Practical No.18

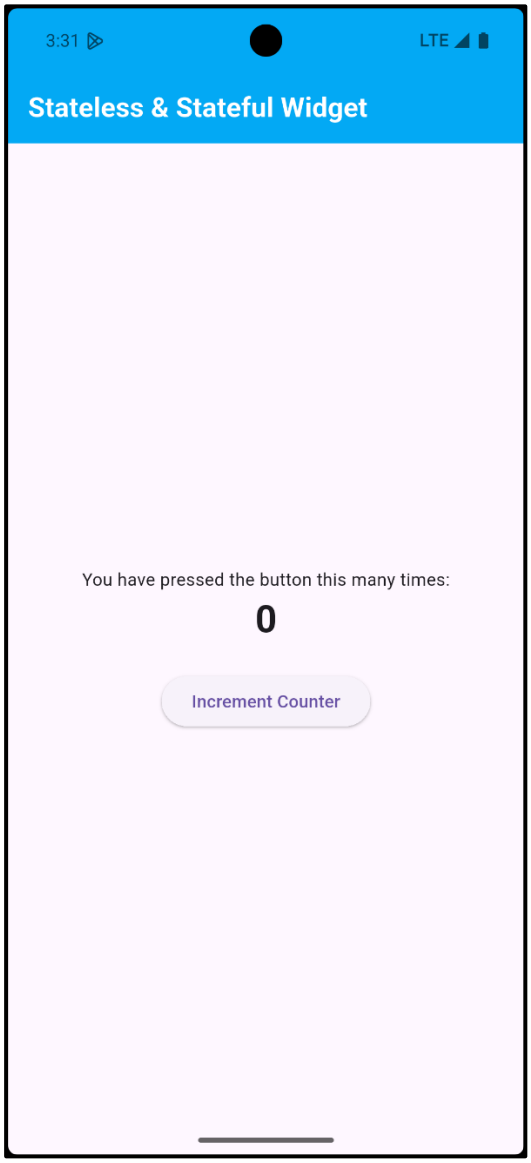
Aim: Flutter Program based on Stateful and Stateless Widgets.

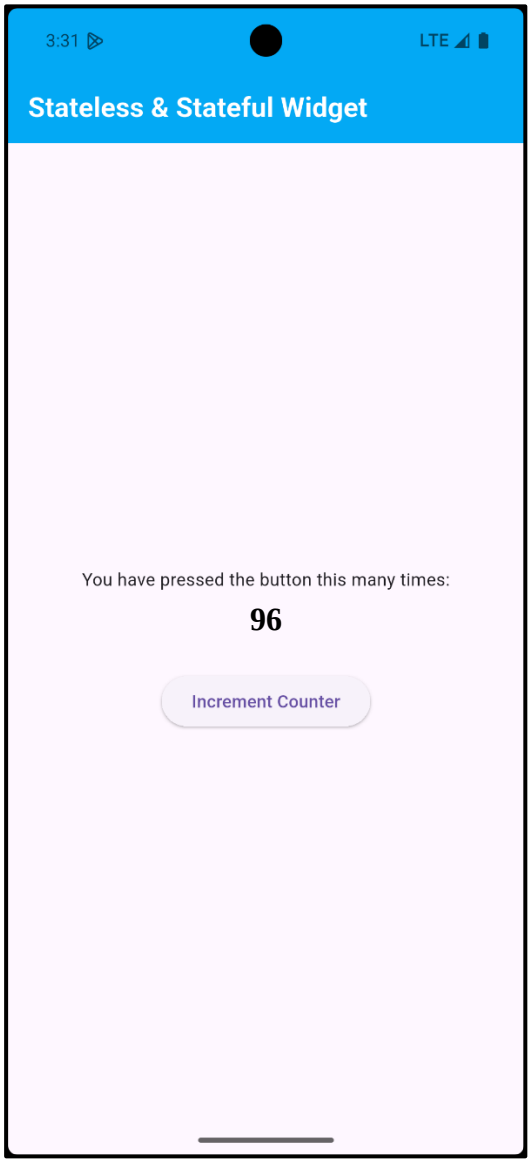
main.dart

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

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

// StatelessWidget: The root of the application
class MyApp extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      debugShowCheckedModeBanner: false,
      title: 'Stateless vs Stateful Demo',
      home: HomePage(),
    );
  }
}

// StatelessWidget: Represents the UI layout
class HomePage extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
        title: Text(
          'Stateless & Stateful Widget',
          style: TextStyle(fontWeight: FontWeight.bold, color: Colors.white),
        ),
        backgroundColor: Colors.lightBlue,
      ),
      body: Center(
        child: CounterWidget(), // Our StatefulWidget
      ),
    );
  }
}

// StatefulWidget: Holds the state (e.g., counter)
class CounterWidget extends StatefulWidget {
  @override
```

```
  _CounterWidgetState createState() => _CounterWidgetState();
}

// The state class that changes over time
class _CounterWidgetState extends State<CounterWidget> {
 int _counter = 0;

 void _incrementCounter() {
  setState(() {
   _counter++; // Update the state
  });
 }

 @override
 Widget build(BuildContext context) {
  return Column(
   mainAxisAlignment: MainAxisAlignment.center,
   children: [
    Text('You have pressed the button this many times:'),
    Text(
     '$_counter',
     style: TextStyle(fontSize: 30, fontWeight: FontWeight.bold),
    ),
    SizedBox(height: 20),
    ElevatedButton(
     onPressed: _incrementCounter,
     child: Text('Increment Counter'),
    ),
   ],
  );
 }
}
```

OUTPUT:


3:31
LTE
Stateless & Stateful Widget
You have pressed the button this many times:
0
Increment Counter


3:31
LTE
Stateless & Stateful Widget
You have pressed the button this many times:
96
Increment Counter

Practical No.19

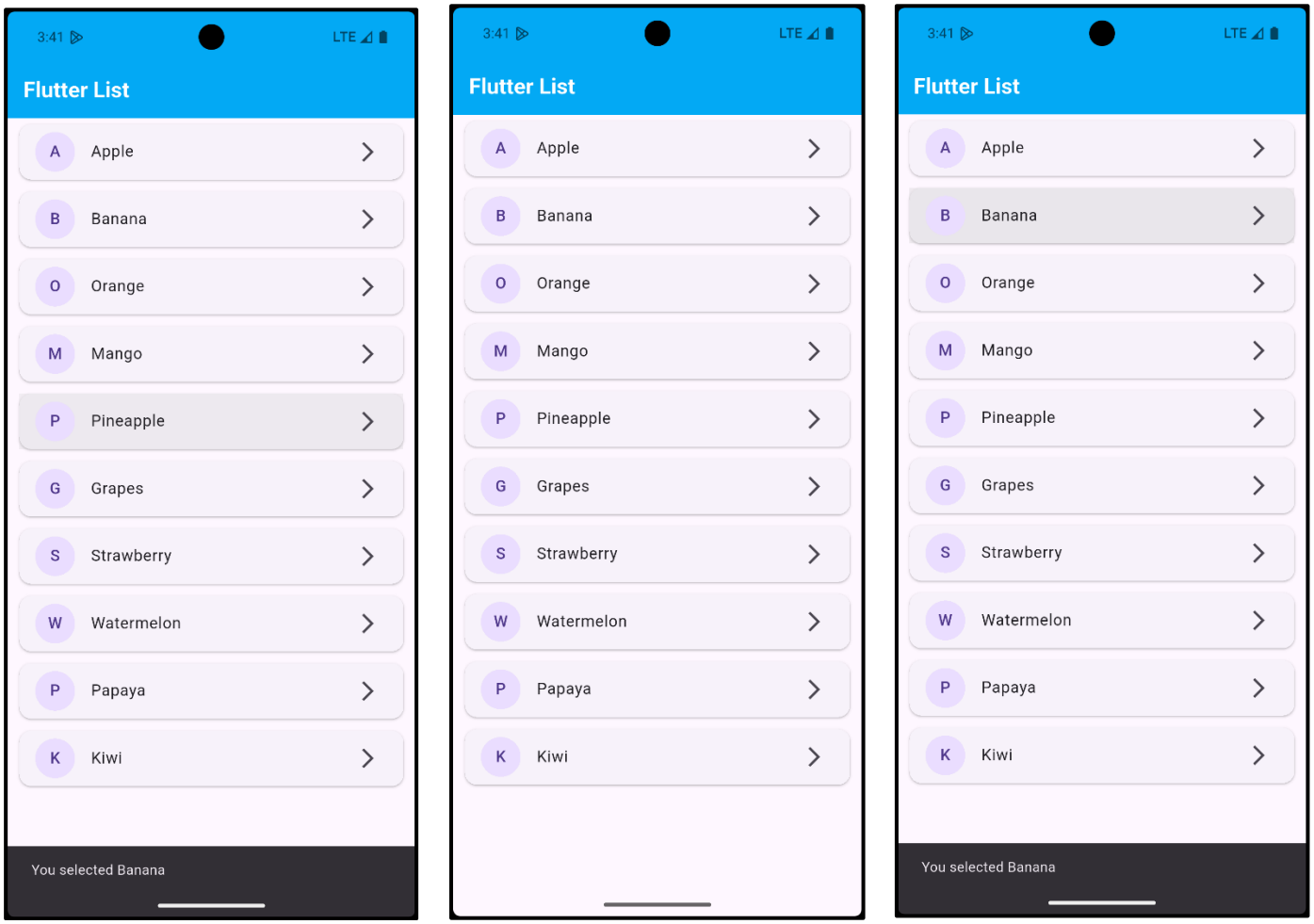
Aim: Flutter Program using List

main.dart

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

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

class MyApp extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      debugShowCheckedModeBanner: false,
      title: 'List Example',
      theme: ThemeData(primarySwatch: Colors.green),
      home: FruitListScreen(),
    ); }}

class FruitListScreen extends StatelessWidget {
  final List<String> fruits = [
    'Apple', 'Banana', 'Orange', 'Mango', 'Pineapple', 'Grapes', 'Strawberry', 'Watermelon',
'Papaya', 'Kiwi', ];

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
        title: Text(
          'Flutter List',
          style: TextStyle(fontWeight: FontWeight.bold, color: Colors.white),
        ),
        backgroundColor: Colors.lightBlue,
      ),
      body: ListView.builder(
        itemCount: fruits.length,
        itemBuilder: (context, index) {
          return Card(
            margin: EdgeInsets.symmetric(vertical: 6, horizontal: 12),
            child: ListTile(
              leading: CircleAvatar(child: Text(fruits[index][0])),
              title: Text(fruits[index]),
              trailing: Icon(Icons.arrow_forward_ios),
              onTap: () {
```

```
ScaffoldMessenger.of(context).showSnackBar(
  SnackBar(content: Text('You selected ${fruits[index]}')),
); }, ),); },),);}}
```

OUTPUT:



Practical No.20

Aim: Flutter Program using TextField, Check Box, Buttons, Drop down, Switch etc.

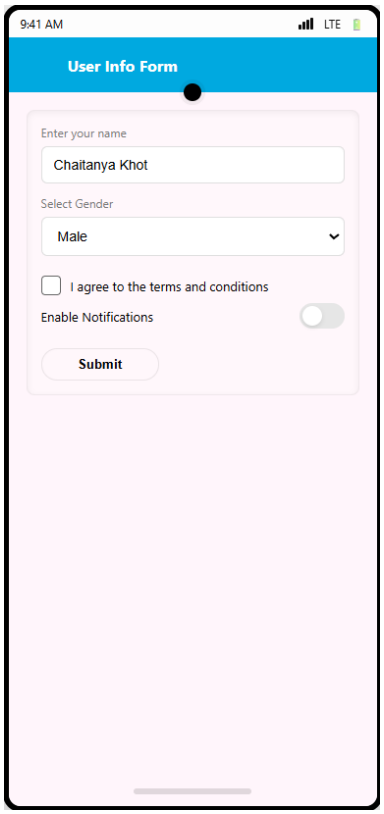
main.dart

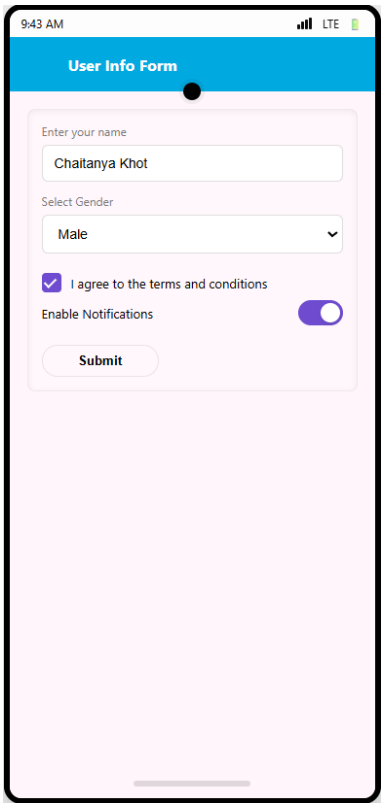
```
import 'package:flutter/material.dart';

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

class MyApp extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      title: 'Flutter Form Example',
      theme: ThemeData(primarySwatch: Colors.blue),
      home: FormScreen(),
    );
  }
}

class FormScreen extends StatefulWidget {
  @override
  _FormScreenState createState() => _FormScreenState();
}

class _FormScreenState extends State<FormScreen> {
  String name = '';
  bool agreeToTerms = false;
  bool notificationsEnabled = false;
  String selectedGender = 'Male';
  String result = '';
  final TextEditingController nameController = TextEditingController();

  final List<String> genderOptions = ['Male', 'Female', 'Other'];

  void handleSubmit() {
    setState(() {
      name = nameController.text;
      result =
          'Name: $name\nGender: $selectedGender\nAgreed to Terms: ${agreeToTerms ? "Yes" :
"No"}\nNotifications: ${notificationsEnabled ? "Enabled" : "Disabled"}';
    });
```

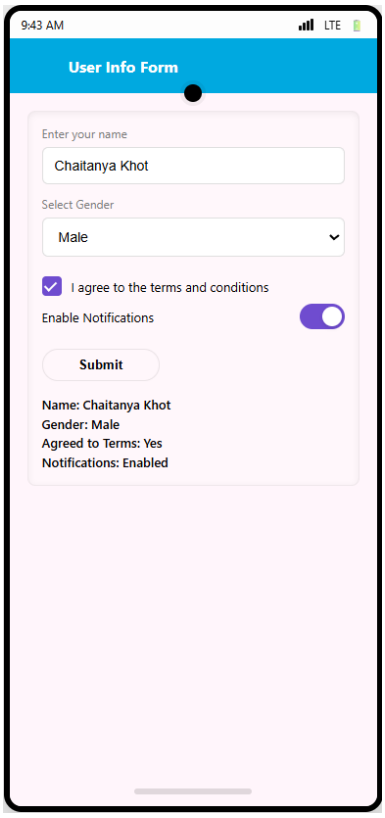
```
}

@override
Widget build(BuildContext context) {
 return Scaffold(
  appBar: AppBar(
   title: Text(
    'User Info Form',
    style: TextStyle(fontWeight: FontWeight.bold, color: Colors.white),
   ),
   backgroundColor: Colors.lightBlue,
  ),
  body: SingleChildScrollView(
   padding: EdgeInsets.all(16.0),
   child: Column(
    crossAxisAlignment: CrossAxisAlignment.start,
    children: [
     TextField(
      controller: nameController,
      decoration: InputDecoration(
       labelText: 'Enter your name',
       border: OutlineInputBorder(),
      ),
     ),
     SizedBox(height: 16),
     DropdownButtonFormField<String>(
      value: selectedGender,
      items: genderOptions
        .map(
         (gender) =>
           DropdownMenuItem(value: gender, child: Text(gender)),
        )
        .toList(),
      onChanged: (value) {
       setState(() {
        selectedGender = value!;
       });
      },
      decoration: InputDecoration(
       labelText: 'Select Gender',
       border: OutlineInputBorder(),
      ),
     ),
     SizedBox(height: 16),
     CheckboxListTile(
      title: Text('I agree to the terms and conditions'),
      value: agreeToTerms,
      onChanged: (value) {
```

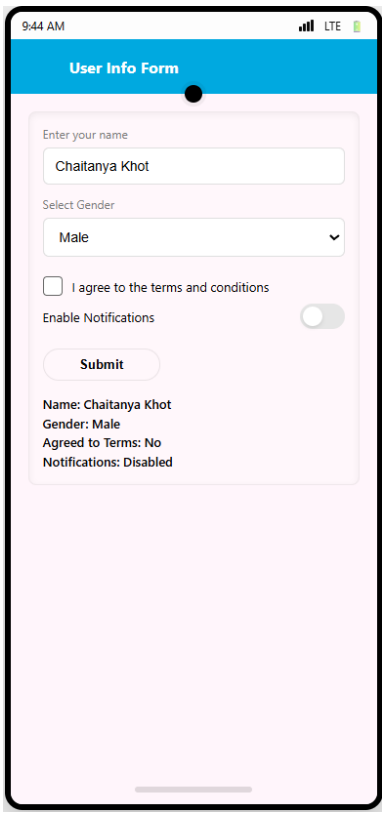
```
              setState(() {
                agreeToTerms = value!;
              });
            },
          ),
          SwitchListTile(
            title: Text('Enable Notifications'),
            value: notificationsEnabled,
            onChanged: (value) {
              setState(() {
                notificationsEnabled = value;
              });
            },
          ),
          Center(
            child: ElevatedButton(
              onPressed: handleSubmit,
              child: Text('Submit'),
            ),
          ),
          SizedBox(height: 20),
          if (result.isNotEmpty)
            Text(
              result,
              style: TextStyle(fontSize: 16, fontWeight: FontWeight.bold),
            ),
        ],
      ),
    ),
  );
 }
}
```

OUTPUT:


9:41 AM
LTE
User Info Form
Enter your name
Chaitanya Khot
Select Gender
Male
I agree to the terms and conditions
Enable Notifications
Submit


9:43 AM
LTE
User Info Form
Enter your name
Chaitanya Khot
Select Gender
Male
I agree to the terms and conditions
Enable Notifications
Submit


9:43 AM
LTE
User Info Form
Enter your name
Chaitanya Khot
Select Gender
Male
I agree to the terms and conditions
Enable Notifications
Submit
Name: Chaitanya Khot
Gender: Male
Agreed to Terms: Yes
Notifications: Enabled


9:44 AM
LTE
User Info Form
Enter your name
Chaitanya Khot
Select Gender
Male
I agree to the terms and conditions
Enable Notifications
Submit
Name: Chaitanya Khot
Gender: Male
Agreed to Terms: No
Notifications: Disabled

Practical No.21

Aim: Program to demonstrate the use of SQFlite Database using flutter

main.dart

```
import 'package:flutter/material.dart';
import 'database_helper.dart';
import 'user.dart';

void main() {
 WidgetsFlutterBinding.ensureInitialized();
 runApp(const MyApp());
}
class MyApp extends StatelessWidget {
 const MyApp({super.key});
 @override
 Widget build(BuildContext context) {
  return MaterialApp(
   debugShowCheckedModeBanner: false,
   title: 'Flutter SQFlite',
   home: const UserListScreen(),
  ); }}

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

 @override  State<UserListScreen> createState() => _UserListScreenState();}

class _UserListScreenState extends State<UserListScreen> {
 final DatabaseHelper _dbHelper = DatabaseHelper();
 List<User> _users = [];
 @override
 void initState() {
  super.initState();
  _loadUsers();  }

 Future<void> _loadUsers() async {
  final users = await _dbHelper.getUsers();
  setState(() {
   _users = users;   }); }

 Future<void> _addUser() async {
  final newUser = User(
   name: 'New User ${DateTime.now().millisecond}',
   age: 30,   );
  await _dbHelper.insertUser(newUser);
  _loadUsers();  }
```

```
Future<void> _updateUser(User user) async {
 final updatedUser = User(
  id: user.id,
  name: '${user.name} (Updated)',
  age: user.age + 1,   );
 await _dbHelper.updateUser(updatedUser);
 _loadUsers(); }

Future<void> _deleteUser(int id) async {
 await _dbHelper.deleteUser(id);
 _loadUsers(); }

@override
Widget build(BuildContext context) {
 return Scaffold(
  appBar: AppBar(
   title: const Text('Flutter SQFlite User List'),
   backgroundColor: Colors.lightBlue,
   actions: [IconButton(icon: const Icon(Icons.add), onPressed: _addUser)],     ),
  body: ListView.builder(
   itemCount: _users.length,
   itemBuilder: (context, index) {
    final user = _users[index];
    return ListTile(
     title: Text(user.name),
     subtitle: Text('Age: ${user.age}'),
     trailing: Row(
      mainAxisSize: MainAxisSize.min,
      children: [
       IconButton(
        icon: const Icon(Icons.edit),
        onPressed: () => _updateUser(user),
       ),
       IconButton(
        icon: const Icon(Icons.delete),
        onPressed: () => _deleteUser(user.id!),
      ),          ],         ),       );      },    ),   ); }}
```

user.dart

```
class User {
 final int? id;
 final String name;
 final int age;

 User({this.id, required this.name, required this.age});
```

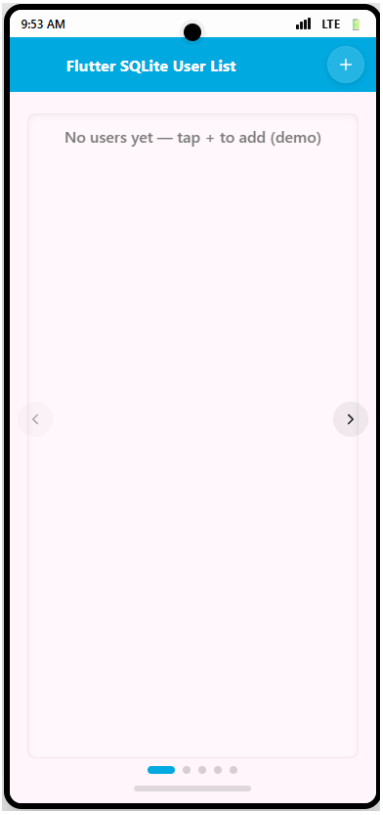
```
 Map<String, dynamic> toMap() {
  return {'id': id, 'name': name, 'age': age};
 }

 factory User.fromMap(Map<String, dynamic> map) {
  return User(id: map['id'], name: map['name'], age: map['age']);
 }

 @override
 String toString() {
  return 'User{id: $id, name: $name, age: $age}';
 }}
```

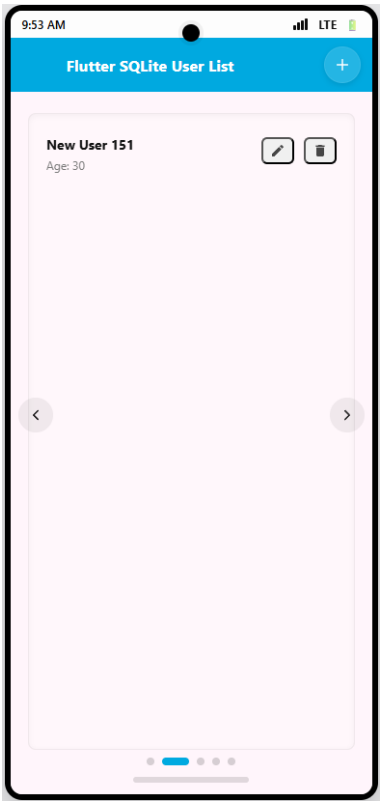
database_helper.dart

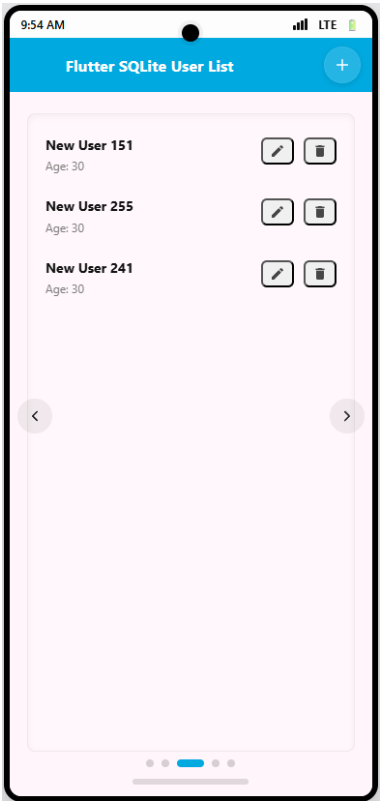
```
import 'dart:async';
import 'package:path/path.dart';
import 'package:sqflite/sqflite.dart';
import 'user.dart'; // Import your User model

class DatabaseHelper {
 static final DatabaseHelper _instance = DatabaseHelper._internal();
 static Database? _database;

 factory DatabaseHelper() {
  return _instance;
 }

 DatabaseHelper._internal();

 Future<Database> get database async {
  if (_database != null) return _database!;
  _database = await _initDatabase();
  return _database!;
 }

 Future<Database> _initDatabase() async {
  String path = join(await getDatabasesPath(), 'user_database.db');
  return await openDatabase(
   path,
   version: 1,
   onCreate: (db, version) {
    return db.execute(
     'CREATE TABLE users(id INTEGER PRIMARY KEY AUTOINCREMENT, name TEXT, age INTEGER)',
    );
   },
  );
 }
```

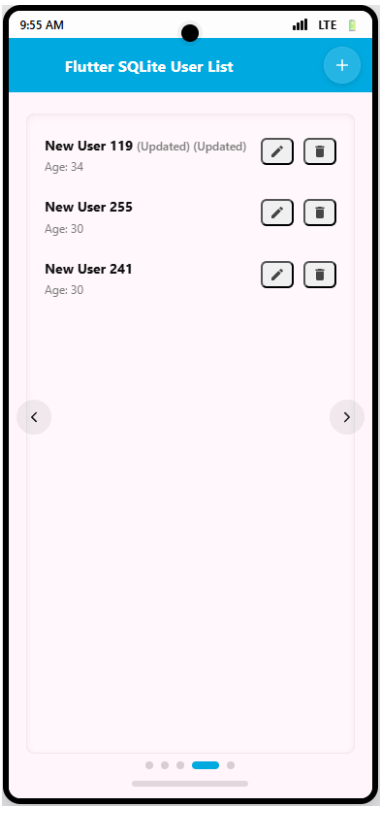
```
  // Insert a user
  Future<int> insertUser(User user) async {
    final db = await database;
    return await db.insert(
      'users',
      user.toMap(),
      conflictAlgorithm: ConflictAlgorithm.replace,
    );
  }

  // Retrieve all users
  Future<List<User>> getUsers() async {
    final db = await database;
    final List<Map<String, dynamic>> maps = await db.query('users');
    return List.generate(maps.length, (i) {
      return User.fromMap(maps[i]);
    });
  }

  // Update a user
  Future<int> updateUser(User user) async {
    final db = await database;
    return await db.update(
      'users',
      user.toMap(),
      where: 'id = ?',
      whereArgs: [user.id],
    );
  }

  // Delete a user
  Future<int> deleteUser(int id) async {
    final db = await database;
    return await db.delete('users', where: 'id = ?', whereArgs: [id]);
  }
}
```

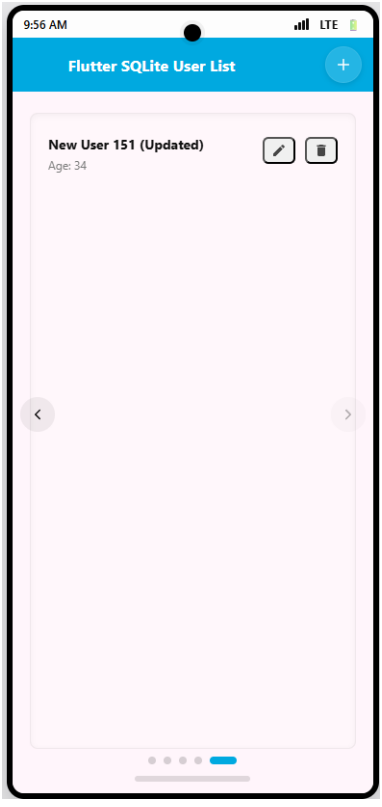
OUTPUT:


9:53 AM
LTE
Flutter SQLite User List
No users yet — tap + to add (demo)


9:53 AM
LTE
Flutter SQLite User List
New User 151
Age: 30


9:54 AM
LTE
Flutter SQLite User List
New User 151
Age: 30
New User 255
Age: 30
New User 241
Age: 30


9:55 AM
LTE
Flutter SQLite User List
New User 119 (Updated) (Updated)
Age: 34
New User 255
Age: 30
New User 241
Age: 30


9:56 AM
LTE
Flutter SQLite User List
New User 151 (Updated)
Age: 34

Practical No.22

Aim: Program to demonstrate the use of RESTAPI

main.dart

```
import 'package:flutter/material.dart';
import 'package:flutter_api/post.dart';
import 'package:flutter_api/post_service.dart';

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

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

 @override
 Widget build(BuildContext context) {
  return MaterialApp(
   title: 'Flutter REST API',
   theme: ThemeData(primarySwatch: Colors.blue),
   home: const PostsPage(),
  ); }}

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

 @override
 State<PostsPage> createState() => _PostsPageState();}

class _PostsPageState extends State<PostsPage> {
 late Future<List<Post>> futurePosts;
 @override
 void initState() {
  super.initState();
  futurePosts = fetchPosts(); }

 @override
 Widget build(BuildContext context) {
  return Scaffold(
   appBar: AppBar(title: const Text('Posts')),
   body: FutureBuilder<List<Post>>(
    future: futurePosts,
    builder: (context, snapshot) {
     if (snapshot.connectionState == ConnectionState.waiting) {
      return const Center(child: CircularProgressIndicator());
```

```
        } else if (snapshot.hasError) {
          return Center(child: Text('Error: ${snapshot.error}'));
        } else if (!snapshot.hasData || snapshot.data!.isEmpty) {
          return const Center(child: Text('No posts found'));
        } else {
          final posts = snapshot.data!;
          return ListView.builder(
            itemCount: posts.length,
            itemBuilder: (context, index) {
              final post = posts[index];
              return ListTile(
                title: Text(post.title),
                subtitle: Text(post.body),
              );          },          );        }      },    );  }}
```

post.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'],
    );  }}
```

post_services.dart

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

Future<List<Post>> fetchPosts() async {
  try {
    print('Fetching posts from API...');
    final response = await http
        .get(
          Uri.parse('https://jsonplaceholder.typicode.com/posts'),
          headers: {'Content-Type': 'application/json'},       )
```

```
      .timeout(Duration(seconds: 10));

  print('Response status code: ${response.statusCode}');
  print('Response body length: ${response.body.length}');

  if (response.statusCode == 200) {
    final List<dynamic> jsonData = json.decode(response.body);
    print('Successfully parsed ${jsonData.length} posts');
    return jsonData.map((item) => Post.fromJson(item)).toList();
  } else {
    throw Exception(
      'Failed to load posts. Status code: ${response.statusCode}, Body: ${response.body}',
    );   } } catch (e) {
  print('Error in fetchPosts: $e');
  throw Exception('Failed to load posts: $e');
}}
```

OUTPUT: