



Experiment :- 01

AIM:- Android Studio setup for Flutter development with along with Dart SDK.

Step1:InstallingFlutter.

i. System Requirements:

- Assurethatyoursystemmeetsthemminimumrequirements.FluttersupportsmacOS, Linux, and Windows.
- OnmacOS,youneedXcodewiththecommand-linetoolsinstalled.
- OnLinux,youneedtohavegit,lib32stdc++6,andotherdependenciesinstalled.

ii. DownloadFlutter:

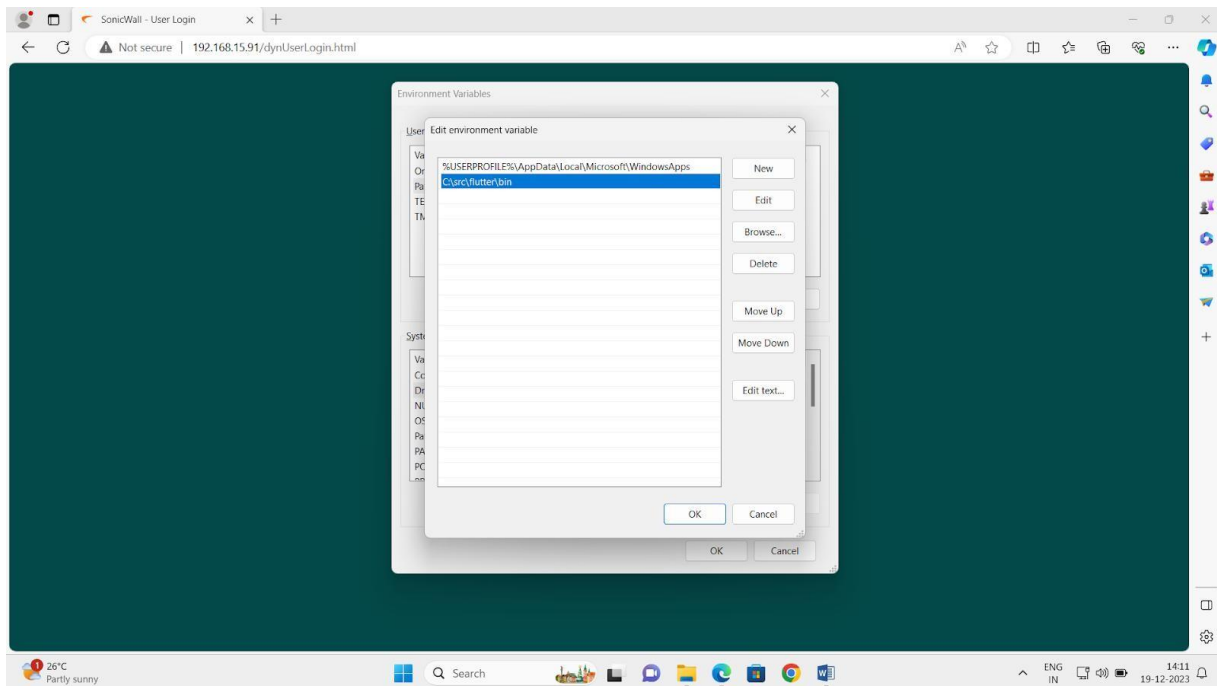
- VisitFlutterWebsiteforInstallationofFlutter-><https://docs.flutter.dev/get-started/install>.

iii. ExtractFlutter:

- IfyoudownloadedtheZIPfile,extractittoalocationonyourmachine.(C:\src\flutter).

iv. SetUpEnvironmentVariables:

- AddtheC:\src\flutter\bindirectorytoyoursystem's PATHvariable.



v. Runflutterdoctor:

- Open a terminal and run the following command: `flutterdoctor`
- This command checks your environment and displays a report of any missing dependencies or issues.

vi. InstallFlutterDependencies:

- Follow the instructions provided by flutterdoctor to install any missing dependencies. This may include things like Android Studio, Xcode command-line tools, etc.

Step2: Installing Android Studio.

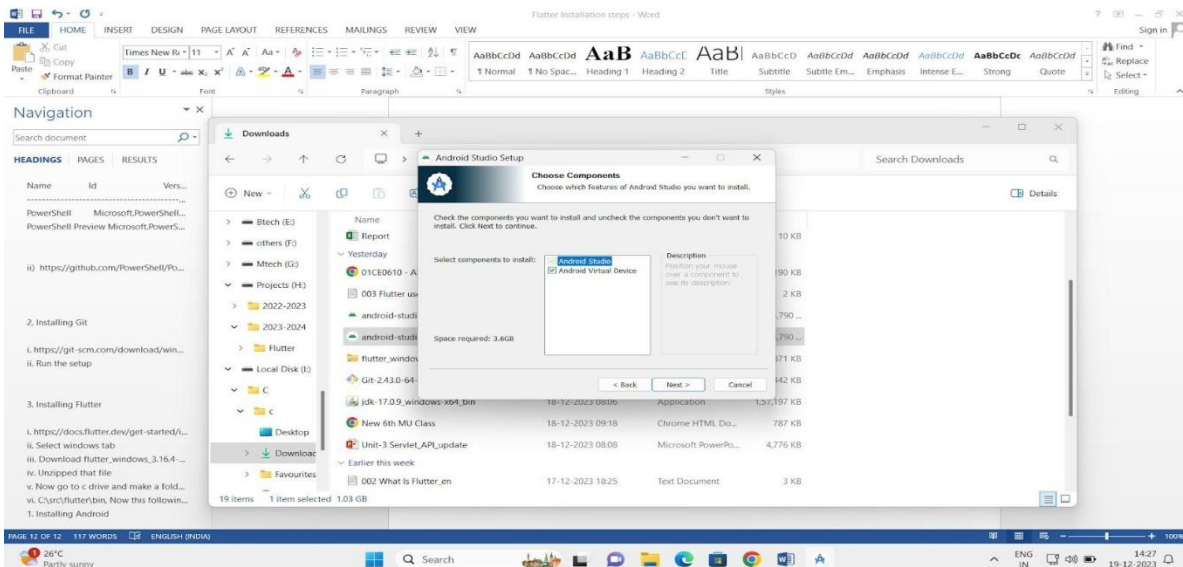
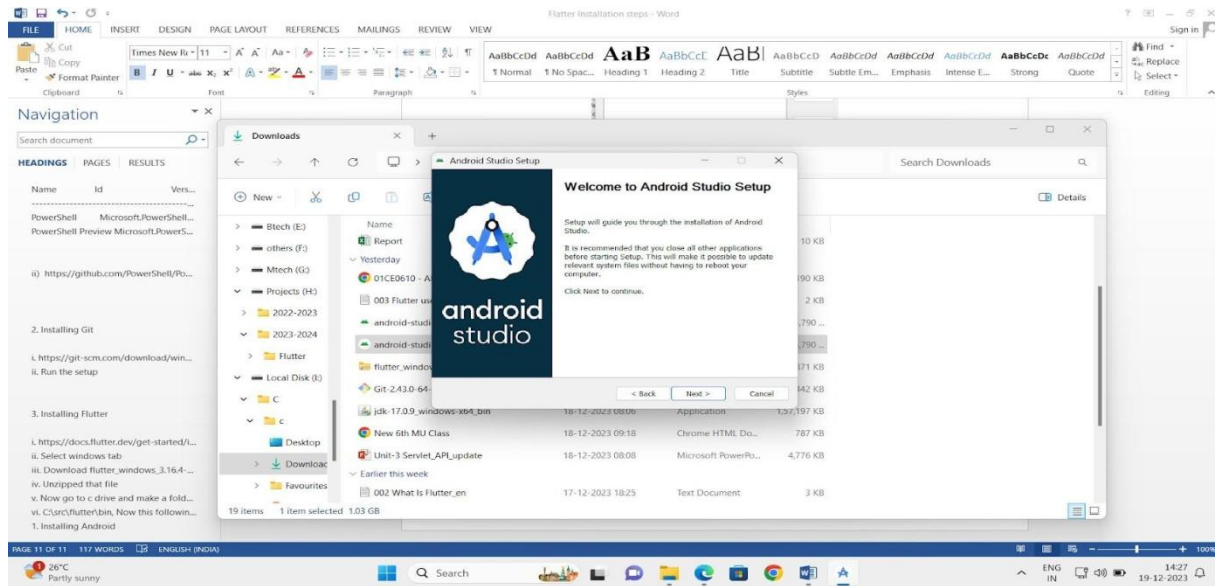
i. Download Android Studio:

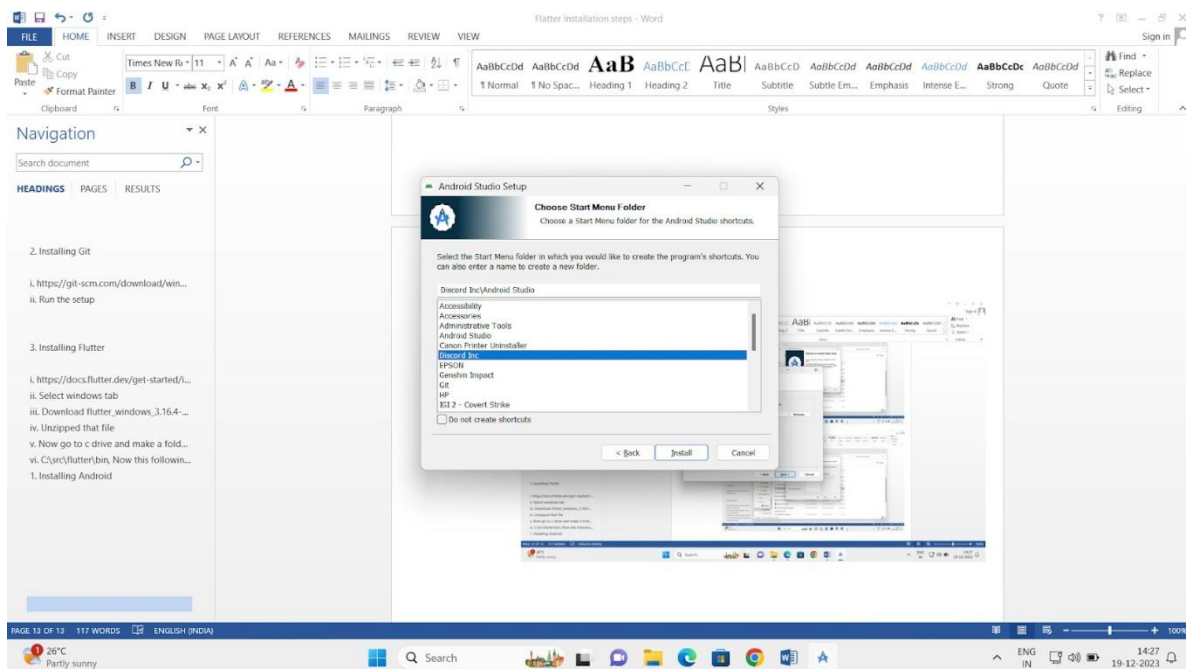
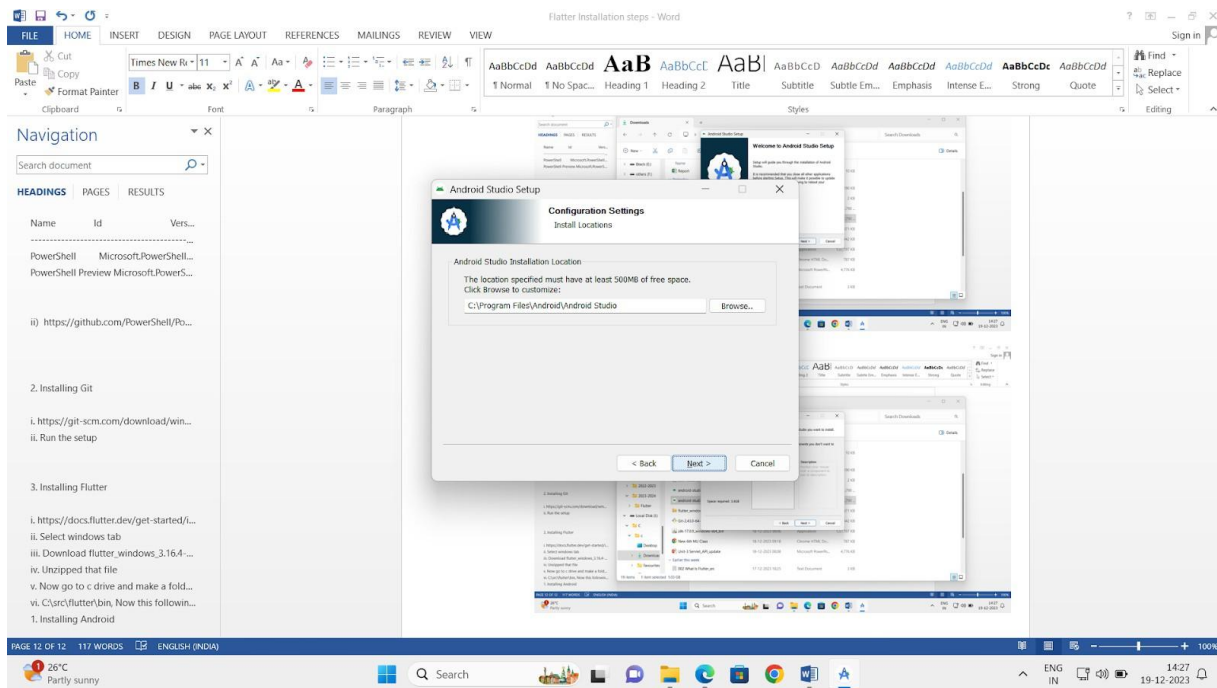
- Visit the Android Studio download page.
- Click on the "Download" button and download the Windows version.

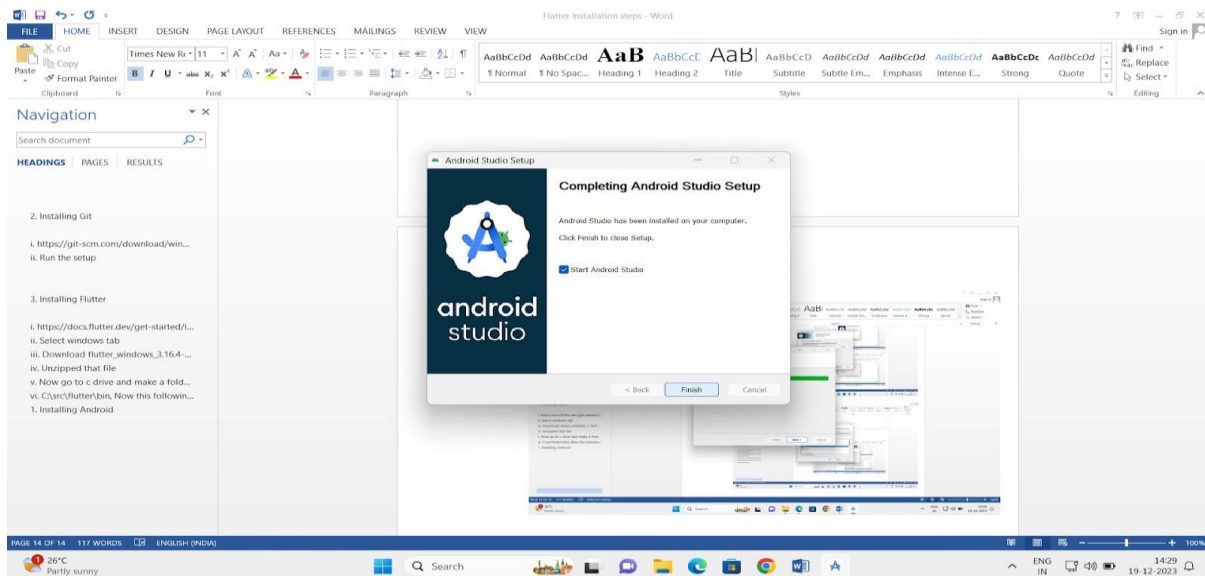
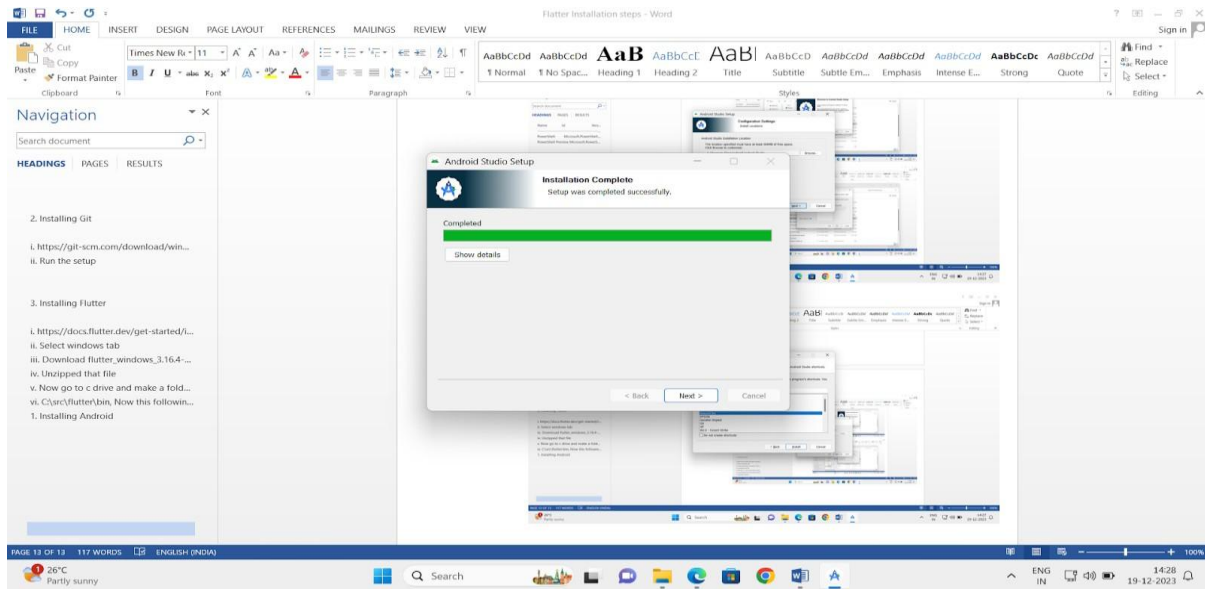
ii. Run the Installer:

- Once the download is complete, run the installer executable (.exe) file.

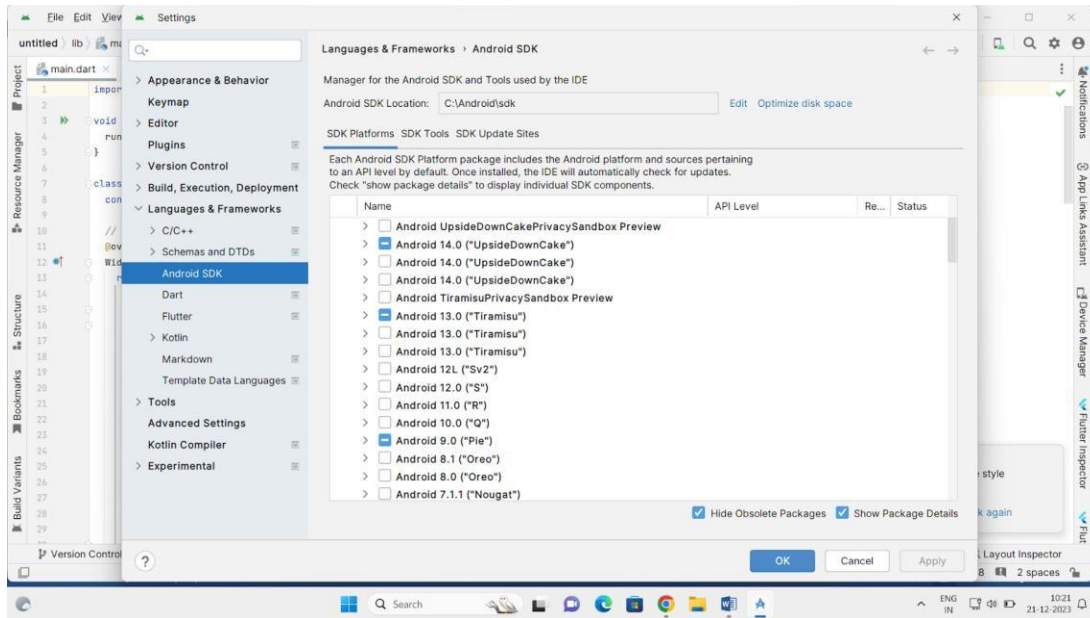
iii. Follow Installation Wizard:



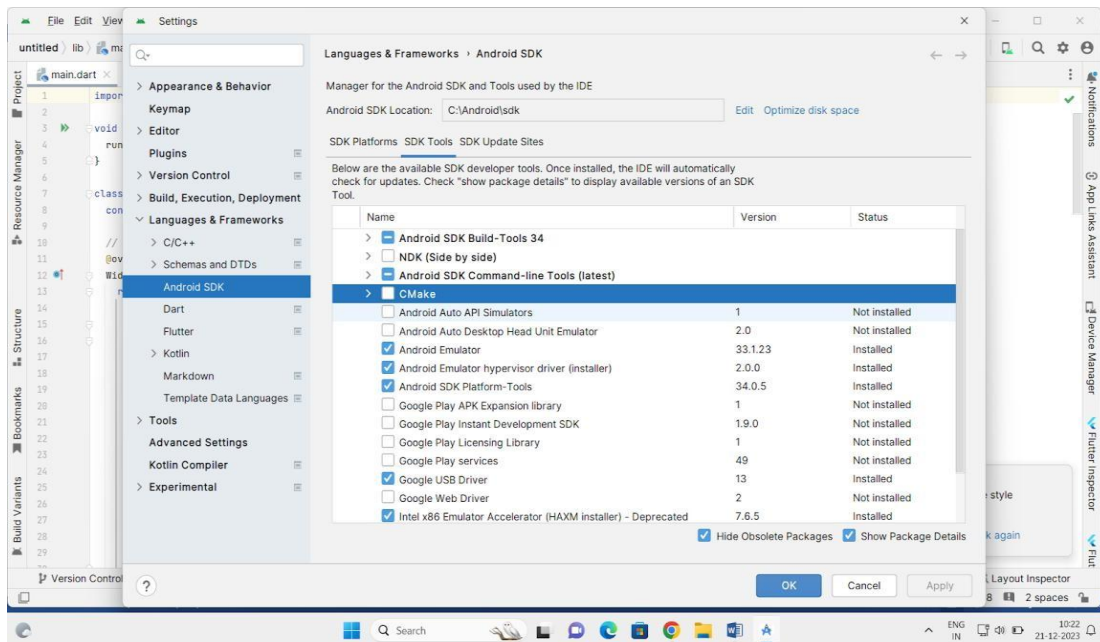




- **AndroidSDKPlatforms:**



- **AndroidSDKTools:**





- **Step3:Run Following Command for checking Flutter dependencies after installation of android.**

iv. AcceptAndroidLicenses

- Flutterdoctor--android-licenses to develop for Android, you need to accept the Android licenses.
- Run the following command: **flutterdoctor--android-licenses**

Experiment 02

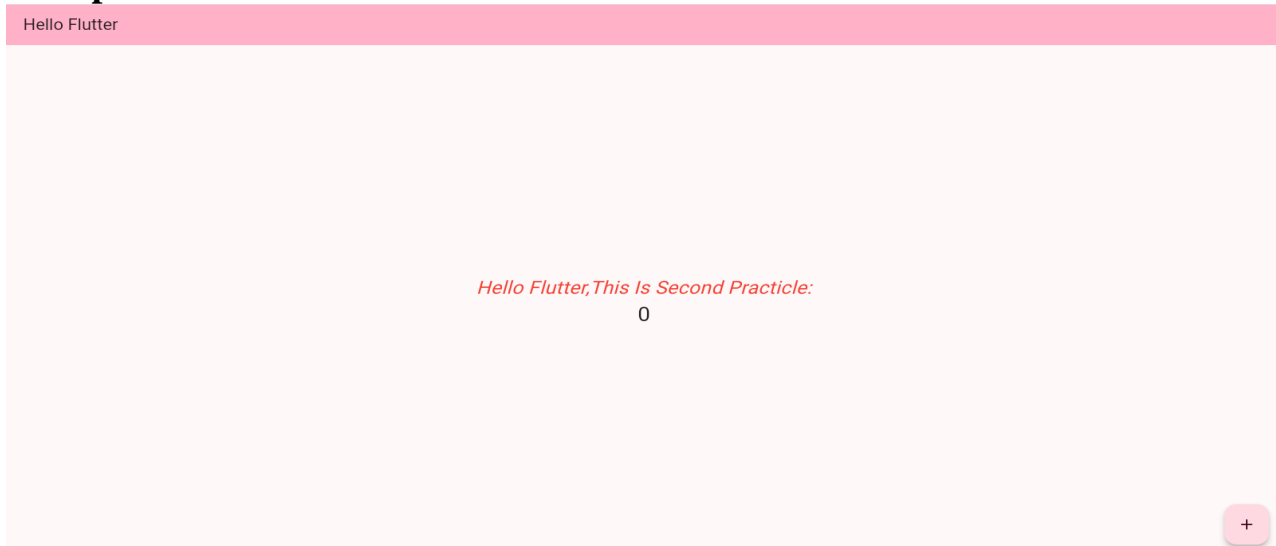
AIM:- Create a Hello Flutter application.

Source :- Mai.dart

Code :-

```
const Text(  
  
  'Hello Flutter:',  
  
  style: TextStyle(color: Colors.red, fontSize: 25, fontStyle: FontStyle.italic, ),  
  
),Text(  
  
  '$_counter',  
  
  style: Theme.of(context).textTheme.headlineMedium, ), ],), ),
```

Output:-



Experiment 03

AIM:- Create and application using Flutter Key Widgets.

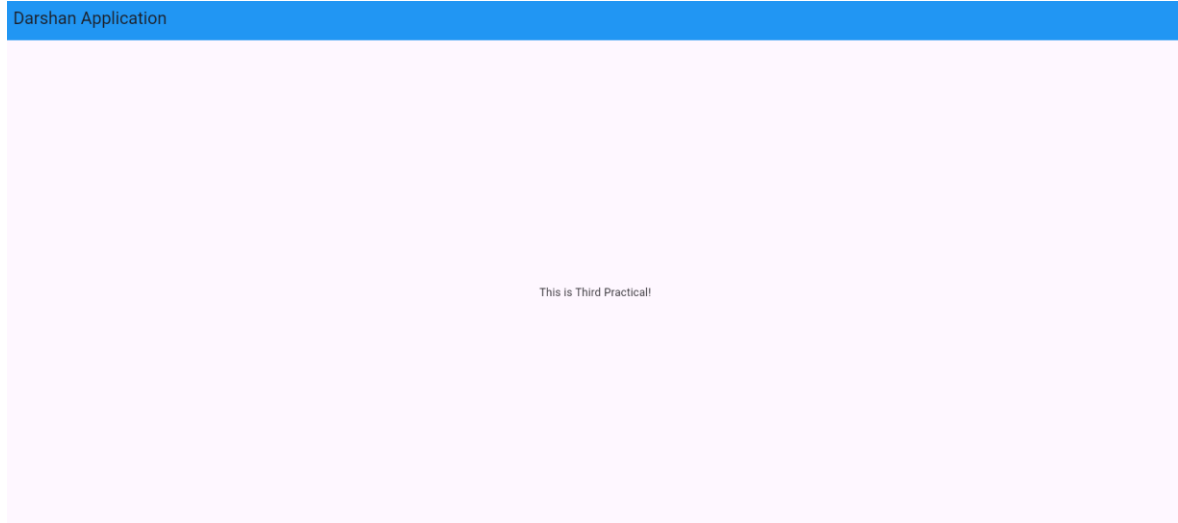
Source :- Mai.dart

Code :-

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

void main() {
  runApp(
    MaterialApp(
      debugShowCheckedModeBanner: false,
      home: Scaffold(
        appBar: AppBar(
          title: Text("Darshan Application"),
          toolbarTextStyle: TextStyle(color: Colors.green, fontWeight: FontWeight.bold, fontSize: 50),
          backgroundColor: Colors.blue, // Set the background color of the AppBar
        ),
        body: Center(
          child: Text(
            'This is Third Practical!',
            textAlign: TextAlign.center,
          ),
        ),
      ),
    ),
  );
}
```

Output:-



Practical 4: Create an application using Flutter Key Widgets

Main.dart

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

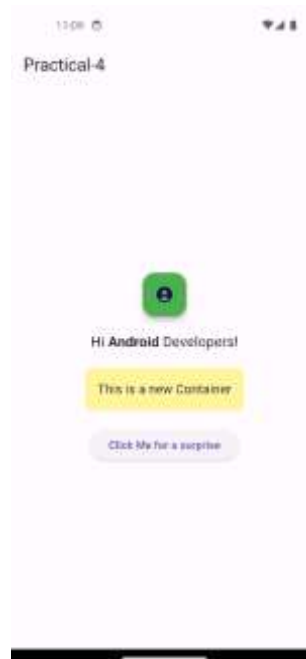
void main() {
  runApp(const MyApp());
}

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

  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      debugShowCheckedModeBanner: false,
      home: Scaffold(
        appBar: AppBar(
          title: const Text('Practical-4'),
        ),
        body: Center(
          child: Padding(
            padding: const EdgeInsets.all(16),
            child: Column(
              mainAxisAlignment: MainAxisAlignment.center,
              crossAxisAlignment: CrossAxisAlignment.center,
              children: [
                FloatingActionButton(
                  onPressed: () {
                    print('Floating Button Pressed');
                  },
                  child: const Icon(Icons.account_circle),
                  backgroundColor: Colors.green,
                ),
                const SizedBox(height: 20),
                Text.rich(
                  TextSpan(
                    text: 'Hi ',
                    style: DefaultTextStyle.of(context).style,
                    children: const <TextSpan>[
                      TextSpan(
                        text: 'Android',
                        style: TextStyle(fontWeight: FontWeight.bold),
                      ),
                      TextSpan(text: ' Developers!'),
                    ],
                  ),
                  style: const TextStyle(fontSize: 18),
                ),
                const SizedBox(height: 20),
                Container(
```

```
padding: const EdgeInsets.all(16),
decoration: BoxDecoration(
  color: Colors.yellow[200],
  borderRadius: BorderRadius.circular(8),
),
child: const Text(
  'This is a new Container',
  style: TextStyle(fontSize: 16),
),
),
const SizedBox(height: 20),
ElevatedButton(
  onPressed: () {
    print('Elevated Button Pressed');
  },
  style: ElevatedButton.styleFrom(),
  child: const Text(
    'Click Me for a surprise',
  ),
),
),
],
),
),
),
),
),
);
}
```

Output:



Practical 5: Create and application with Flutter UI Components.

Main.dart

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

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

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

class MyHomePage extends StatefulWidget {
  @override
  _MyHomePageState createState() => _MyHomePageState();
}

class _MyHomePageState extends State<MyHomePage> {
  TextEditingController _textEditingController = TextEditingController();
  String _displayText = "";

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
        title: Text('Flutter UI Components'),
      ),
      body: Padding(
        padding: const EdgeInsets.all(16.0),
        child: Column(
          mainAxisAlignment: MainAxisAlignment.center,
          children: [
            TextField(
              controller: _textEditingController,
              decoration: InputDecoration(
                hintText: 'Enter text...',
              ),
            ),
            SizedBox(height: 16),
            ElevatedButton(
```

```

    onPressed: () {
      setState(() {
        _displayText = _textEditingController.text;
      });
    },
    child: Text('Submit'),
  ),
  SizedBox(height: 16),
  Text(
    'Display Area:',
    style: TextStyle(fontSize: 18, fontWeight: FontWeight.bold),
  ),
  SizedBox(height: 8),
  Text(
    _displayText,
    style: TextStyle(fontSize: 16),
  ),
],
),
),
);
}
}

```

Login_page.dart

```

import 'package:flutter/material.dart';
class LoginPageDesign extends StatelessWidget {
  const LoginPageDesign({super.key});

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
        title: const Text("LAB 5"),
        backgroundColor: const Color.fromARGB(239, 183, 154, 58),
        foregroundColor: Colors.white,
        automaticallyImplyLeading: true,
      ),
      body: Container(
        child: SingleChildScrollView(
          child: Column(
            children: [
              const SizedBox(height: 100),
              Image.asset('images/phoenix.png', fit: BoxFit.contain, width: 200,height: 200,),
              const SizedBox(height: 15),
              const Text("Login to your account"),
              const SizedBox(height: 20),
              Container(
                margin: const EdgeInsets.symmetric(horizontal: 20),
                padding: const EdgeInsets.symmetric(horizontal: 10),
                decoration: BoxDecoration(

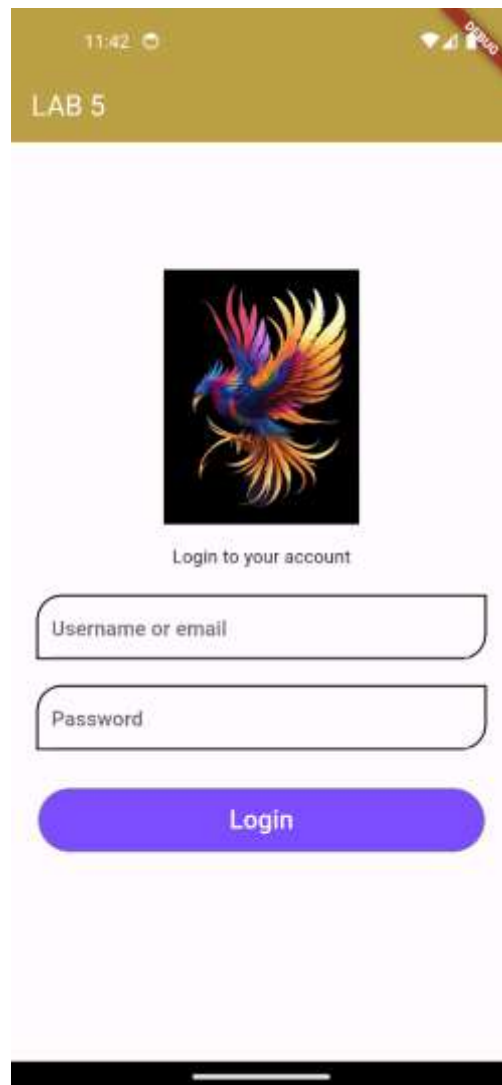
```

```
border: Border.all(color: Colors.black, width: 1.5),
borderRadius: const BorderRadius.only(
  topLeft: Radius.circular(20),
  bottomRight: Radius.circular(20)),
),
child: const TextField(
  //keyboardType: TextInputType.number,
  decoration: InputDecoration(
    hintText: "Username or email",
  ),
  spellCheckConfiguration: SpellCheckConfiguration(
    misspelledSelectionColor: Colors.red),
),
),
const SizedBox(height: 20),
Container(
  margin: const EdgeInsets.symmetric(horizontal: 20),
  padding: const EdgeInsets.symmetric(horizontal: 10),
  decoration: BoxDecoration(
    border: Border.all(color: Colors.black, width: 1.5),
    borderRadius: const BorderRadius.only(
      topLeft: Radius.circular(20),
      bottomRight: Radius.circular(20)),
  ),
  child: const TextField(
    obscureText: true,
    decoration: InputDecoration(
      hintText: "Password",
    ),
  ),
),
const SizedBox(height: 30),
const Container(
  height: 50,
  width: 350,
  child: OutlinedButton(
    onPressed: () {},
    style: ButtonStyle(
      foregroundColor: MaterialStateProperty.all(Colors.white),
      backgroundColor:
        MaterialStateProperty.all(Colors.deepPurpleAccent),
    ),
    child: const Text(
      "Login",
      style: TextStyle(
        fontSize: 20,
      ),
    ),
  ),
),
],
),
),
```



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

Output:



Practical 6: Create and application with Flutter UI Components.

Main.dart

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

import 'loginpagevalidation.dart';

void main(){
  runApp(
    MaterialApp(
      debugShowCheckedModeBanner: false,
      home: loginpagevalidation(),
    ));
}
```

Loginpagevalidation.dart

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

class loginpagevalidation extends StatefulWidget {
  const loginpagevalidation({super.key});
  @override
  State<loginpagevalidation> createState() => _loginpagevalidationState();
}

class _loginpagevalidationState extends State<loginpagevalidation> {
  final GlobalKey<FormState> _formKey = GlobalKey<FormState>();
  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
        title: const Text("LAB 6"),
        backgroundColor: Color.fromARGB(255, 212, 0, 0),
        foregroundColor: Colors.black,
        automaticallyImplyLeading: true,
      ),
      body: Container(
        alignment: Alignment.center,
        decoration: const BoxDecoration(
          gradient: LinearGradient(
            colors: [
              Color.fromARGB(255, 34, 139, 192),
              Color.fromARGB(255, 8, 60, 155)
            ],
          ),
          begin: Alignment.topLeft,
          end: Alignment.topRight,
        ),
        child: SingleChildScrollView(
          child: Form(
```

```
key: _formKey,
child: Column(
  children: [
    ClipRRect(
      borderRadius: BorderRadius.circular(20),
      child: Image.asset(
        "images/phoenix.jpg",
        fit: BoxFit.contain,
        width: 200,
        height: 200,
      ),
    ),
    const SizedBox(height: 15),
    const Text("Login to your account", style: TextStyle(color: Colors.white)),
    const SizedBox(height: 20),
    Container(
      margin: const EdgeInsets.symmetric(horizontal: 20),
      padding: const EdgeInsets.symmetric(horizontal: 10),
      decoration: BoxDecoration(
        border: Border.all(color: Colors.black, width: 1.5),
        borderRadius: const BorderRadius.only(
          topLeft: Radius.circular(20),
          bottomRight: Radius.circular(20),
        ),
      ),
    ),
    child: TextFormField(
      keyboardType: TextInputType.number,
      decoration: const InputDecoration(
        hintText: "Username or email",
      ),
      validator: (value) {
        if (value == null || value.isEmpty) {
          return 'Please enter your username and email';
        }
        return null;
      },
      spellCheckConfiguration: const SpellCheckConfiguration(
        misspelledSelectionColor: Colors.red,
      ),
    ),
    const SizedBox(height: 20),
    Container(
      margin: const EdgeInsets.symmetric(horizontal: 20),
      padding: const EdgeInsets.symmetric(horizontal: 10),
      decoration: BoxDecoration(
        border: Border.all(color: Colors.black, width: 1.5),
        borderRadius: const BorderRadius.only(
          topLeft: Radius.circular(20),
          bottomRight: Radius.circular(20)),
      ),
    ),
    child: TextFormField(
      obscureText: true,
      decoration: const InputDecoration(
        hintText: "Password",
```

```

    ),
    validator: (value) {
      if (value == null || value.isEmpty) {
        return 'Please enter your Password';
      }
      return null;
    },
  ),
),
const SizedBox(height: 30),
Container(
  height: 50,
  width: 350,
  child: OutlinedButton(
    onPressed: () {
      if (_formKey.currentState?.validate() ?? false) {
        Navigator.push(
          context,
          MaterialPageRoute(
            builder: (context) => OnTapExample(),
          ));
      }
    },
    style: ButtonStyle(
      foregroundColor: MaterialStateProperty.all(Colors.white),
      backgroundColor:
        MaterialStateProperty.all(Colors.deepPurpleAccent),
    ),
    child: const Text(
      "Login",
      style: TextStyle(
        fontSize: 20,
      ),
    ),
  ),
),
),
),
],
),
),
),
),
);
}
}

```

Custom_gesture.dart

```

import 'package:flutter/material.dart';

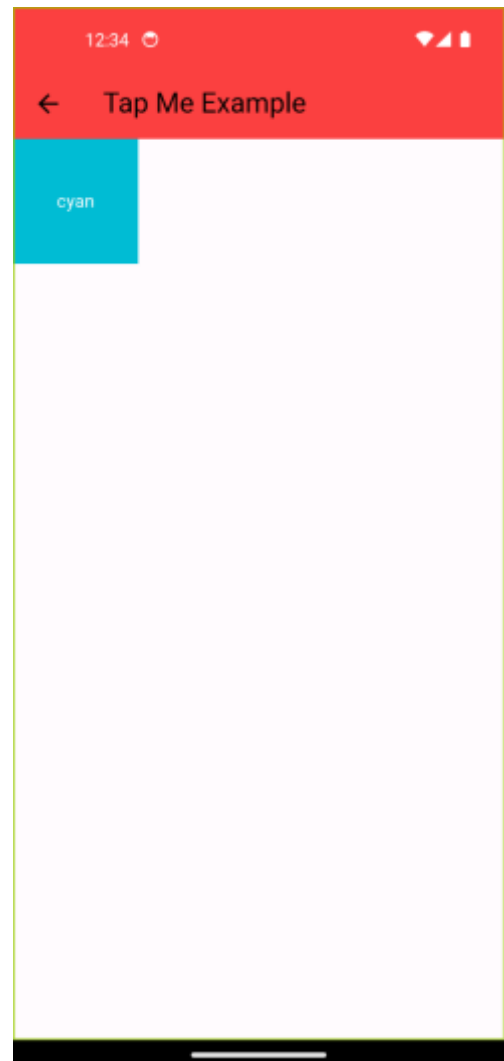
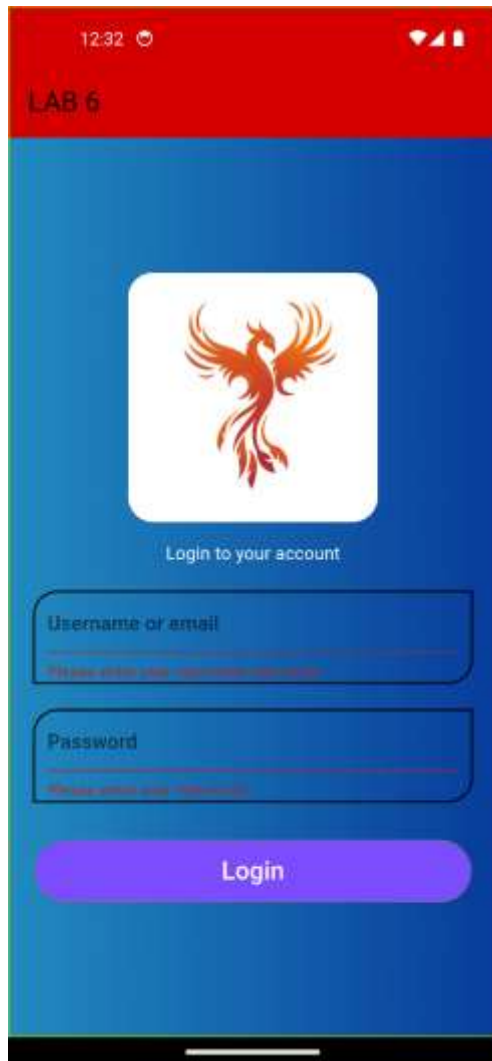
class OnTapExample extends StatefulWidget {
  const OnTapExample({super.key});
  @override
  State<OnTapExample> createState() => _OnTapExampleState();
}

```

```
}

class _OnTapExampleState extends State<OnTapExample> {
  Color color1 = Colors.cyan;
  String text1 = "cyan";
  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
        backgroundColor: Colors.purpleAccent,
        foregroundColor: Colors.white,
        title: const Text("Tap Me Example"),
      ),
      body: GestureDetector(
        onTap: () {
          setState(
            () {
              if (color1 == Colors.cyan) {
                color1 = Colors.red;
                text1 = "red";
              } else {
                color1 = Colors.cyan;
                text1 = "cyan";
              }
            }
          );
        },
      ),
      child: Container(
        height: 100,
        width: 100,
        color: color1,
        child: Center(
          child: Text(
            text1,
            style: const TextStyle(
              color: Colors.white,
            ),
          ),
        ),
      ),
    );
  }
}
```

Output:



Practical 7: Create and application with Navigation in Flutter.

Main.dart

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

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

  @override
  State<NavigationExample> createState() => _NavigationExampleState();
}

class _NavigationExampleState extends State<NavigationExample> {
  int _currentIndex = 0;

  final List<Widget> _pages = [
    FirstPage(),
    SecondPage(),
    ThirdPage(),
  ];

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
        title: const Text("Practical-4"),
      ),
      body: _pages[_currentIndex],
      bottomNavigationBar: BottomNavigationBar(
        currentIndex: _currentIndex,
        onTap: (index) {
          setState(() {
            _currentIndex = index;
          });
        },
        items: const [
          BottomNavigationBarItem(icon: Icon(Icons.home), label: 'Home'),
          BottomNavigationBarItem(icon: Icon(Icons.work), label: 'work'),
          BottomNavigationBarItem(
            icon: Icon(Icons.account_circle_rounded), label: 'Account'),
        ],
      ),
    );
  }

  class FirstPage extends StatelessWidget {
    @override
    Widget build(BuildContext context) {
      return const Center(
        child: Text(
```

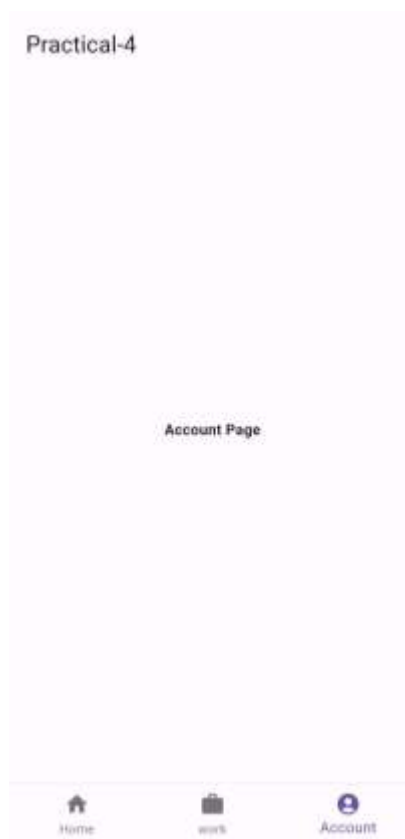


```
"Home Page",
style: TextStyle(
  fontWeight: FontWeight.bold,
),
));
}
}

class SecondPage extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return const Center(
      child: Text(
        "Work Page",
        style: TextStyle(
          fontWeight: FontWeight.bold,
        ),
      ),
    );
  }
}

class ThirdPage extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return const Center(
      child: Text(
        "Account Page",
        style: TextStyle(
          fontWeight: FontWeight.bold,
        ),
      ),
    );
  }
}
```

Output:



Practical 8: Create an application with list view in Flutter.

Main.dart

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

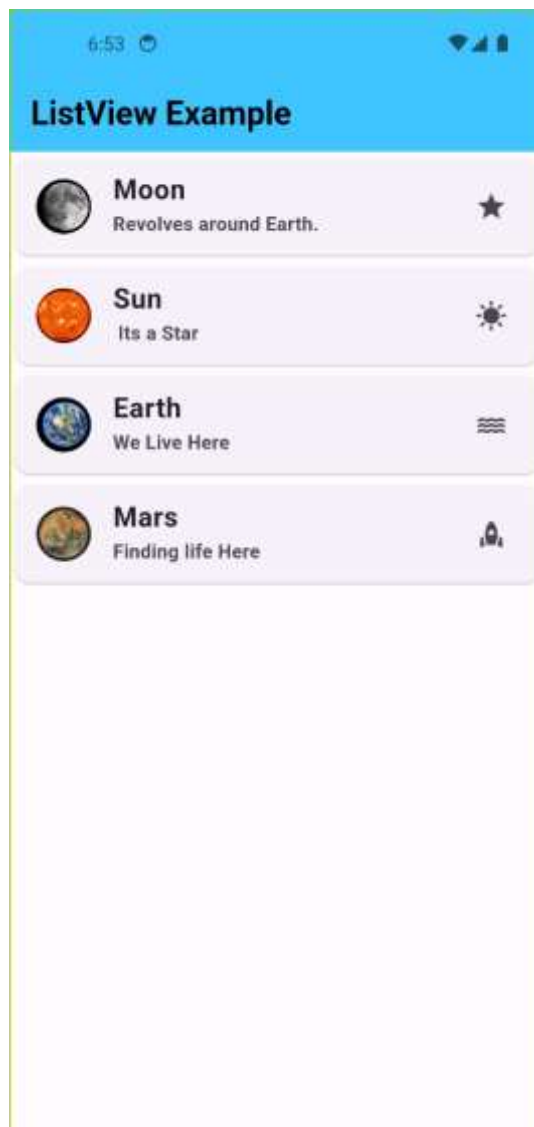
void main() {
  runApp(const MyApp());
}

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

  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      debugShowCheckedModeBanner: false,
      home: Scaffold(
        appBar: AppBar(
          title: const Text(
            'ListView Example',
            style: TextStyle(fontSize: 24, fontWeight: FontWeight.bold,)),
        ),
        foregroundColor: Colors.black,
        backgroundColor: Colors.lightBlueAccent,
      ),
      body: ListView(
        children: const <Widget>[
          Card(
            child: ListTile(
              title: Text("Moon", style: TextStyle(fontSize: 20, fontWeight: FontWeight.bold,)),
              subtitle: Text("Revolves around Earth.", style: TextStyle(fontWeight: FontWeight.bold,)),
              leading: CircleAvatar(
                backgroundImage: AssetImage('images/moon.jpg')),
              trailing: Icon(Icons.star))),
          Card(
            child: ListTile(
              title: Text("Sun", style: TextStyle(fontSize: 20, fontWeight: FontWeight.bold,)),
              subtitle: Text("Its a Star ", style: TextStyle(fontWeight: FontWeight.bold,)),
              leading: CircleAvatar(
                backgroundImage: AssetImage('images/sun.jpg')),
              trailing: Icon(Icons.sunny))),
          Card(
            child: ListTile(
              title: Text("Earth", style: TextStyle(fontSize: 20, fontWeight: FontWeight.bold,)),
              subtitle: Text("We Live Here", style: TextStyle(fontWeight: FontWeight.bold,)),
              leading: CircleAvatar(
                backgroundImage: AssetImage('images/earth.jpg')),
              trailing: Icon(Icons.water))),
          Card(
            child: ListTile(
```

```
title: Text("Mars", style: TextStyle(fontSize: 20, fontWeight: FontWeight.bold,)),  
subtitle: Text("Finding life Here", style: TextStyle(fontWeight: FontWeight.bold,)),  
leading: CircleAvatar(  
  backgroundImage: AssetImage('images/mars.jpg')),  
trailing: Icon(Icons.rocket))),  
// Add more ListTiles as needed  
],  
,  
,  
);  
}  
}
```

Output:



Practical 9: Create an application with Grid view in Flutter.

Main.dart

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

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

class MyApp extends StatelessWidget {
  final List<String> items = List.generate(8, (index) => 'Grid ${index + 1}');

  // Define a list of colors
  final List<Color> colors = [
    Colors.red,
    Colors.blue,
    Colors.green,
    Colors.orange,
    Colors.purple,
    Colors.teal,
    Colors.amber,
    Colors.indigo,
  ];

  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      debugShowCheckedModeBanner: false,
      home: Scaffold(
        appBar: AppBar(
          title: const Text('GridView', style: TextStyle(color: Colors.white, fontWeight: FontWeight.bold)),
          backgroundColor: Colors.black,
        ),
        body: GridView.builder(
          gridDelegate: const SliverGridDelegateWithFixedCrossAxisCount(
            crossAxisCount: 2,
            crossAxisSpacing: 15.0,
            mainAxisSpacing: 15.0,
          ),
          padding: const EdgeInsets.all(10),
          itemCount: items.length,
          itemBuilder: (BuildContext context, int index) {
            return Card(
              elevation: 6,
              color: colors[index % colors.length],
              child: Center(
                child: Text(
                  items[index],
                  style: const TextStyle(fontSize: 20.0, color: Colors.white), // Text color
                ),
              ),
            );
          },
        ),
      ),
    );
  }
}
```

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

Output:



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

Main.dart

```
import 'package:prac10/model/User.dart';
import 'package:prac10/screens/edit_User.dart';
import 'package:prac10/screens/add_User.dart';
import 'package:prac10/screens/view_User.dart';
import 'package:prac10/services/user_Service.dart';
import 'package:flutter/material.dart';

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

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

  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      title: 'Flutter Demo',
      debugShowCheckedModeBanner: false,
      theme: ThemeData(
        primarySwatch: Colors.teal,
      ),
      home: const MyHomePage(),
    );
  }
}

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

  @override
  State<MyHomePage> createState() => _MyHomePageState();
}

class _MyHomePageState extends State<MyHomePage> {
  late List<User> _userList = <User>[];
  final _userService = UserService();

  getAllUserDetails() async {
    var users = await _userService.readAllUsers();
    _userList = <User>[];
    users.forEach((user) {
      setState(() {
        var userModel = User();
        userModel.id = user['id'];
        userModel.name = user['name'];
        userModel.contact = user['contact'];
        userModel.description = user['description'];
      });
    });
  }
}
```



```
_userList.add(userModel);
});
});
}

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

_showSuccessSnackBar(String message) {
  ScaffoldMessenger.of(context).showSnackBar(
    SnackBar(
      content: Text(message),
    ),
  );
}

_deleteFormDialog(BuildContext context, userId) {
  return showDialog(
    context: context,
    builder: (param) {
      return AlertDialog(
        title: const Text(
          'Are You Sure to Delete',
          style: TextStyle(color: Colors.teal, fontSize: 20),
        ),
        actions: [
          TextButton(
            style: TextButton.styleFrom(
              foregroundColor: Colors.white, backgroundColor: Colors.red),
            onPressed: () async{
              var result=await _userService.deleteUser(userId);
              if (result != null) {
                Navigator.pop(context);
                getAllUserDetails();
                _showSuccessSnackBar(
                  'User Detail Deleted Success');
              }
            },
            child: const Text('Delete')),
          TextButton(
            style: TextButton.styleFrom(
              foregroundColor: Colors.white, backgroundColor: Colors.teal),
            onPressed: () {
              Navigator.pop(context);
            },
            child: const Text('Close'))
        ],
      );
    }
  );
}
```

```
@override
Widget build(BuildContext context) {
  return Scaffold(
    appBar: AppBar(
      title: const Text("SQLite CRUD"),
    ),
    body: ListView.builder(
      itemCount: _userList.length,
      itemBuilder: (context, index) {
        return Card(
          child: ListTile(
            onTap: () {
              Navigator.push(
                context,
                MaterialPageRoute(
                  builder: (context) => ViewUser(
                    user: _userList[index],
                  ));
            },
            leading: const Icon(Icons.person),
            title: Text(_userList[index].name ?? ""),
            subtitle: Text(_userList[index].contact ?? ""),
            trailing: Row(
              mainAxisAlignment: MainAxisAlignment.min,
              children: [
                IconButton(
                  onPressed: () {
                    Navigator.push(
                      context,
                      MaterialPageRoute(
                        builder: (context) => EditUser(
                          user: _userList[index],
                        )),
                    ).then((data) {
                      if (data != null) {
                        getAllUserDetails();
                        _showSuccessSnackBar(
                          'User Detail Updated Success');
                      }
                    });
                  },
                ),
                icon: const Icon(
                  Icons.edit,
                  color: Colors.teal,
                ),
              ],
            ),
            IconButton(
              onPressed: () {
                _deleteFormDialog(context, _userList[index].id);
              },
              icon: const Icon(
                Icons.delete,
                color: Colors.red,
              ),
            )
          )
        );
      }
    )
  );
}
```

```

        ],
      ),
    ),
  );
}),
floatingActionButton: FloatingActionButton(
  onPressed: () {
    Navigator.push(context,
      MaterialPageRoute(builder: (context) => const AddUser()))
      .then((data) {
        if (data != null) {
          getAllUserDetails();
          _showSuccessSnackBar('User Detail Added Success');
        }
      });
  },
  child: const Icon(Icons.add),
),
);
}
}

```

database_connection.dart

```

import 'package:sqflite/sqflite.dart';
import 'package:path/path.dart';
import 'package:path_provider/path_provider.dart';

class DatabaseConnection {
  Future<Database> setDatabase() async {
    var directory = await getApplicationDocumentsDirectory();
    var path = join(directory.path, 'db_crud');
    var database =
      await openDatabase(path, version: 1, onCreate: _createDatabase);
    return database;
  }

  Future<void> _createDatabase(Database database, int version) async {
    String sql =
      "CREATE TABLE users (id INTEGER PRIMARY KEY,name TEXT,contact Text,description TEXT);";
    await database.execute(sql);
  }
}

```

repository.dart:

```

import 'package:prac10/db_helper/database_connection.dart';
import 'package:sqflite/sqflite.dart';

class Repository
{
  late DatabaseConnection _databaseConnection;
}

```

```
Repository(){
  _databaseConnection = DatabaseConnection();
}
static Database? _database;
Future<Database?> get database async {
  if (_database != null) {
    return _database;
  } else {
    _database = await _databaseConnection.setDatabase();
    return _database;
  }
}

insertData(table, data) async {
  var connection = await database;
  return await connection?.insert(table, data);
}
readData(table) async {
  var connection = await database;
  return await connection?.query(table);
}
readDataById(table, itemId) async {
  var connection = await database;
  return await connection?.query(table, where: 'id=?', whereArgs: [itemId]);
}
updateData(table, data) async {
  var connection = await database;
  return await connection
    ?.update(table, data, where: 'id=?', whereArgs: [data['id']]);
}
deleteDataById(table, itemId) async {
  var connection = await database;
  return await connection?.rawDelete("delete from $table where id=$itemId");
}
}
```

Model.dart

```
class User {
  int? id;
  String? name;
  String? contact;
  String? description;

  userMap() {
    var mapping = Map<String, dynamic>();
    mapping['id'] = id ?? null;
    mapping['name'] = name!;
    mapping['contact'] = contact!;
    mapping['description'] = description!;
    return mapping;
  }
}
```

Add_user.dart

```
import 'package:prac10/model/User.dart';
import 'package:prac10/services/user_Service.dart';
import 'package:flutter/material.dart';

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

  @override
  State<AddUser> createState() => _AddUserState();
}

class _AddUserState extends State<AddUser> {
  var _userNameController = TextEditingController();
  var _userContactController = TextEditingController();
  var _userDescriptionController = TextEditingController();
  bool _validateName = false;
  bool _validateContact = false;
  bool _validateDescription = false;
  var _userService=UserService();
  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
        title: const Text("SQLite CRUD"),
      ),
      body: SingleChildScrollView(
        child: Container(
          padding: const EdgeInsets.all(16.0),
          child: Column(
            crossAxisAlignment: CrossAxisAlignment.start,
            children: [
              const Text(
                'Add New User',
                style: TextStyle(
                  fontSize: 20,
                  color: Colors.teal,
                  fontWeight: FontWeight.w500),
              ),
              const SizedBox(
                height: 20.0,
              ),
              TextField(
                controller: _userNameController,
                decoration: InputDecoration(
                  border: const OutlineInputBorder(),
                  hintText: 'Enter Name',
                  labelText: 'Name',
                  errorText:
                    _validateName ? 'Name Value Can\'t Be Empty' : null,
                ),
              ),
            ],
          ),
        ),
      ),
    );
  }
}
```

```
const SizedBox(
  height: 20.0,
),
TextField(
  controller: _userContactController,
  decoration: InputDecoration(
    border: const OutlineInputBorder(),
    hintText: 'Enter Contact',
    labelText: 'Contact',
    errorText: _validateContact
      ? 'Contact Value Can\'t Be Empty'
      : null,
  ),
),
const SizedBox(
  height: 20.0,
),
TextField(
  controller: _userDescriptionController,
  decoration: InputDecoration(
    border: const OutlineInputBorder(),
    hintText: 'Enter Description',
    labelText: 'Description',
    errorText: _validateDescription
      ? 'Description Value Can\'t Be Empty'
      : null,
  ),
),
const SizedBox(
  height: 20.0,
),
Row(
  children: [
    TextButton(
      style: TextButton.styleFrom(
        foregroundColor: Colors.white, backgroundColor: Colors.teal,
        textStyle: const TextStyle(fontSize: 15)),
      onPressed: () async {
        setState(() {
          _userNameController.text.isEmpty
            ? _validateName = true
            : _validateName = false;
          _userContactController.text.isEmpty
            ? _validateContact = true
            : _validateContact = false;
          _userDescriptionController.text.isEmpty
            ? _validateDescription = true
            : _validateDescription = false;
        });
        if (_validateName == false &&
            _validateContact == false &&
            _validateDescription == false) {
          // print("Good Data Can Save");
          var _user = User();
```

```

        _user.name = _userNameController.text;
        _user.contact = _userContactController.text;
        _user.description = _userDescriptionController.text;
        var result=await _userService.SaveUser(_user);
        Navigator.pop(context,result);
    }
},
child: const Text('Save Details')),
const SizedBox(
  width: 10.0,
),
IconButton(
  style: IconButton.styleFrom(
    foregroundColor: Colors.white, backgroundColor: Colors.red,
    textStyle: const TextStyle(fontSize: 15)),
  onPressed: () {
    _userNameController.text = "";
    _userContactController.text = "";
    _userDescriptionController.text = "";
  },
  child: const Text('Clear Details'))
),
)
],
),
),
),
);
}
}

```

Edit_user.dart

```

import 'package:prac10/model/User.dart';
import 'package:prac10/services/user_Service.dart';
import 'package:flutter/material.dart';
class EditUser extends StatefulWidget {
  final User user;
  const EditUser({Key? key,required this.user}) : super(key: key);

  @override
  State<EditUser> createState() => _EditUserState();
}

class _EditUserState extends State<EditUser> {
  var _userNameController = TextEditingController();
  var _userContactController = TextEditingController();
  var _userDescriptionController = TextEditingController();
  bool _validateName = false;
  bool _validateContact = false;
  bool _validateDescription = false;
  var _userService=UserService();

```



```
@override
void initState() {
  setState(() {
    _userNameController.text=widget.user.name??";
    _userContactController.text=widget.user.contact??";
    _userDescriptionController.text=widget.user.description??";
  });
  super.initState();
}
@override
Widget build(BuildContext context) {
  return Scaffold(
    appBar: AppBar(
      title: const Text("SQLite CRUD"),
    ),
    body: SingleChildScrollView(
      child: Container(
        padding: const EdgeInsets.all(16.0),
        child: Column(
          crossAxisAlignment: CrossAxisAlignment.start,
          children: [
            const Text(
              'Edit New User',
              style: TextStyle(
                fontSize: 20,
                color: Colors.teal,
                fontWeight: FontWeight.w500),
            ),
            const SizedBox(
              height: 20.0,
            ),
            TextField(
              controller: _userNameController,
              decoration: InputDecoration(
                border: const OutlineInputBorder(),
                hintText: 'Enter Name',
                labelText: 'Name',
                errorText:
                  _validateName ? 'Name Value Can\'t Be Empty' : null,
              ),
            ),
            const SizedBox(
              height: 20.0,
            ),
            TextField(
              controller: _userContactController,
              decoration: InputDecoration(
                border: const OutlineInputBorder(),
                hintText: 'Enter Contact',
                labelText: 'Contact',
                errorText: _validateContact
                  ? 'Contact Value Can\'t Be Empty'
                  : null,
              ),
            ),
          ],
        ),
      ),
    ),
  );
}
```

```
const SizedBox(
  height: 20.0,
),
TextField(
  controller: _userDescriptionController,
  decoration: InputDecoration(
    border: const OutlineInputBorder(),
    hintText: 'Enter Description',
    labelText: 'Description',
    errorText: _validateDescription
      ? 'Description Value Can\'t Be Empty'
      : null,
  ),
),
const SizedBox(
  height: 20.0,
),
Row(
  children: [
    TextButton(
      style: TextButton.styleFrom(
        foregroundColor: Colors.white, backgroundColor: Colors.teal,
        textStyle: const TextStyle(fontSize: 15)),
      onPressed: () async {
        setState(() {
          _userNameController.text.isEmpty
            ? _validateName = true
            : _validateName = false;
          _userContactController.text.isEmpty
            ? _validateContact = true
            : _validateContact = false;
          _userDescriptionController.text.isEmpty
            ? _validateDescription = true
            : _validateDescription = false;
        });
        if (_validateName == false &&
            _validateContact == false &&
            _validateDescription == false) {
          // print("Good Data Can Save");
          var _user = User();
          _user.id=widget.user.id;
          _user.name = _userNameController.text;
          _user.contact = _userContactController.text;
          _user.description = _userDescriptionController.text;
          var result=await _userService.UpdateUser(_user);
          Navigator.pop(context,result);
        }
      },
      child: const Text('Update Details')),
    const SizedBox(
      width: 10.0,
    ),
    TextButton(
```

```

        style: TextButton.styleFrom(
          foregroundColor: Colors.white, backgroundColor: Colors.red,
          textStyle: const TextStyle(fontSize: 15)),
        onPressed: () {
          _userNameController.text = "";
          _userContactController.text = "";
          _userDescriptionController.text = "";
        },
        child: const Text('Clear Details'))
      ],
    ),
  ],
),
),
),
);
}
}

```

View_user.dart

```

import 'package:prac10/model/User.dart';
import 'package:flutter/material.dart';

class ViewUser extends StatefulWidget {
  final User user;

  const ViewUser({Key? key, required this.user}) : super(key: key);

  @override
  State<ViewUser> createState() => _ViewUserState();
}

class _ViewUserState extends State<ViewUser> {
  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
        title: const Text("SQLite CRUD"),
      ),
      body: Container(
        padding: EdgeInsets.all(16.0),
        child: Column(
          crossAxisAlignment: CrossAxisAlignment.start,
          children: [
            const Text(
              "Full Details",
              style: TextStyle(
                fontWeight: FontWeight.w600,
                color: Colors.blueGrey,
                fontSize: 20),
            ),
            const SizedBox(

```

```
        height: 20,
      ),
      Row(
        children: [
          const Text('Name',
            style: TextStyle(
              color: Colors.teal,
              fontSize: 16,
              fontWeight: FontWeight.w600)),
          Padding(
            padding: const EdgeInsets.only(left: 30),
            child: Text(widget.user.name ?? "", style: TextStyle(fontSize: 16)),
          ),
        ],
      ),
      const SizedBox(
        height: 20,
      ),
      Row(
        children: [
          const Text('Contact',
            style: TextStyle(
              color: Colors.teal,
              fontSize: 16,
              fontWeight: FontWeight.w600)),
          Padding(
            padding: const EdgeInsets.only(left: 25),
            child: Text(widget.user.contact ?? "", style: TextStyle(fontSize: 16)),
          ),
        ],
      ),
      const SizedBox(
        height: 20,
      ),
      Column(
        crossAxisAlignment: CrossAxisAlignment.start,
        children: [
          const Text('Description',
            style: TextStyle(
              color: Colors.teal,
              fontSize: 16,
              fontWeight: FontWeight.w600)),
          const SizedBox(
            height: 20,
          ),
          Text(widget.user.description ?? "", style: const TextStyle(fontSize: 16)),
        ],
      ),
    ],
  ),
),
));
}
```

User_service.dart

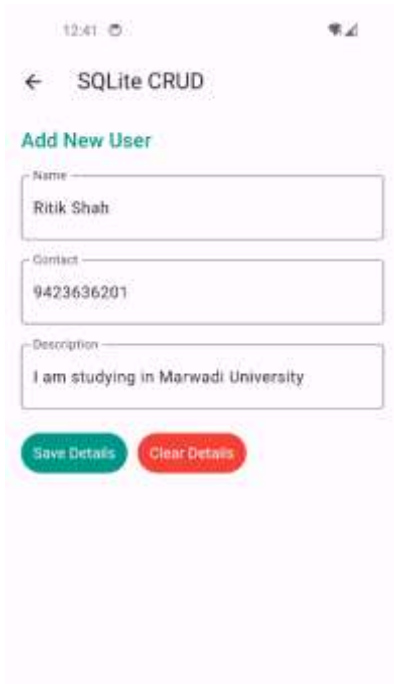
```
import 'package:prac10/db_helper/repository.dart';
import 'package:prac10/model/User.dart';

class UserService
{
  late Repository _repository;
  UserService(){
    _repository = Repository();
  }
  SaveUser(User user) async{
    return await _repository.insertData('users', user.userMap());
  }
  readAllUsers() async{
    return await _repository.readData('users');
  }
  UpdateUser(User user) async{
    return await _repository.updateData('users', user.userMap());
  }

  deleteUser(userId) async {
    return await _repository.deleteDataById('users', userId);
  }
}
```

Outputs:

Adding new User



12:41

← SQLite CRUD

Add New User

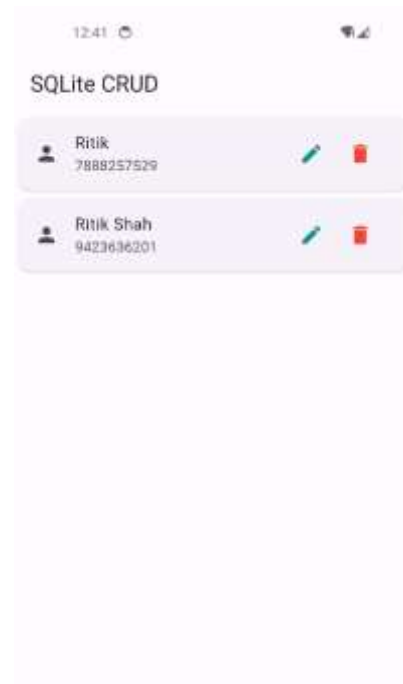
Name
Ritik Shah

Contact
9423636201

Description
I am studying in Marwadi University





Save Details Clear Details

View Data



12:41

SQLite CRUD

Ritik 7888257529	 
Ritik Shah 9423636201	 

Edit Data



12:42

← SQLite CRUD

Edit New User

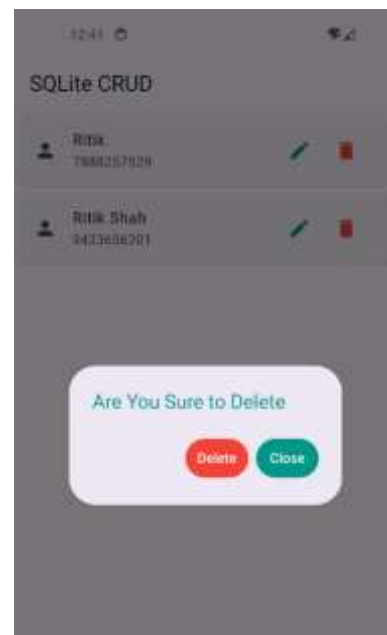
Name
Ritik Shah

Contact
9423636201

Description
I am studying in Marwadi University





Update Details Clear Details

Delete Data



12:41

SQLite CRUD

Ritik 7888257529	 
Ritik Shah 9423636201	 

Are You Sure to Delete

Delete Close

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(const MyApp());
}

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

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

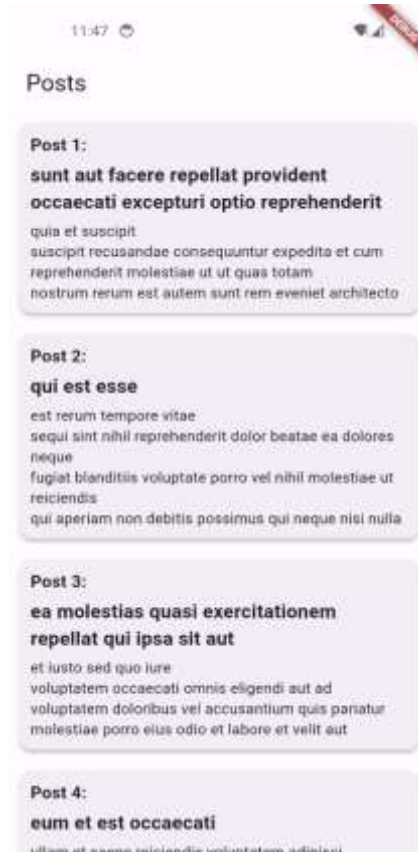
```



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

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

Output:



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

Main.dart

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

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

class MyApp extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      title: 'Flutter Demo',
      home: PostListScreen(),
    );
  }
}

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

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

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

class PostListScreen extends StatefulWidget {
  @override
  _PostListScreenState createState() => _PostListScreenState();
}

class _PostListScreenState extends State<PostListScreen> {
  late Future<List<Post>> futurePosts;

  @override
  void initState() {
    super.initState();
    futurePosts = fetchPosts();
  }
}
```

```
Future<List<Post>> fetchPosts() async {
  final response =
    await http.get(Uri.parse('https://jsonplaceholder.typicode.com/posts'));

  if (response.statusCode == 200) {
    List<dynamic> data = json.decode(response.body);
    return data.map((json) => Post.fromJson(json)).toList();
  } else {
    throw Exception('Failed to load posts');
  }
}

@override
Widget build(BuildContext context) {
  return Scaffold(
    appBar: AppBar(
      title: Text('Posts'),
      backgroundColor: Colors.amberAccent,
    ),
    body: FutureBuilder<List<Post>>(
      future: futurePosts,
      builder: (context, snapshot) {
        if (snapshot.connectionState == ConnectionState.waiting) {
          return Center(child: CircularProgressIndicator());
        } else if (snapshot.hasError) {
          return Center(child: Text('Error: ${snapshot.error}'));
        } else {
          List<Post> posts = snapshot.data ?? [];
          return ListView.builder(
            itemCount: posts.length,
            itemBuilder: (context, index) {
              return Card(
                margin: EdgeInsets.all(8.0),
                elevation: 4.0,
                child: ListTile(
                  title: Text(posts[index].title, style: TextStyle(fontWeight: FontWeight.bold)),
                  subtitle: Text(posts[index].body),
                ),
              );
            },
          );
        }
      },
    ),
  );
}
```

Output:



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

Main.dart

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

void main() async {
  WidgetsFlutterBinding.ensureInitialized();
  final cameras = await availableCameras();
  final firstCamera = cameras.first;
  runApp(CameraApp(camera: firstCamera));
}

class CameraApp extends StatelessWidget {
  final CameraDescription camera;

  const CameraApp({
    Key? key,
    required this.camera,
  }) : super(key: key);

  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      theme: ThemeData.dark(),
      home: CameraScreen(camera: camera),
    );
  }
}

class CameraScreen extends StatefulWidget {
  final CameraDescription camera;

  const CameraScreen({
    Key? key,
    required this.camera,
  }) : super(key: key);

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

class _CameraScreenState extends State<CameraScreen> {
  late CameraController _controller;
  late Future<void> _initializeControllerFuture;
  double _brightnessLevel = 0;
  Offset _focusPoint = Offset(0.5, 0.5);
```

```
@override
void initState() {
  super.initState();
  _controller = CameraController(
    widget.camera,
    ResolutionPreset.medium,
  );
  _initializeControllerFuture = _controller.initialize();
}

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

@override
Widget build(BuildContext context) {
  return Scaffold(
    appBar: AppBar(title: const Text('Camera Example')),
    body: Stack(
      children: [
        FutureBuilder<void>(
          future: _initializeControllerFuture,
          builder: (context, snapshot) {
            if (snapshot.connectionState == ConnectionState.done) {
              return CameraPreview(_controller);
            } else {
              return const Center(child: CircularProgressIndicator());
            }
          },
        ),
        Positioned(
          left: 0,
          right: 0,
          bottom: 20,
          child: Row(
            mainAxisAlignment: MainAxisAlignment.spaceEvenly,
            children: [
              IconButton(
                icon: Icon(Icons.camera),
                onPressed: () async {
                  try {
                    await _initializeControllerFuture;
                    final image = await _controller.takePicture();
                    _showSaveDialog(context, image.path);
                  } catch (e) {
                    print(e);
                  }
                },
              ),
              IconButton(
                icon: Icon(Icons.center_focus_strong),
```

```
        onPressed: () async {
          try {
            await _controller.setFocusPoint(_focusPoint);
          } catch (e) {
            print(e);
          }
        },
      ),
    Slider(
      min: -1,
      max: 1,
      value: _brightnessLevel,
      onChanged: (value) {
        setState(() {
          _brightnessLevel = value;
        });
        _adjustBrightness();
      },
    ),
  ],
),
],
),
);
}

void _showSaveDialog(BuildContext context, String imagePath) {
  showDialog(
    context: context,
    builder: (context) => AlertDialog(
      title: Text('Save Photo?'),
      content: Text('Do you want to save the photo to your device?'),
      actions: [
        TextButton(
          onPressed: () {
            Navigator.pop(context);
          },
          child: Text('No'),
        ),
        TextButton(
          onPressed: () {
            Navigator.pop(context);
            _savePhoto(imagePath);
          },
          child: Text('Yes'),
        ),
      ],
    ),
  );
}
```

```
void _savePhoto(String imagePath) {
```



```
// Implement photo saving logic here, for example using the 'path_provider' package
// For simplicity, this example just prints the path.
print('Photo saved at: $imagePath');
}

void _adjustBrightness() {
  // Adjust brightness by setting exposure compensation
  _controller.setExposureMode(ExposureMode.auto);
  _controller.setExposureOffset(_brightnessLevel);
}
}
```

Output:

