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

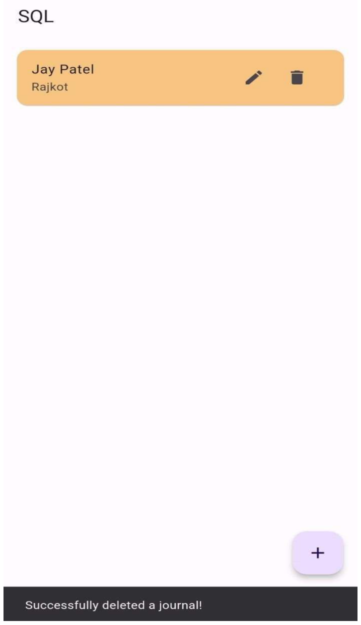
main.dart:

```
import 'package:flutter/material.dart';
import 'sql_helper.dart';
void main() => runApp(const MyApp());
class MyApp extends StatelessWidget {
  const MyApp({super.key});
  @override
  Widget build(BuildContext context) => MaterialApp(
    debugShowCheckedModeBanner: false,
    title: 'SQLite',   theme: ThemeData(primarySwatch: Colors.orange),
    home: const HomePage(),  );}
class HomePage extends StatefulWidget {
  const HomePage({super.key});
  @override
  State<HomePage> createState() => _HomePageState();}
class _HomePageState extends State<HomePage> {
  List<Map<String, dynamic>> _journals = [];
  bool _isLoading = true;
  final _titleController = TextEditingController();
  final _descController = TextEditingController();
  void _refresh() async {
    _journals = await SQLHelper.getItems();
    setState(() => _isLoading = false);
  }
  @override
  void initState() {
    super.initState();
    _refresh();
  }
  void _showForm(int? id) async {
    if (id != null) {
      final existing = _journals.firstWhere((e) => e['id'] == id);
      _titleController.text = existing['title'];
      _descController.text = existing['description'];
    }
    showModalBottomSheet(
      context: context,
      isScrollControlled: true,
      builder: (_) => Padding(
        padding: EdgeInsets.only(
          top: 15, left: 15, right: 15,
          bottom: MediaQuery.of(context).viewInsets.bottom + 120,
        ),
        child: Column(mainAxisSize: MainAxisSize.min, children: [
          TextField(controller: _titleController, decoration: const InputDecoration(hintText: 'Title')),
          const SizedBox(height: 10),
          TextField(controller: _descController, decoration: const InputDecoration(hintText: 'Description')),
          const SizedBox(height: 20),
          ElevatedButton(
            onPressed: () async {
              if (id == null) await SQLHelper.createItem(_titleController.text, _descController.text);
              else await SQLHelper.updateItem(id, _titleController.text, _descController.text);
              _titleController.clear();
              _descController.clear();
              Navigator.of(context).pop();
              _refresh();          },
            child: Text(id == null ? 'Create New' : 'Update'),
          ), ]),), );  }
  void _delete(int id) async {
    await SQLHelper.deleteItem(id);
    ScaffoldMessenger.of(context).showSnackBar(const SnackBar(content: Text('Deleted successfully')));
    _refresh();
  }
  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(title: const Text('SQLite')),
      body: _isLoading
          ? const Center(child: CircularProgressIndicator())
          : ListView.builder(
              itemCount: _journals.length,
              itemBuilder: (_, i) => Card(
                color: Colors.orange[200],
                margin: const EdgeInsets.all(10),
                child: ListTile(
                  title: Text(_journals[i]['title']),
                  subtitle: Text(_journals[i]['description']),
                  trailing: Row(mainAxisSize: MainAxisSize.min, children: [
                    IconButton(icon: const Icon(Icons.edit), onPressed: () => _showForm(_journals[i]['id'])),
                    IconButton(icon: const Icon(Icons.delete), onPressed: () => _delete(_journals[i]['id'])),
                  ]),
                ),
              ),
            ),
      floatingActionButton: FloatingActionButton(
        onPressed: () => _showForm(null),
        child: const Icon(Icons.add),
      ),
    );
  }
}
```

sqlHelper.dart:

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

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

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

  static Future<int> createItem(String title, String? desc) async {
    final db = await SQLHelper.db();
    return db.insert('items', {'title': title, 'description': desc}, conflictAlgorithm: sql.ConflictAlgorithm.replace);
  }

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

  static Future<int> updateItem(int id, String title, String? desc) async {
    final db = await SQLHelper.db();
    return db.update('items', {
      'title': title,
      'description': desc,
      'createdAt': DateTime.now().toString(),
    }, where: "id = ?", whereArgs: [id]);
  }

  static Future<void> deleteItem(int id) async {
    final db = await SQLHelper.db();
    try {
      await db.delete("items", where: "id = ?", whereArgs: [id]);
    } catch (e) {
      debugPrint("Delete error: $e");
    }
  }
}dependencies: flutter:
sdk: flutter sqflite: ^2.0.0 path: ^1.9.0
path_provider: any
```

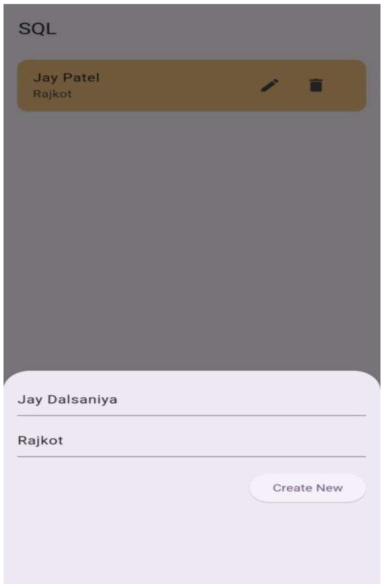
Output:



SQL

Jay Patel
Rajkot

Successfully deleted a journal!

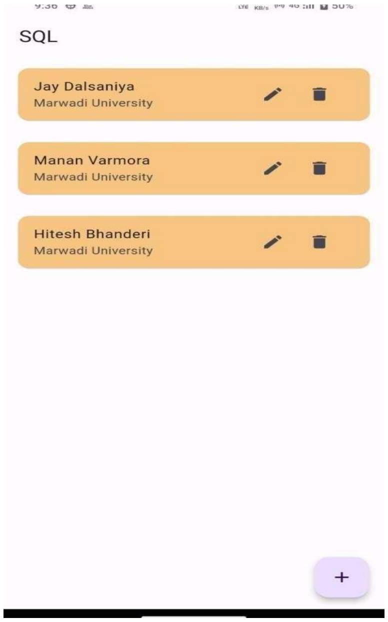


SQL

Jay Patel
Rajkot

Jay Dalsaniya

Rajkot



SQL

Jay Dalsaniya
Marwadi University

Manan Varmora
Marwadi University

Hitesh Bhanderi
Marwadi University

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

main.dart:

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

void main() => runApp(MaterialApp(
  debugShowCheckedModeBanner: false,
  home: DataScreen(),
 ));
```

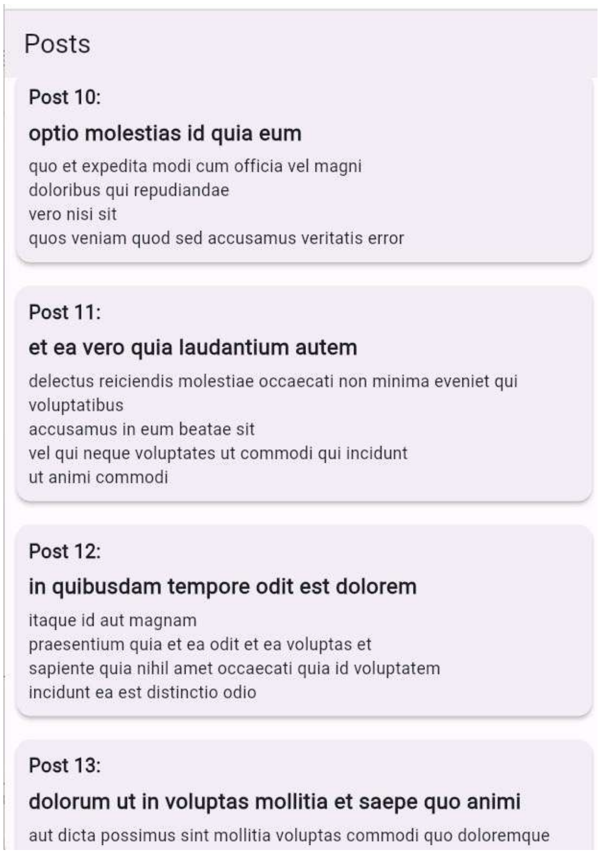
api_service.dart:

```
import 'dart:convert';
import 'package:http/http.dart' as http;
class Post {
  final int userId, id;
  final String title, body;
  Post({required this.userId, required this.id, required this.title, required this.body});
  factory Post.fromJson(Map<String, dynamic> json) => Post(
    userId: json['userId'],
    id: json['id'],
    title: json['title'],
    body: json['body'],
  );}
class ApiService {
  static const _url = 'https://jsonplaceholder.typicode.com/posts';
  static Future<List<Post>> fetchPosts() async {
    final res = await http.get(Uri.parse(_url));
    if (res.statusCode == 200) {
      final data = json.decode(res.body) as List;
      return data.map((e) => Post.fromJson(e)).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();
},),),);}}
```

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:

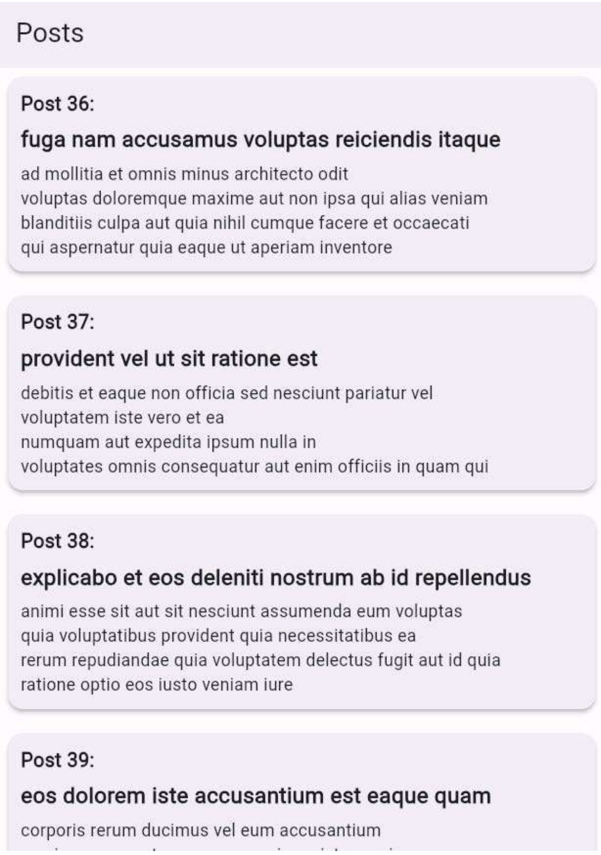
in quibusdam tempore odit est dolorem

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 dolorem iste accusantium est eaque quam

corporis rerum ducimus vel eum accusantium

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

main.dart:

```
import 'package:flutter/material.dart';
import 'data_screen.dart';
void main() => runApp(const MaterialApp(
  debugShowCheckedModeBanner: false,
  home: DataScreen(),
 ));
```

api_service.dart:

```
import 'dart:convert';
import 'package:http/http.dart' as http;
import 'post_model.dart';
class ApiService {
  static Future<List<Post>> fetchPosts() async {
    final res = await http.get(Uri.parse('https://jsonplaceholder.typicode.com/posts'));
    if (res.statusCode == 200) {
      return (jsonDecode(res.body) as List).map((e) => Post.fromJson(e)).toList();
    } else {
      throw Exception('Failed to load posts');
    }  }}
```

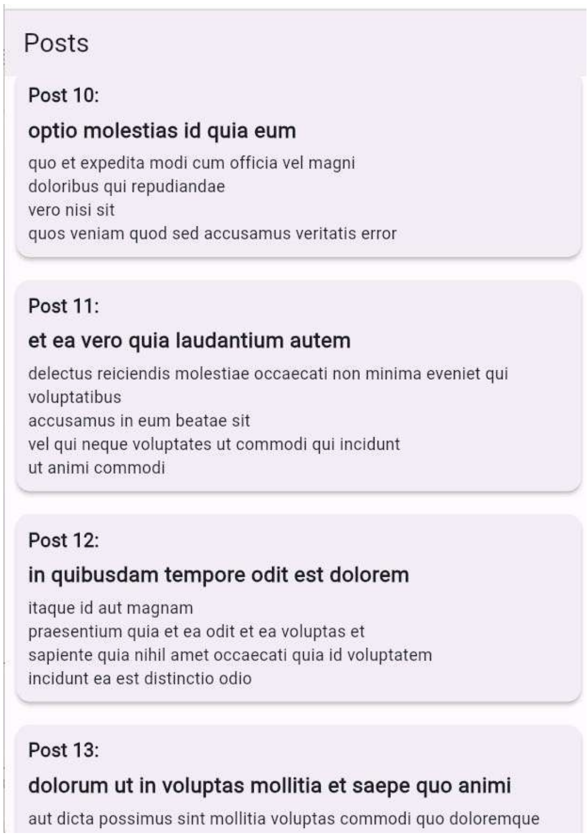
data_screen.dart:

```
import 'package:flutter/material.dart';
import 'api_service.dart';
import 'post_model.dart';
class DataScreen extends StatelessWidget {
  const DataScreen({super.key});
  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(title: const Text('Posts')),
      body: FutureBuilder<List<Post>>(
        future: ApiService.fetchPosts(),
        builder: (context, snapshot) {
          if (snapshot.hasData) {
            return ListView.builder(
              itemCount: snapshot.data!.length,
              padding: const EdgeInsets.all(10),
              itemBuilder: (context, i) {
                final post = snapshot.data![i];
                return Card(margin: const EdgeInsets.symmetric(vertical: 8),
                  child: Padding( padding: const EdgeInsets.all(10),
                    child: Column(crossAxisAlignment: CrossAxisAlignment.start,
                      children: [ Text('Post ${i + 1}', style: const TextStyle(fontWeight: FontWeight.bold)),
                        const SizedBox(height: 5),Text(post.title, style: const TextStyle(fontSize: 16, fontWeight:
FontWeight.bold)),
                        const SizedBox(height: 5),
                        Text(post.body), ], ),), ); },);
          } else if (snapshot.hasError) {return Center(child: Text('${snapshot.error}'));
          }return const Center(child: CircularProgressIndicator());
        },),);}}
```

post_model.dart:

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

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

Output:



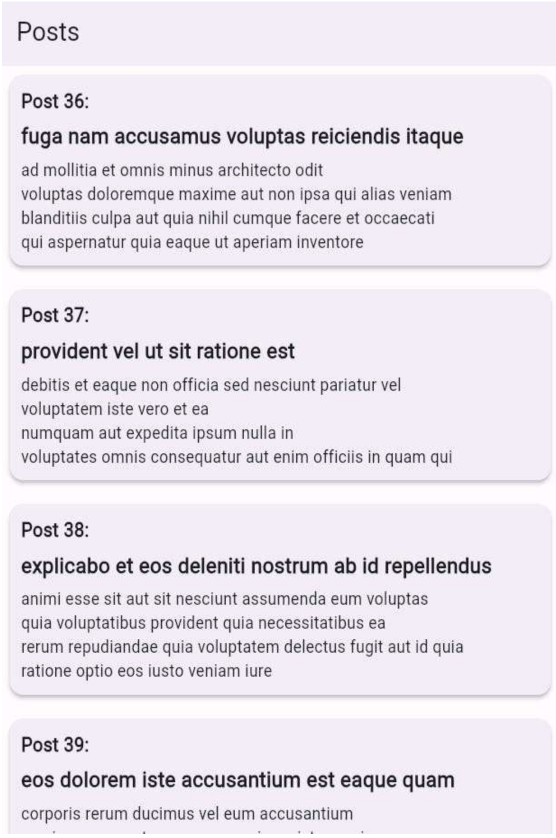
Posts

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 dolorem
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 dolorem iste accusantium est eaque quam
corporis rerum ducimus vel eum accusantium

Practical 13: Create and application using Hardware Interaction in Flutter.

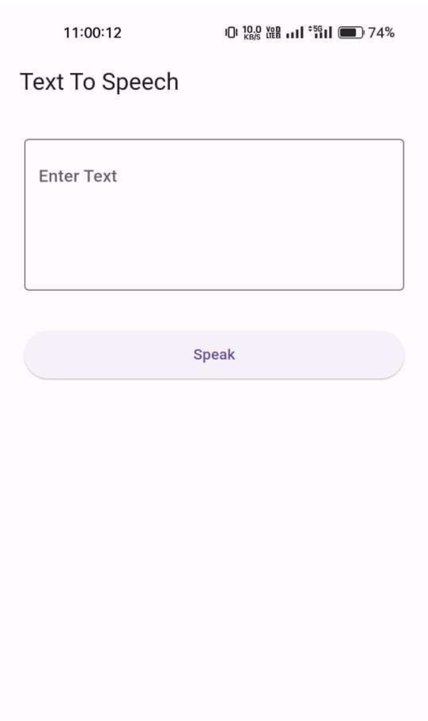
main.dart:

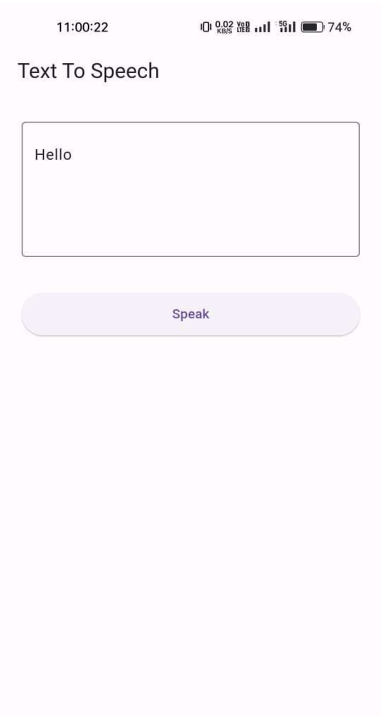
```
import 'package:flutter/material.dart';
import 'home_screen.dart';
void main(){
  runApp(MyApp());
}

class MyApp extends StatelessWidget {
  const MyApp({super.key});

  @override
  Widget build(BuildContext context) {
    return MaterialApp(
    debugShowCheckedModeBanner: false,
    title: "Text To Speech",
      theme: ThemeData(
        primarySwatch: Colors.indigo,
      ),
      home: HomeScreen(),
    );
  }
}
```

homescreen.dart:

```
import 'dart:async';
import 'package:flutter/material.dart';
import 'package:flutter_tts/flutter_tts.dart';

class HomeScreen extends StatefulWidget {
  const HomeScreen({super.key});

  @override
  State<HomeScreen> createState() => _HomeScreenState();
}

class _HomeScreenState extends State<HomeScreen> {
  final FlutterTts flutterTts = FlutterTts();
  final TextEditingController textController = TextEditingController();

  @override
  void dispose() {
    textController.dispose();
    super.dispose();
  }
  Future<void> speak(String text) async{
    await  flutterTts.setLanguage('en-US');
    await flutterTts.setPitch(1.0);
    await flutterTts.setSpeechRate(0.5);
    await flutterTts.speak(text);
  }
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
        title: Text("Text To Speech"),
      ),
      body: Padding(
        padding: EdgeInsets.all(20),
        child: Column(
          crossAxisAlignment: CrossAxisAlignment.stretch,
          children: [
            TextField(
              controller: textController,
              decoration: InputDecoration(
              hintText: 'Enter Text',
                border: OutlineInputBorder(),
              ),
              maxLines: 4,
            ),
            SizedBox(height: 30,),
            ElevatedButton(onPressed: () {
            speak(textController.text);
            },
              child: Text('Speak'),
            ),
          ],
        ),
      ),
    );
  }
}
```

Output:





Hello