong when deleting an item: $err");
  }
}
}
```

**dependencies:**

**flutter:**

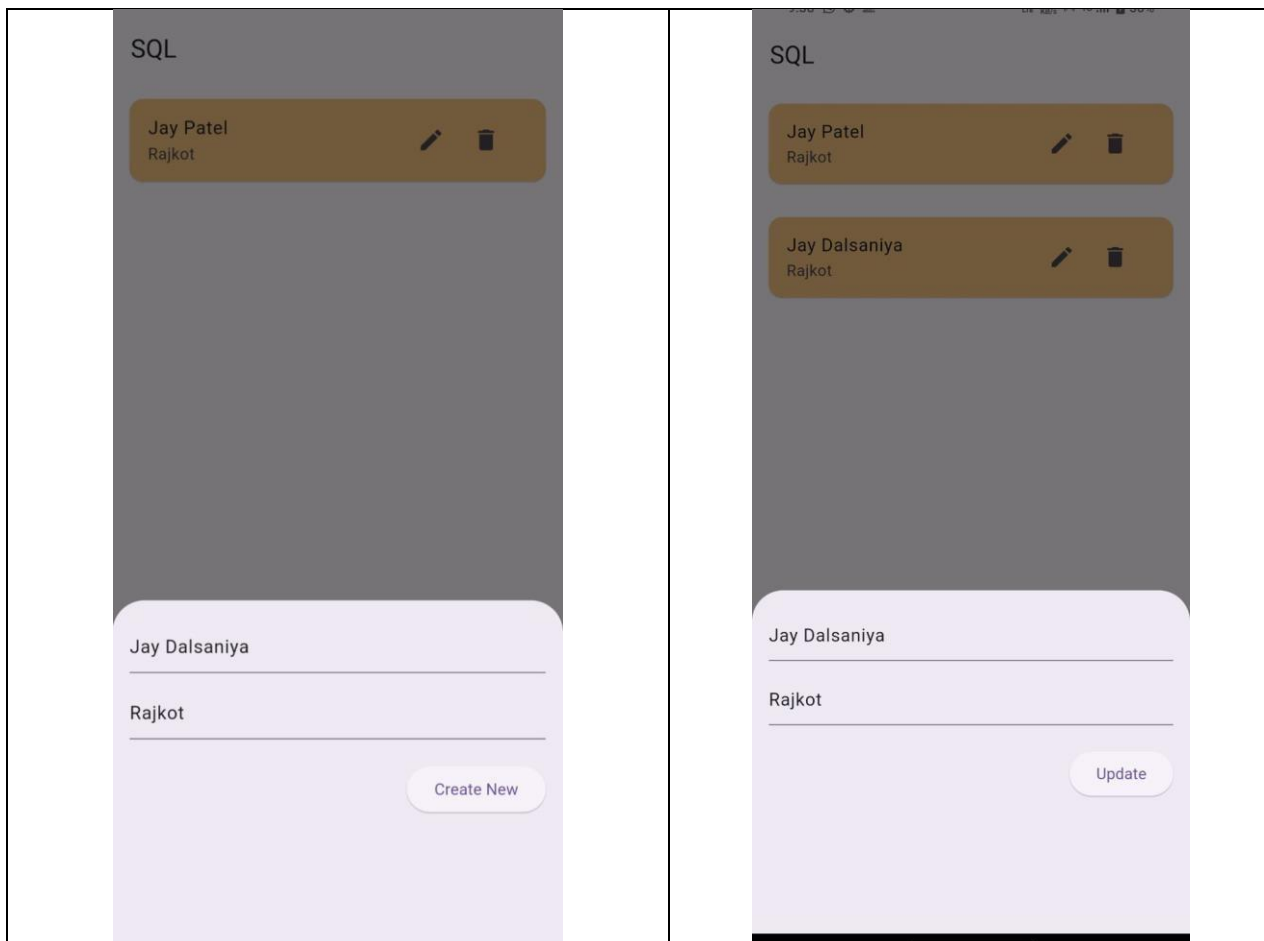
**sdk: flutter**

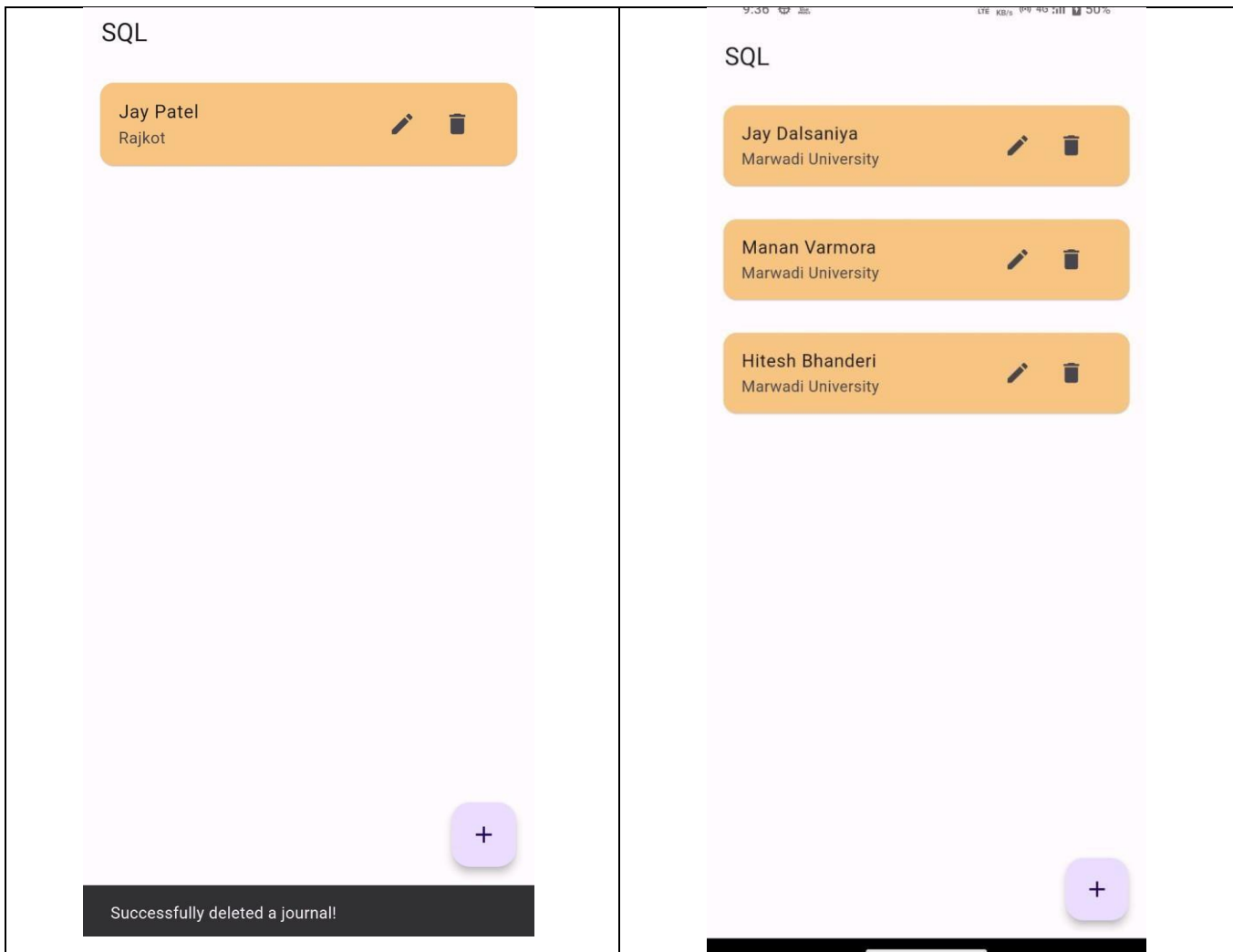
**sqlite: ^2.0.0**

**path: ^1.9.0**

**path\_provider: any**

**Output:**





---

## Practical 11 : Create and application Connecting to REST API in Flutter

### Main.dart

```
import 'package:flutter/material.dart';
import 'package:resetapi/data_screen.dart';

void main() {
  runApp(MyApp());
}

class MyApp extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      debugShowCheckedModeBanner: false,
      title: 'Flutter REST API Demo',
      theme: ThemeData(
        primarySwatch: Colors.blue,
      ),
      home: DataScreen(),
    );
  }
}
```

### api\_service.dart

```
import 'dart:convert';
import 'package:http/http.dart' as http;

class Post {
  final int userId;
  final int id;
  final String title;
  final String body;

  Post({
    required this.userId,
    required this.id,
    required this.title,
    required this.body,
  });

  factory Post.fromJson(Map<String, dynamic> json) {
    return Post(
      userId: json['userId'],
      id: json['id'],
      title: json['title'],
      body: json['body'],
    );
  }
}
```



```
}  
  
class ApiService {  
  static const String baseUrl = 'https://jsonplaceholder.typicode.com/todos/1';  
  
  static Future<List<Post>> fetchPosts() async {  
    final response = await http.get(Uri.parse('$baseUrl/posts'));  
  
    if (response.statusCode == 200) {  
      List<dynamic> jsonResponse = json.decode(response.body);  
      return jsonResponse.map((post) => Post.fromJson(post)).toList();  
    } else {  
      throw Exception('Failed to load posts');  
    }  
  }  
}
```

[data\\_screen.dart](#)

```
import 'package:flutter/material.dart';  
import 'package:resetapi/api_service.dart';  
  
class DataScreen extends StatefulWidget {  
  @override  
  _DataScreenState createState() => _DataScreenState();  
}  
  
class _DataScreenState extends State<DataScreen> {  
  late Future<List<Post>> posts;  
  
  @override  
  void initState() {  
    super.initState();  
    posts = ApiService.fetchPosts();  
  }  
  
  @override  
  Widget build(BuildContext context) {  
    return Scaffold(  
      appBar: AppBar(  
        title: Text('Posts'),  
      ),  
      body: Center(  
        child: FutureBuilder<List<Post>>(  
          future: posts,  
          builder: (context, snapshot) {  
            if (snapshot.hasData) {  
              return ListView.builder(  
                itemCount: snapshot.data!.length,  
                itemBuilder: (context, index) {  
                  return Card(  

```

```

        elevation: 3,
        margin: EdgeInsets.all(10),
        child: Padding(
          padding: EdgeInsets.all(10),
          child: Column(
            crossAxisAlignment: CrossAxisAlignment.start,
            children: [
              Text(
                'Post ${index + 1}:', // Add label here
                style: TextStyle(
                  fontWeight: FontWeight.bold,
                  fontSize: 16,
                ),
              ),
              SizedBox(height: 5),
              Text(
                snapshot.data![index].title,
                style: TextStyle(
                  fontWeight: FontWeight.bold,
                  fontSize: 18,
                ),
              ),
              SizedBox(height: 5),
              Text(snapshot.data![index].body),
            ],
          ),
        ),
      );
    },
  );
} else if (snapshot.hasError) { return
  Text("${snapshot.error}");
}

// By default, show a loading spinner.
return CircularProgressIndicator();
},
),
),
);
}
}

```

**dev\_dependencies:**

**flutter\_test:**

**sdk: flutter** **http:**

**^0.13.**

**Output:**

<p><b>Posts</b></p> <p><b>Post 10:</b> <b>optio molestias id quia eum</b> quo et expedita modi cum officia vel magni doloribus qui repudiandae vero nisi sit quos veniam quod sed accusamus veritatis error</p> <p><b>Post 11:</b> <b>et ea vero quia laudantium autem</b> delectus reiciendis molestiae occaecati non minima eveniet qui voluptatibus accusamus in eum beatae sit vel qui neque voluptates ut commodi qui incidunt ut animi commodi</p> <p><b>Post 12:</b> <b>in quibusdam tempore odit est dolorem</b> itaque id aut magnam praesentium quia et ea odit et ea voluptas et sapiente quia nihil amet occaecati quia id voluptatem incidunt ea est distinctio odio</p> <p><b>Post 13:</b> <b>dolorum ut in voluptas mollitia et saepe quo animi</b> aut dicta possimus sint mollitia voluptas commodi quo doloremque</p>	<p><b>Posts</b></p> <p><b>Post 36:</b> <b>fuga nam accusamus voluptas reiciendis itaque</b> ad mollitia et omnis minus architecto odit voluptas doloremque maxime aut non ipsa qui alias veniam blanditiis culpa aut quia nihil cumque facere et occaecati qui aspernatur quia eaque ut aperiam inventore</p> <p><b>Post 37:</b> <b>provident vel ut sit ratione est</b> debitis et eaque non officia sed nesciunt pariatur vel voluptatem iste vero et ea numquam aut expedita ipsum nulla in voluptates omnis consequatur aut enim officiis in quam qui</p> <p><b>Post 38:</b> <b>explicabo et eos deleniti nostrum ab id repellendus</b> animi esse sit aut sit nesciunt assumenda eum voluptas quia voluptatibus provident quia necessitatibus ea rerum repudiandae quia voluptatem delectus fugit aut id quia ratione optio eos iusto veniam iure</p> <p><b>Post 39:</b> <b>eos dolorem iste accusantium est eaque quam</b> corporis rerum ducimus vel eum accusantium</p>
---	---

---

## **Practical 12 : Create and application Parsing JSON data from REST API in Flutter.**

### **Main.dart**

```
import 'package:flutter/material.dart';
import 'package:resetapi/data_screen.dart';

void main() {
  runApp(MyApp());
}

class MyApp extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      debugShowCheckedModeBanner: false,
      title: 'Flutter REST API Demo',
      theme: ThemeData(
        primarySwatch: Colors.blue,
      ),
      home: DataScreen(),
    );
  }
}
```

### **api\_service.dart**

```
import 'dart:convert';
import 'package:http/http.dart' as http;

class Post {
  final int userId;
  final int id;
  final String title;
  final String body;

  Post({
    required this.userId,
    required this.id,
    required this.title,
    required this.body,
  });

  factory Post.fromJson(Map<String, dynamic> json) {
    return Post(
      userId: json['userId'],
      id: json['id'],
      title: json['title'],
      body: json['body'],
    );
  }
}
```

```
}  
}  
  
class ApiService {  
  static const String baseUrl = 'https://jsonplaceholder.typicode.com/todos/1';  
  
  static Future<List<Post>> fetchPosts() async {  
    final response = await http.get(Uri.parse('$baseUrl/posts'));  
  
    if (response.statusCode == 200) {  
      List<dynamic> jsonResponse = json.decode(response.body);  
      return jsonResponse.map((post) => Post.fromJson(post)).toList();  
    } else {  
      throw Exception('Failed to load posts');  
    }  
  }  
}
```

#### [data\\_screen.dart](#)

```
import 'package:flutter/material.dart';  
import 'package:resetapi/api_service.dart';  
  
class DataScreen extends StatefulWidget {  
  @override  
  _DataScreenState createState() => _DataScreenState();  
}  
  
class _DataScreenState extends State<DataScreen> {  
  late Future<List<Post>> posts;  
  
  @override  
  void initState() {  
    super.initState();  
    posts = ApiService.fetchPosts();  
  }  
  
  @override  
  Widget build(BuildContext context) {  
    return Scaffold(  
      appBar: AppBar(  
        title: Text('Posts'),  
      ),  
      body: Center(  
        child: FutureBuilder<List<Post>>(  
          future: posts,  
          builder: (context, snapshot) {  
            if (snapshot.hasData) {  
              return ListView.builder(  
                itemCount: snapshot.data!.length,  
                itemBuilder: (context, index) {
```

```

return Card(
  elevation: 3,
  margin: EdgeInsets.all(10),
  child: Padding(
    padding: EdgeInsets.all(10),
    child: Column(
      crossAxisAlignment: CrossAxisAlignment.start,
      children: [
        Text(
          'Post ${index + 1}:', // Add label here
          style: TextStyle(
            fontWeight: FontWeight.bold,
            fontSize: 16,
          ),
        ),
        SizedBox(height: 5),
        Text(
          snapshot.data![index].title,
          style: TextStyle(
            fontWeight: FontWeight.bold,
            fontSize: 18,
          ),
        ),
        SizedBox(height: 5),
        Text(snapshot.data![index].body),
      ],
    ),
  ),
);
},
);
);
} else if (snapshot.hasError) {
  return Text("${snapshot.error}");
}

// By default, show a loading spinner.
return CircularProgressIndicator();
},
),
),
);
}
}

```

[post\\_model.dart](#)

```

class Post {
  final int userId;
  final int id;
  final String title;
  final String body;
}

```

```
Post({
  required this.userId, required this.id, required this.title, required this.body,
});
```

```
factory Post.fromJson(Map<String, dynamic> json) { return Post(
  userId: json['userId'], id: json['id'],
  title: json['title'], body: json['body'],
);
}
```

```
dev_dependencies: flutter_test:
  sdk: flutter http: ^0.13.3
```

### Output:

#### Posts

##### Post 10:

##### optio molestias id quia eum

quo et expedita modi cum officia vel magni  
doloribus qui repudiandae  
vero nisi sit  
quos veniam quod sed accusamus veritatis error

##### Post 11:

##### et ea vero quia laudantium autem

delectus reiciendis molestiae occaecati non minima eveniet qui  
voluptatibus  
accusamus in eum beatae sit  
vel qui neque voluptates ut commodi qui incidunt  
ut animi commodi

##### Post 12:

##### in quibusdam tempore odit est dolore

itaque id aut magnam  
praesentium quia et ea odit et ea voluptas et  
sapiente quia nihil amet occaecati quia id voluptatem  
incidunt ea est distinctio odio

##### Post 13:

##### dolorum ut in voluptas mollitia et saepe quo animi

aut dicta possimus sint mollitia voluptas commodi quo doloremque

#### Posts

##### Post 36:

##### fuga nam accusamus voluptas reiciendis itaque

ad mollitia et omnis minus architecto odit  
voluptas doloremque maxime aut non ipsa qui alias veniam  
blanditiis culpa aut quia nihil cumque facere et occaecati  
qui aspernatur quia eaque ut aperiam inventore

##### Post 37:

##### provident vel ut sit ratione est

debitis et eaque non officia sed nesciunt pariatur vel  
voluptatem iste vero et ea  
numquam aut expedita ipsum nulla in  
voluptates omnis consequatur aut enim officiis in quam qui

##### Post 38:

##### explicabo et eos deleniti nostrum ab id repellendus

animi esse sit aut sit nesciunt assumenda eum voluptas  
quia voluptatibus provident quia necessitatibus ea  
rerum repudiandae quia voluptatem delectus fugit aut id quia  
ratione optio eos iusto veniam iure

##### Post 39:

##### eos dolore iste accusantium est eaque quam

corporis rerum ducimus vel eum accusantium

---

## **Practical 13 : Create and application using Hardware Interaction in Flutter**

### **Main.dart**

```
import 'package:flutter/material.dart';
import 'package:camera/camera.dart';

void main() {
  runApp(MyApp());
}

class MyApp extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      debugShowCheckedModeBanner: false,
      title: 'Flashlight App',
      theme: ThemeData(
        primarySwatch: Colors.blue,
        visualDensity: VisualDensity.adaptivePlatformDensity,
      ),
      home: FlashlightPage(),
    );
  }
}

class FlashlightPage extends StatefulWidget {
  @override
  _FlashlightPageState createState() => _FlashlightPageState();
}

class _FlashlightPageState extends State<FlashlightPage> {
  late CameraController _controller;
  bool _flashOn = false;

  @override
  void initState() {
    super.initState();
    _initializeCamera();
  }

  void _initializeCamera() async {
    final cameras = await availableCameras();
    final camera = cameras.firstWhere(
      (camera) => camera.lensDirection == CameraLensDirection.back,
    );

    _controller = CameraController(camera, ResolutionPreset.low);
    await _controller.initialize();

    // Set initial flash mode to off
    if (_controller.value.isInitialized) {
```



```
_controller.setFlashMode(FlashMode.off);
}
}

@override
void dispose() {
  _controller.dispose();
  super.dispose();
}

void _toggleFlashlight() {
  if (_controller.value.isInitialized) {
    final flashMode = _controller.value.flashMode;
    if (flashMode == FlashMode.off) {
      _controller.setFlashMode(FlashMode.torch);
      setState(() {
        _flashOn = true;
      });
    } else {
      _controller.setFlashMode(FlashMode.off);
      setState(() {
        _flashOn = false;
      });
    }
  }
}

@override
Widget build(BuildContext context) {
  return Scaffold(
    appBar: AppBar(
      title: Text('Flashlight'),
    ),
    body: Center(
      child: Column(
        mainAxisAlignment: MainAxisAlignment.center,
        children: [
          IconButton(
            icon: Icon(
              _flashOn ? Icons.flash_on : Icons.flash_off,
              size: 48,
              color: _flashOn ? Colors.yellow : Colors.grey,
            ),
            onPressed: _toggleFlashlight,
          ),
          SizedBox(height: 16),
          Text(
            _flashOn ? 'Flashlight On' : 'Flashlight Off',
            style: TextStyle(
              fontSize: 24,
              fontWeight: FontWeight.bold,
            ),
          ),
        ],
      ),
    ),
  );
}
```

```
),  
,  
],  
,  
,  
,  
};  
}  
}
```

**dependencies: flutter:**

**camera: ^0.10.5+9**

**Output:**

