



Practical - 1

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

Solution:

Step 1: Installing a Flutter.

I. System Requirements:

- Assure that your system meets the minimum requirements. Flutter supports macOS, Linux, and Windows.
- On macOS, you need Xcode with the command-line tools installed.
- On Linux, you need to have git, lib32stdc++6, and other dependencies installed.

II. Download Flutter

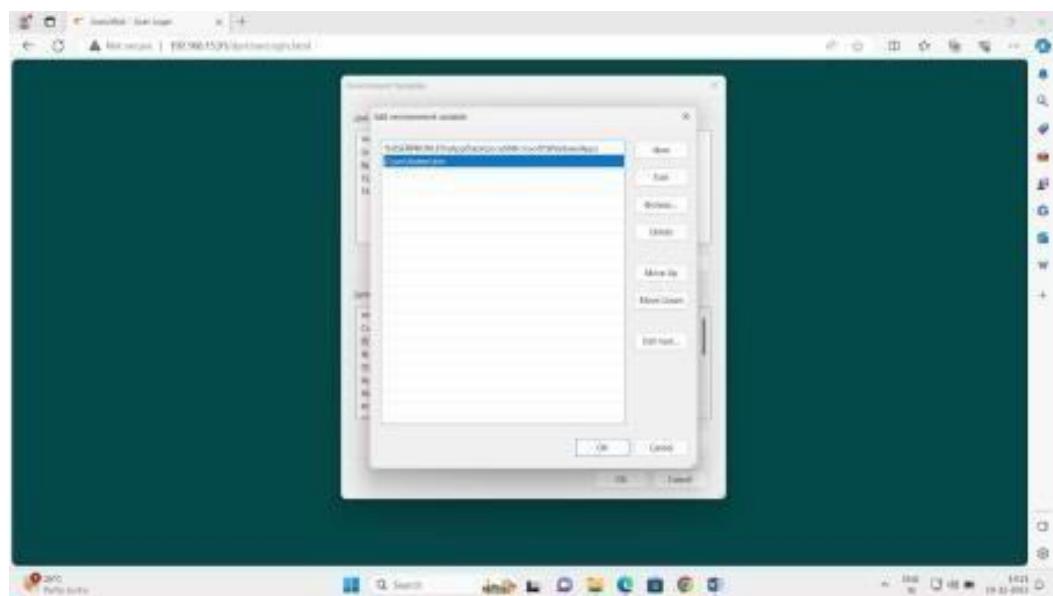
- Visit Flutter Website for Installation of Flutter -> <https://docs.flutter.dev/get-started/install>

III. Extract Flutter:

- If you downloaded the ZIP file, extract it to a location on your machine. (**C:\src\flutter**).

IV. Setup Environment Variables:

- Add the **C:\src\flutter\bin** directory to your system's PATH variable





V. Run Flutter Doctor:

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

VI. Install Flutter Dependencies:

- Follow the instruction provided by the flutter doctor to install any missing dependencies. This may include things like Android studio, Xcode, command-line-tools, etc.

Step 2: Installing Android Studio.

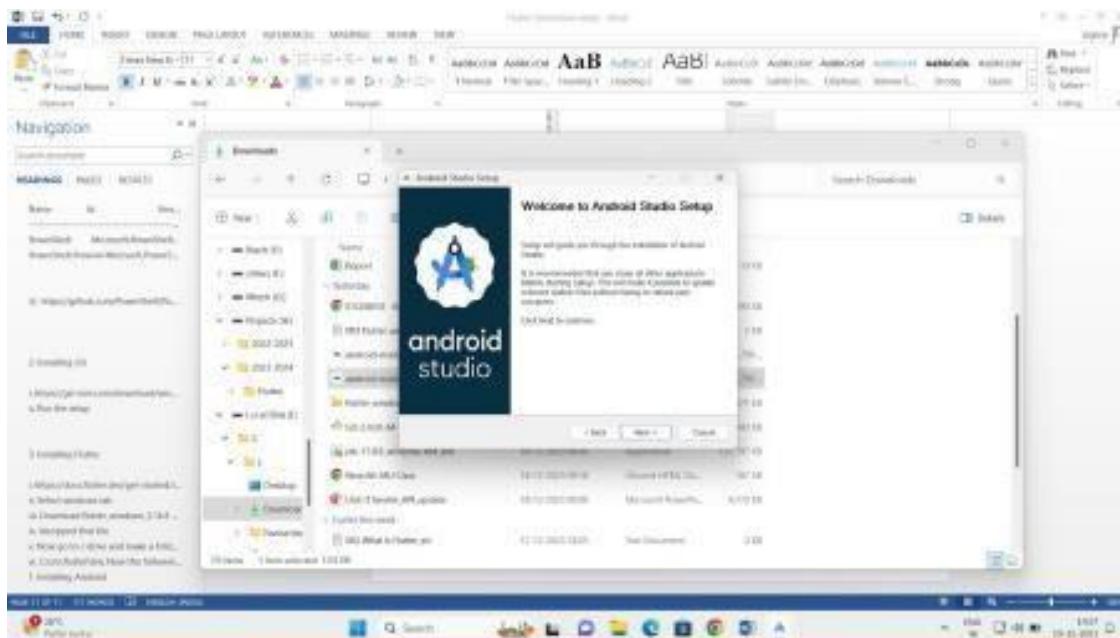
I. Download Android Studio:

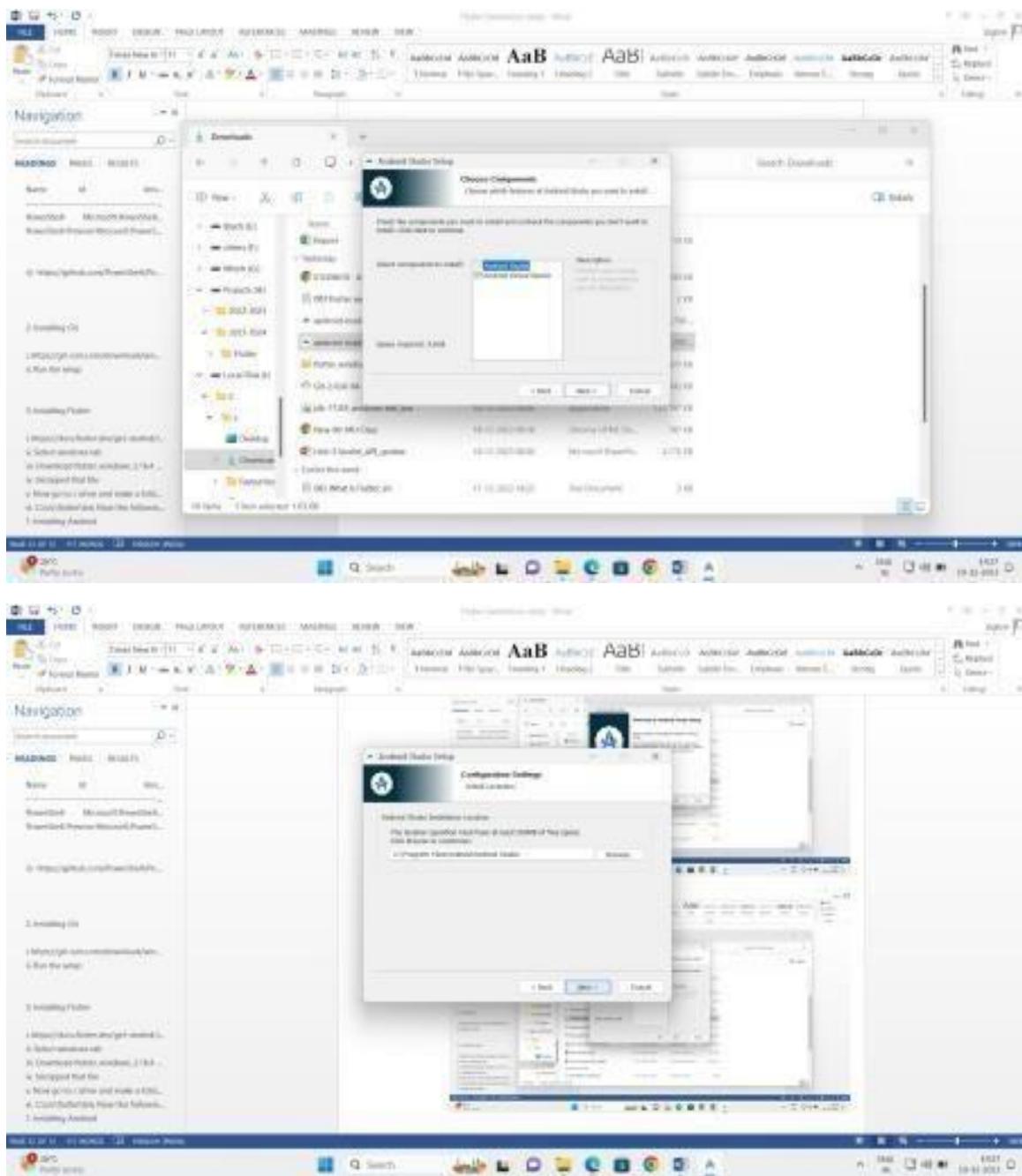
- Visit the Android Studio download page.
- Click on the “Download” button and download the windows version.

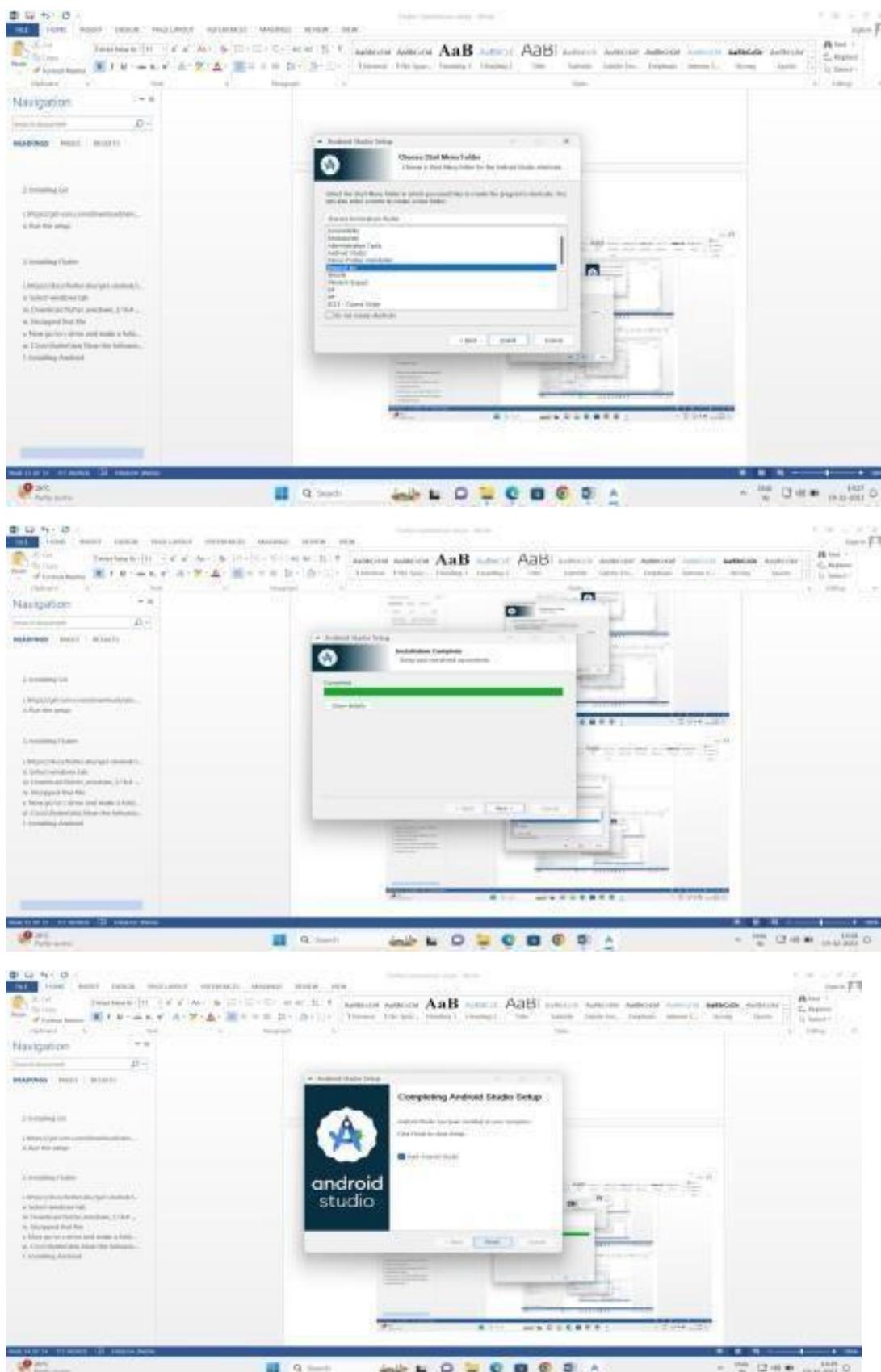
II. Run the Installer:

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

III. Follow Installation Wizard:

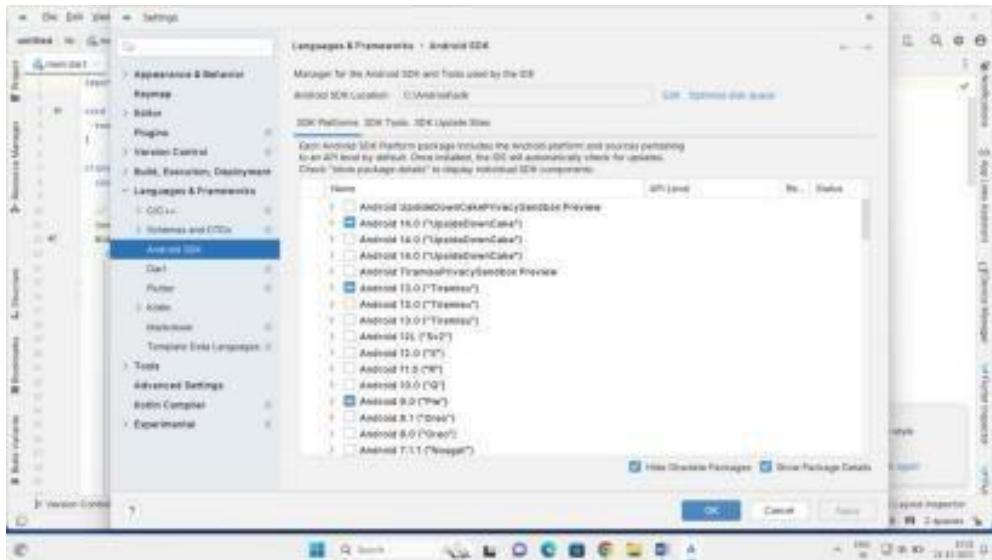




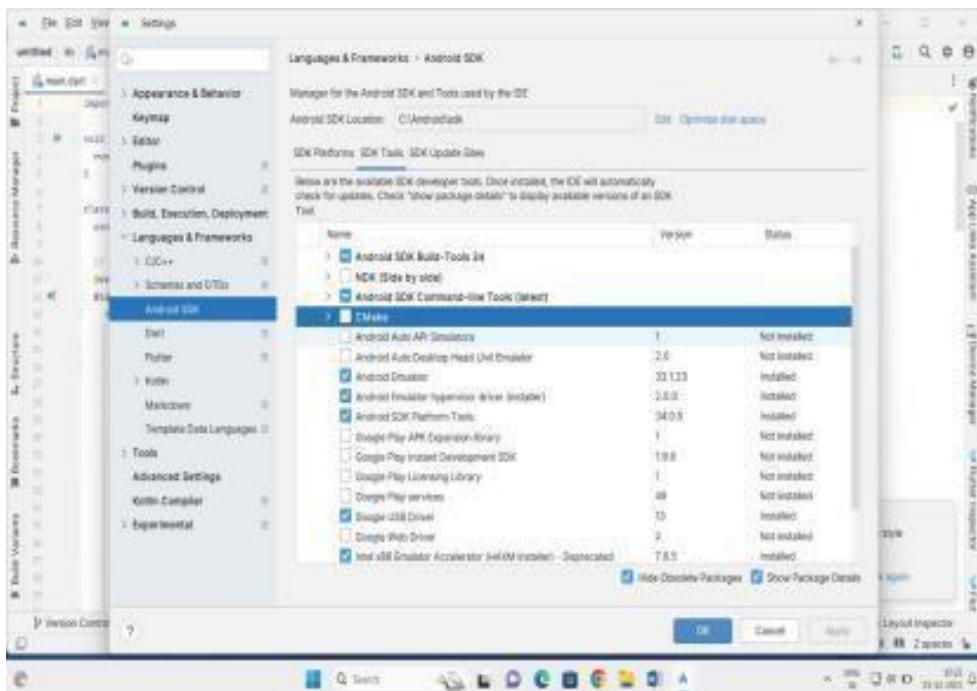




- **Android SDK Platforms:**



- **Android SDK Tools:**



Step 3: Run Following Command for checking Flutter dependencies after installation of android.

- **Accept Android Licenses:** Flutter doctor –android-licenses to develop for android, you need to accept the android licenses.
- Run the following Command: **flutter doctor –android-licenses**



Practical - 2

Title: Create a Hello Flutter 'application.

Code :

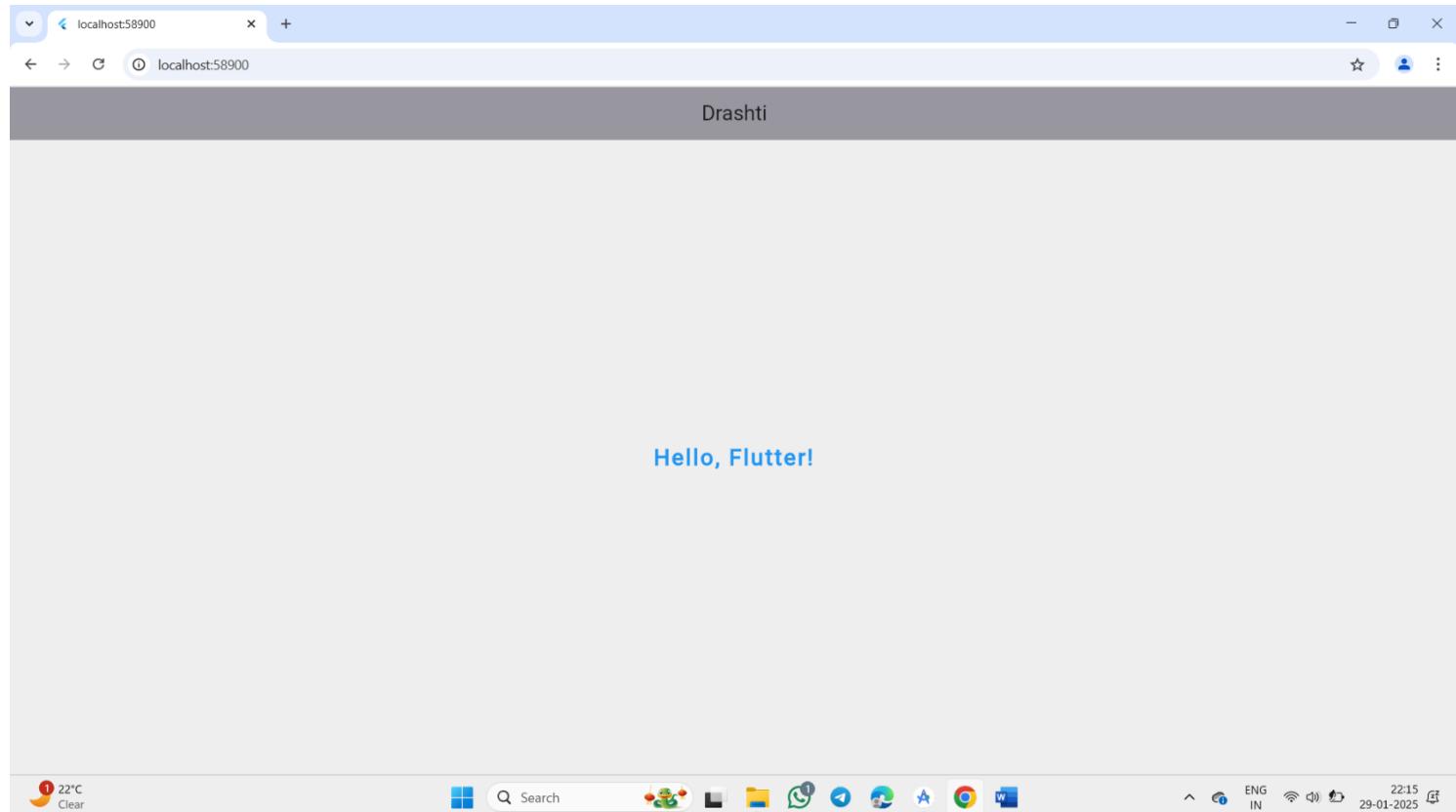
main.dart :

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

void main() {
  runApp(MaterialApp(
    debugShowCheckedModeBanner: false,
    home: Scaffold(
      appBar: AppBar(
        title: Text('MyApp'), // Title of AppBar
        backgroundColor: Colors.grey, // Background color
        centerTitle: true, // Center title
        elevation: 5.0, // Adds shadow
      ),
      backgroundColor: Colors.grey[200], // Background color of Scaffold

      body: Center(
        child: Column(
          mainAxisAlignment: MainAxisAlignment.center,
          children: [
            Text(
              'Hello, Flutter!',
              style: TextStyle(
                fontSize: 24.0, // Font size
                fontWeight: FontWeight.bold, // Font weight
                color: Colors.blue, // Text color
                letterSpacing: 2.0, // Letter spacing
              ),
              textAlign: TextAlign.center, // Align text
            ),
          ],
        ),
      ),
    )));
}
```

Output :





Practical - 3

Title: Create a Hello Flutter 'application.

Code :

main.dart :

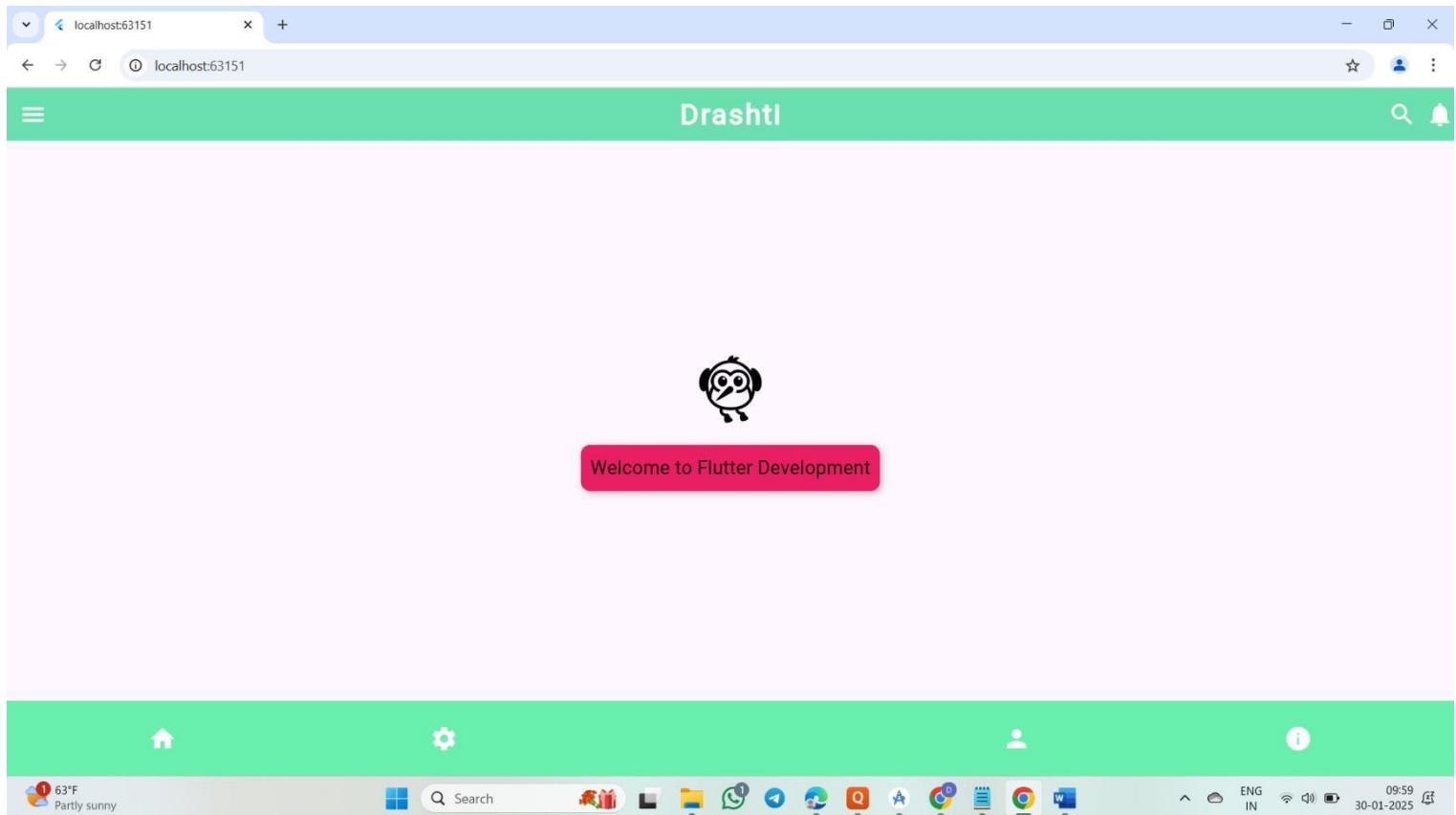
```
import 'package:flutter/material.dart';

void main() {
  runApp(MaterialApp(
    debugShowCheckedModeBanner: false, // Removes the debug banner
    home: Scaffold(
      appBar: AppBar(
        title: Text('Drashti',
          style: TextStyle(
            color: Colors.white, // Text color
            fontSize: 30, // Font size
            fontWeight: FontWeight.bold, // Font weight
            letterSpacing: 2.0, // Spacing between letters
          ),
        ),
        backgroundColor: Colors.greenAccent, // AppBar background color
        centerTitle: true, // Center the title
        elevation: 5.0, // Adds shadow below AppBar
        leading: Icon(Icons.menu, color: Colors.white, size: 30), // Left-side icon
        actions: [
          Icon(Icons.search, color: Colors.white, size: 30), // Right-side icon
          SizedBox(width: 10), // Adds spacing between icons
          Icon(Icons.notifications, color: Colors.white, size: 30),
        ],
      ), // AppBar
      body: Center(
        child: Column(
          mainAxisAlignment: MainAxisAlignment.center, // Center content vertically
          children: [
            Icon(
              Icons.flutter_dash,
              size: 80, // Size of the icon
              color: Colors.black, // Icon color
            ),
            SizedBox(height: 20), // Space between widgets
            Container(
              margin: EdgeInsets.symmetric(horizontal: 20), // Margin around the container
              padding: EdgeInsets.all(10), // Padding inside the container

```




Output :





Practical - 4

Title: Create and application using Flutter Key Widgets.

Code :

main.dart :

```
import 'package:flutter/material.dart';
void main() {
  runApp(
    MaterialApp(
      debugShowCheckedModeBanner: false,
      home: Scaffold(
        backgroundColor: Colors.white,
        appBar: AppBar(
          title: Text("Row & Column UI"),
          backgroundColor: Colors.deepPurpleAccent,
          centerTitle: true,
        ),
        body: Padding(
          padding: EdgeInsets.all(20),
          child: Column(
            mainAxisAlignment: MainAxisAlignment.center,
            children: [
              // Row title
              Text(
                "Popular Technologies",
                style: TextStyle(fontSize: 20, fontWeight: FontWeight.bold),
              ),
              SizedBox(height: 15),
            ],
          ),
        ),
      ),
    ),
  );
}

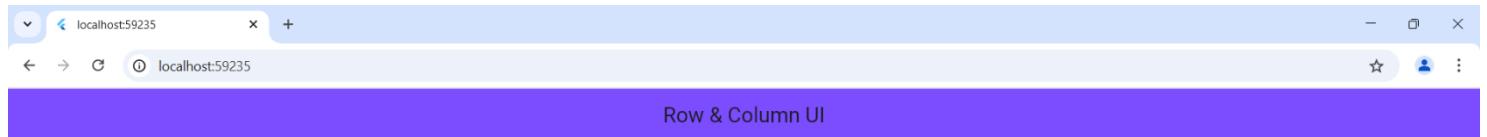
// Row of boxes
Row(
  mainAxisAlignment: MainAxisAlignment.spaceEvenly,
  children: [
    buildBox("Flutter", Colors.lightBlue),
    buildBox("Android", Colors.green),
    buildBox("Firebase", Colors.amber),
  ],
),
SizedBox(height: 40),

// Column title
Text(
  "Top Frameworks",
```



```
style: TextStyle(fontSize: 20, fontWeight: FontWeight.bold),  
),  
SizedBox(height: 15),  
  
// Column of boxes  
buildBox("React.js", Colors.redAccent),  
SizedBox(height: 10),  
buildBox("Flutter", Colors.orange),  
SizedBox(height: 10),  
buildBox("MySQL", Colors.deepOrange),  
],  
,  
,  
,  
);  
}  
  
// Reusable widget for styled boxes  
Widget buildBox(String text, Color color) {  
    return Container(  
        padding: EdgeInsets.symmetric(horizontal: 25, vertical: 12),  
        decoration: BoxDecoration(  
            color: color,  
            borderRadius: BorderRadius.circular(10),  
            boxShadow: [  
                BoxShadow(  
                    color: color.withOpacity(0.4),  
                    blurRadius: 6,  
                    offset: Offset(0, 4),  
                ),  
            ],  
        ),  
        child: Text(  
            text,  
            style: TextStyle(  
                fontSize: 16,  
                fontWeight: FontWeight.bold,  
                color: Colors.white,  
            ),  
        ),  
    );  
}
```

Output :



Popular Technologies

Flutter

Android

Firebase

Top Frameworks

React.js

Flutter

MySQL





Practical - 5

Title: Create and application with Flutter UI Components.

Code :

main.dart :

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

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

class CustomLoginScreen extends StatelessWidget {
  final TextEditingController usernameController = TextEditingController();
  final TextEditingController passwordController = TextEditingController();

  void handleLogin(BuildContext context) {
    String username = usernameController.text;
    String password = passwordController.text;

    if (username.isNotEmpty && password.isNotEmpty) {
      ScaffoldMessenger.of(context).showSnackBar(
        SnackBar(
          content: Text('Login Successful! Welcome $username'),
          backgroundColor: Colors.green,
          duration: Duration(seconds: 2),
        ),
      );
    } else {
      ScaffoldMessenger.of(context).showSnackBar(
        SnackBar(
          content: Text('Please fill in all fields'),
          backgroundColor: Colors.redAccent,
          duration: Duration(seconds: 2),
        ),
      );
    }
  }

  @override
  Widget build(BuildContext context) {
    return Scaffold(
```



```
backgroundColor: Colors.white,  
appBar: AppBar(  
    title: Text("Practical - 5: Login UI"),  
    centerTitle: true,  
    backgroundColor: Colors.deepPurpleAccent,  
,  
body: Padding(  
    padding: const EdgeInsets.symmetric(horizontal: 24.0),  
    child: Column(  
        mainAxisAlignment: MainAxisAlignment.center,  
        children: [  
            CircleAvatar(  
                radius: 60,  
                backgroundColor: Colors.deepPurple,  
                child: Icon(  
                    Icons.lock_outline,  
                    size: 60,  
                    color: Colors.white,  
                ),  
            ),  
            SizedBox(height: 10),  
            Text(  
                "Secure Login",  
                style: TextStyle(fontSize: 18, color: Colors.grey[700]),  
            ),  
            SizedBox(height: 30),  
  
            TextField(  
                controller: usernameController,  
                decoration: InputDecoration(  
                    labelText: "Username or Email",  
                    border: OutlineInputBorder(),  
                    prefixIcon: Icon(Icons.person),  
                ),  
            ),  
            SizedBox(height: 15),  
  
            TextField(  
                controller: passwordController,  
                obscureText: true,  
                decoration: InputDecoration(  
                    labelText: "Password",  
                    border: OutlineInputBorder(),  
                    prefixIcon: Icon(Icons.lock),  
                ),  
            ),  
            SizedBox(height: 30),
```

```

SizedBox(
  width: double.infinity,
  child: ElevatedButton(
    onPressed: () => handleLogin(context),
    style: ElevatedButton.styleFrom(
      backgroundColor: Colors.lightGreenAccent,
      padding: EdgeInsets.symmetric(vertical: 14),
      shape: RoundedRectangleBorder(
        borderRadius: BorderRadius.circular(30),
      ),
    ),
    child: Text("Login", style: TextStyle(fontSize: 16)),
  ),
),
],
),
),
),
),
);
}
}
}

```

Output :

The screenshot shows a web browser window with the URL `localhost:60229`. The title bar reads "Practical - 5: Login UI". The main content area contains a purple circular icon with a white padlock symbol, labeled "Secure Login". Below it are two input fields: one for "Username or Email" with a person icon and one for "Password" with a lock icon. A large green "Login" button is centered below the inputs. At the bottom of the page, a red bar displays the message "Please fill in all fields". The browser's status bar at the very bottom shows various icons and information, including the date and time (14-04-2025).



localhost:60229 localhost:60229

Practical - 5: Login UI

Secure Login

Username or Email
drashtibaraiya7@gmail.com

Password
.....

Login



Practical - 6

Title: Create and application with Flutter UI Components.

Code :

main.dart :

```
import 'package:flutter/material.dart';
void main() {
  runApp(MaterialApp(
    debugShowCheckedModeBanner: false,
    home: RegistrationForm(),
  ));
}

class RegistrationForm extends StatefulWidget {
  @override
  _RegistrationFormState createState() => _RegistrationFormState();
}

class _RegistrationFormState extends State<RegistrationForm> {
  final _formKey = GlobalKey<FormState>();

  final nameController = TextEditingController();
  final emailController = TextEditingController();
  final mobileController = TextEditingController();
  final passwordController = TextEditingController();
  final confirmPasswordController = TextEditingController();

  void _registerUser() {
    if (_formKey.currentState!.validate()) {
      if (passwordController.text == confirmPasswordController.text) {
        ScaffoldMessenger.of(context).showSnackBar(
          SnackBar(
            content: Text("Registration Successful 🎉"),
            backgroundColor: Colors.green,
          ),
        );
      } else {
        ScaffoldMessenger.of(context).showSnackBar(
          SnackBar(
            content: Text("Passwords do not match ✗"),
            backgroundColor: Colors.red,
          ),
        );
      }
    }
  }
}
```



```
}
```

```
@override
Widget build(BuildContext context) {
  return Scaffold(
    appBar: AppBar(
      title: Text("Practical - 6: Register"),
      backgroundColor: Colors.redAccent,
      centerTitle: true,
    ),
    body: SingleChildScrollView(
      padding: EdgeInsets.all(20),
      child: Form(
        key: _formKey,
        child: Column(
          children: [
            SizedBox(height: 10),
            Text(
              "Register New Account",
              style: TextStyle(fontSize: 18, color: Colors.grey[800]),
            ),
            SizedBox(height: 25),

            // Name
            TextFormField(
              controller: nameController,
              decoration: InputDecoration(
                labelText: "Name",
                border: OutlineInputBorder(),
              ),
              validator: (value) =>
                  value!.isEmpty ? "Please enter your name" : null,
            ),
            SizedBox(height: 15),

            // Email
            TextFormField(
              controller: emailController,
              decoration: InputDecoration(
                labelText: "Email Address",
                border: OutlineInputBorder(),
              ),
              validator: (value) =>
                  value!.isEmpty ? "Please enter your email" : null,
            ),
          ],
        ),
      ),
    ),
  );
}
```



SizedBox(height: 15),

```
// Mobile
TextField(
  controller: mobileController,
  keyboardType: TextInputType.phone,
  decoration: InputDecoration(
    labelText: "Mobile Number",
    border: OutlineInputBorder(),
  ),
  validator: (value) =>
  value!.isEmpty ? "Enter mobile number" : null,
),
```

SizedBox(height: 15),

```
// Password
TextField(
  controller: passwordController,
  obscureText: true,
  decoration: InputDecoration(
    labelText: "Password",
    border: OutlineInputBorder(),
  ),
  validator: (value) =>
  value!.isEmpty ? "Enter password" : null,
),
```

SizedBox(height: 15),

```
// Confirm Password
TextField(
  controller: confirmPasswordController,
  obscureText: true,
  decoration: InputDecoration(
    labelText: "Confirm Password",
    border: OutlineInputBorder(),
  ),
  validator: (value) =>
  value!.isEmpty ? "Confirm your password" : null,
),
```

SizedBox(height: 25),

```
// Register Button
SizedBox(
  width: double.infinity,
  child: ElevatedButton(
    onPressed: _registerUser,
    style: ElevatedButton.styleFrom(
```

```
        backgroundColor: Colors.amber,  
        foregroundColor: Colors.black,  
        padding: EdgeInsets.symmetric(vertical: 14),  
        shape: RoundedRectangleBorder(  
            borderRadius: BorderRadius.circular(30),  
        ),  
        ),  
        child: Text("Register", style: TextStyle(fontSize: 16)),  
    ),  
    ),  
    ],  
    ),  
    ),  
    ),  
    );  
}  
}
```

Output :

The screenshot shows a web browser window with the URL `localhost:61069` in the address bar. The page title is "Practical - 6: Register". The main content is a registration form titled "Register New Account". It contains five input fields: "Name", "Email Address", "Mobile Number", "Password", and "Confirm Password". A large yellow button at the bottom is labeled "Register". The browser interface includes standard navigation buttons (back, forward, search) and a toolbar with icons for file operations.

Practical - 6: Register

Register New Account

Name

Email Address

Mobile Number

Password

Confirm Password

Register

localhost:61069

Practical - 6: Register

Register New Account

Name
Please enter your name

Email Address
Please enter your email

Mobile Number
Enter mobile number

Password
Enter password

Confirm Password
Confirm your password

Register



localhost:61069

Practical - 6: Register

Register New Account

Name — Drashti Baraiya

Email Address — drashtibaraiya7@gmail.com

Mobile Number — 98526321000

Password —
.....

Confirm Password —
.....

Register



Practical - 7

Title: Create and application with Flutter UI Components.

Code :

main.dart :

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

class MyApp extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      title: 'Navigation Demo',
      debugShowCheckedModeBanner: false,
      theme: ThemeData(primarySwatch: Colors.deepPurple),
      home: SignUpPage(),
    );
  }
}
// ----- Sign Up Page -----
class SignUpPage extends StatelessWidget {
  final nameController = TextEditingController();
  final emailController = TextEditingController();
  final phoneController = TextEditingController();
  final passwordController = TextEditingController();
  final confirmPasswordController = TextEditingController();

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      backgroundColor: Color(0xFFFF9F5FC),
      body: Padding(
        padding: const EdgeInsets.all(25.0),
        child: Center(
          child: SingleChildScrollView(
            child: Column(
              children: [
                Text("Sign up", style: TextStyle(fontSize: 28, fontWeight: FontWeight.bold)),
                Text("Create your account", style: TextStyle(fontSize: 16, color: Colors.grey[700])),
                SizedBox(height: 30),
              ],
            ),
          ),
        ),
      ),
    );
  }
}
```



```
// Name
    _inputField(Icons.person, "Name", nameController),
    SizedBox(height: 15),

// Email
    _inputField(Icons.email, "Email", emailController),
    SizedBox(height: 15),

// Phone
    _inputField(Icons.phone, "Phone", phoneController),
    SizedBox(height: 15),

// Password
    _inputField(Icons.lock, "Password", passwordController, isPassword: true),
    SizedBox(height: 15),

// Confirm Password
    _inputField(Icons.lock_outline, "Confirm Password", confirmPasswordController,
    isPassword: true),
    SizedBox(height: 30),

// Sign up Button
    SizedBox(
        width: double.infinity,
        height: 50,
        child: ElevatedButton(
            onPressed: () {
                Navigator.push(
                    context,
                    MaterialPageRoute(builder: (context) => WelcomePage()),
                );
            },
            style: ElevatedButton.styleFrom(
                backgroundColor: Colors.lightBlueAccent,
                shape: RoundedRectangleBorder(borderRadius: BorderRadius.circular(30)),
            ),
            child: Text("Sign up", style: TextStyle(fontSize: 18)),
        ),
    ),
    SizedBox(height: 20),
    Text("Already have an account? ", style: TextStyle(color: Colors.grey)),
    TextButton(onPressed: () {}, child: Text("Login")),
],
),
),
),
),
),
```



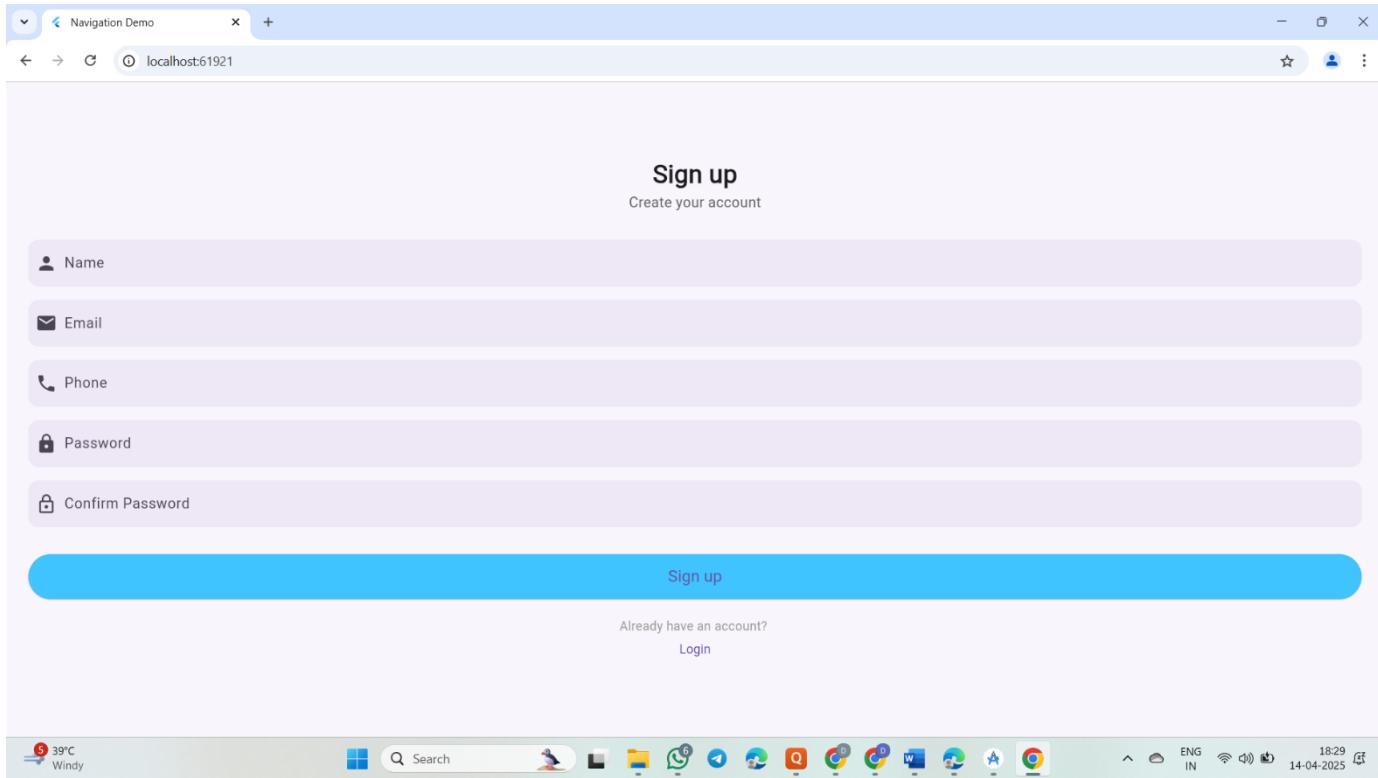
```
};

}

Widget _inputField(IconData icon, String hint, TextEditingController controller, {bool isPassword = false}) {
    return Container(
        decoration: BoxDecoration(
            color: Colors.deepPurple[50],
            borderRadius: BorderRadius.circular(12),
        ),
        child: TextField(
            controller: controller,
            obscureText: isPassword,
            decoration: InputDecoration(
                prefixIcon: Icon(icon),
                border: InputBorder.none,
                hintText: hint,
                contentPadding: EdgeInsets.symmetric(horizontal: 15, vertical: 18),
            ),
        ),
    );
}

// ----- Welcome Page -----
class WelcomePage extends StatelessWidget {
    @override
    Widget build(BuildContext context) {
        return Scaffold(
            appBar: AppBar(
                title: Text("Welcome Screen"),
                backgroundColor: Colors.deepPurple,
            ),
            body: Container(
                color: Colors.deepPurple,
                child: Center(
                    child: Column(
                        mainAxisAlignment: MainAxisAlignment.center,
                        children: [
                            Icon(Icons.waving_hand, size: 80, color: Colors.white),
                            SizedBox(height: 20),
                            Text("Welcome to purple Page!", style: TextStyle(color: Colors.white, fontSize: 24)),
                        ],
                    ),
                ),
            );
    }
}
```

Output :



Sign up
Create your account

Name

Email

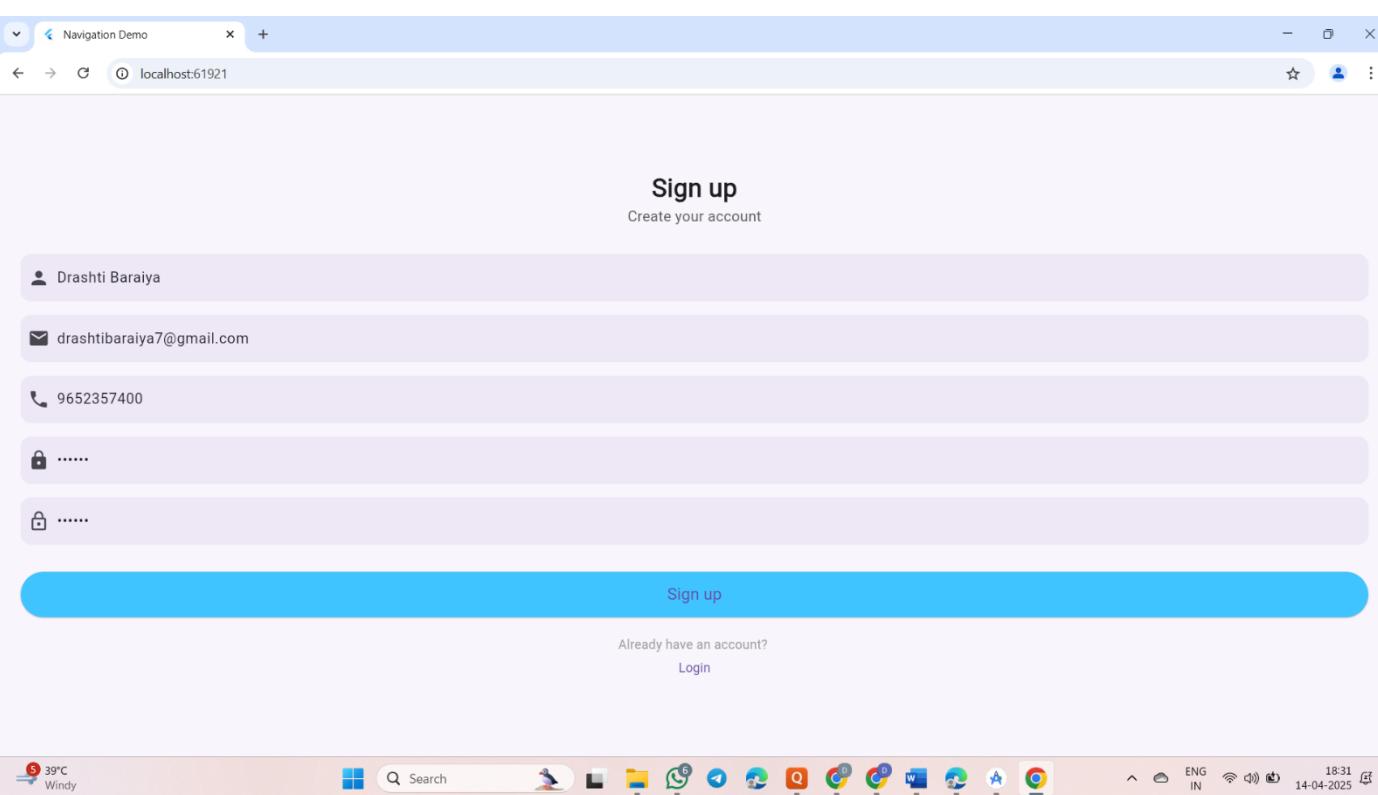
Phone

Password

Confirm Password

Sign up

Already have an account?
[Login](#)



Sign up
Create your account

Drashti Baraiya

drashtibaraiya7@gmail.com

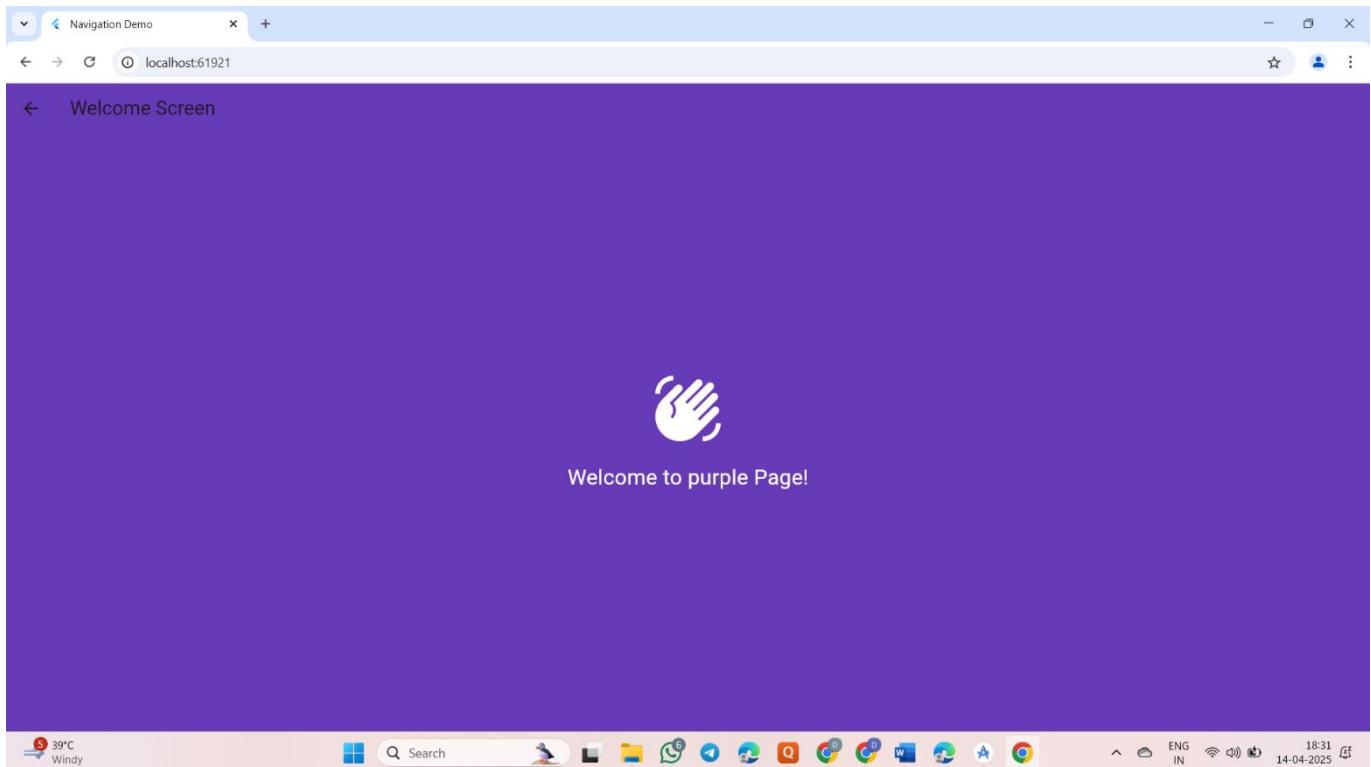
9652357400

.....

.....

Sign up

Already have an account?
[Login](#)





Practical - 8

Title: Create an application with list view in Flutter.

Code :

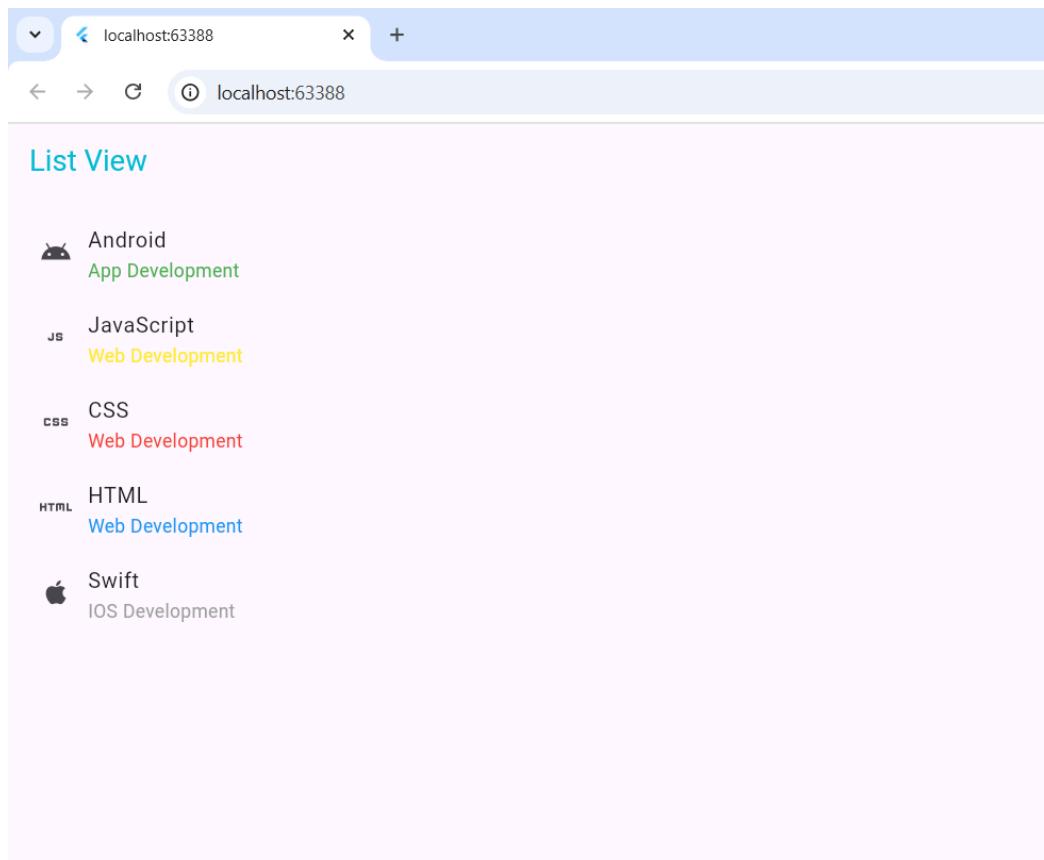
main.dart :

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

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

  // This widget is the root of your application.
  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      debugShowCheckedModeBanner: false,
      home: Scaffold(
        appBar: AppBar(
          title: Text('List View', style: TextStyle(color: Colors.cyan),),
        ),
        body: Center(
          child: Container(
            child: ListView(
              padding: const EdgeInsets.all(8),
              children: <Widget>[
                ListTile(
                  leading: Icon(Icons.android),
                  title: Text('Android'),
                  subtitle: Text("App Development", style: TextStyle(color: Colors.green),),
                ),
                ListTile(
                  leading: Icon(Icons.javascript),
                  title: Text('JavaScript'),
                  subtitle: Text("Web Development", style: TextStyle(color: Colors.yellow),),
                ),
                ListTile(
                  leading: Icon(Icons.css),
                  title: Text('CSS'),
                  subtitle: Text("Web Development", style: TextStyle(color: Colors.red),),
                ),
                ListTile(
                  leading: Icon(Icons.html),
                  title: Text('HTML'),
                  subtitle: Text("Web Development", style: TextStyle(color: Colors.blue),),
                )
              ],
            ),
          ),
        ),
      ),
    );
  }
}
```

```
        ),  
        ListTile(  
            leading: Icon(Icons.apple),  
            title: Text('Swift'),  
            subtitle: Text("IOS Development", style: TextStyle(color: Colors.grey),),  
        ),  
    ],  
),  
),  
),  
),  
),  
),  
);  
}  
}
```

Output :



Practical - 9

Title: Create and application with grid view in Flutter.

Code :

main.dart :

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

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

  @override
  Widget build(BuildContext context) {
    return const MaterialApp(
      title: 'Grid View Example',
      home: GridScreen(),
      debugShowCheckedModeBanner: false,
    );
  }
}

class GridScreen extends StatelessWidget {
  const GridScreen({super.key});
  final List<Map<String, dynamic>> items = const [
    {
      'title': 'Android',
      'subtitle': 'App Development',
      'color': Colors.green,
      'icon': Icons.android,
    },
    {
      'title': 'Web',
      'subtitle': 'HTML & CSS',
      'color': Colors.blue,
      'icon': Icons.language,
    },
    {
      'title': 'Python',
      'subtitle': 'AI / ML',
      'color': Colors.deepPurple,
      'icon': Icons.memory,
    },
  ];
}
```

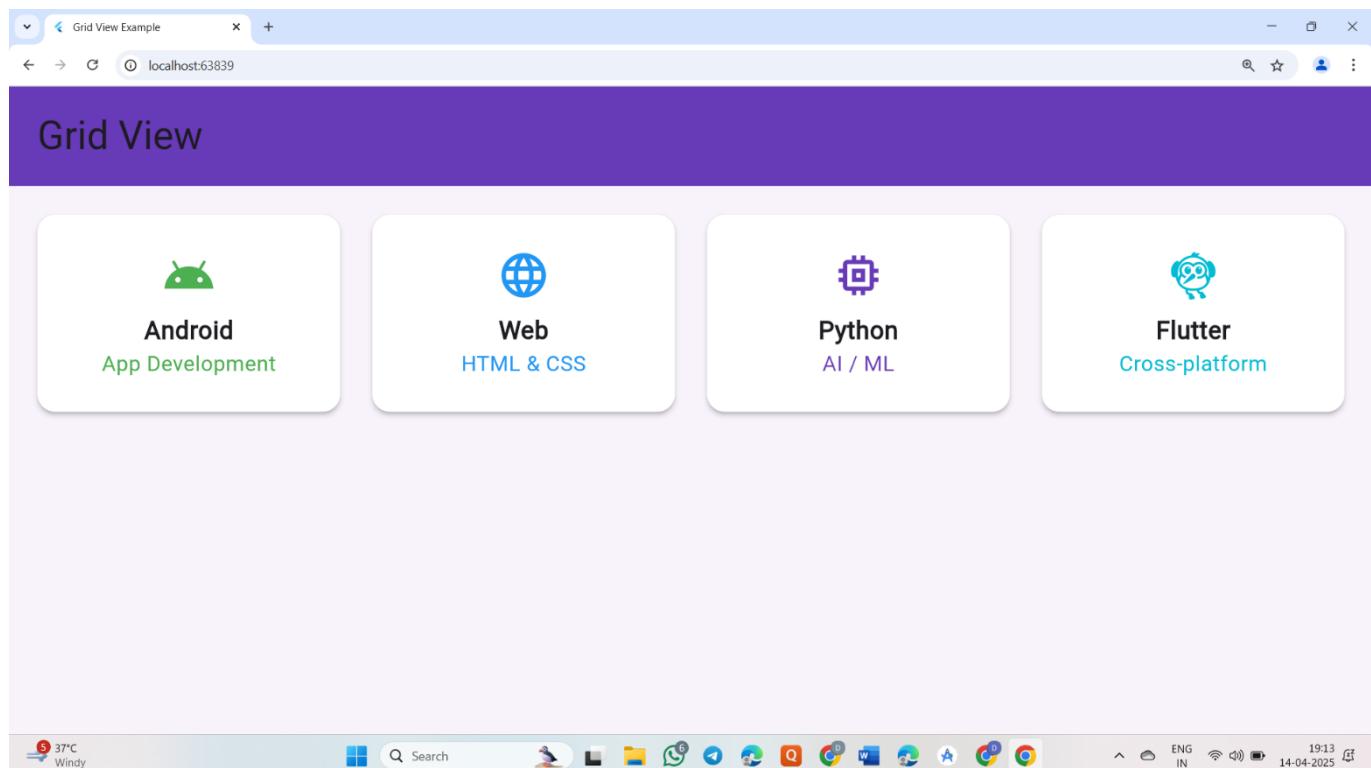


```
'title': 'Flutter',
'subtitle': 'Cross-platform',
'color': Colors.cyan,
'icon': Icons.flutter_dash,
},
];
}

@Override
Widget build(BuildContext context) {
return Scaffold(
  backgroundColor: const Color(0xFFFF8F3FB),
  appBar: AppBar(
    title: const Text("Grid View"),
    backgroundColor: Colors.deepPurple,
  ),
  body: Padding(
    padding: const EdgeInsets.all(12.0),
    child: GridView.builder(
      itemCount: items.length,
      gridDelegate: const SliverGridDelegateWithMaxCrossAxisExtent(
        maxCrossAxisExtent: 200, // Each item max width
        mainAxisSpacing: 10,
        crossAxisSpacing: 10,
        childAspectRatio: 3 / 2, // Smaller cards
      ),
      itemBuilder: (context, index) {
        final item = items[index];
        return Card(
          color: Colors.white,
          elevation: 2,
          shape: RoundedRectangleBorder(
            borderRadius: BorderRadius.circular(10),
          ),
          child: Center(
            child: Column(
              mainAxisAlignment: MainAxisAlignment.spaceEvenly,
              children: [
                Icon(item['icon'], size: 30, color: item['color']),
                const SizedBox(height: 6),
                Text(
                  item['title'],
                  style: const TextStyle(fontWeight: FontWeight.bold),
                ),
                Text(
                  item['subtitle'],
                  style: TextStyle(

```

```
fontSize: 12,  
color: item['color'],  
)  
,  
,  
,  
);  
,  
,  
);  
}  
}
```

Output :

Practical - 10

Title: Create and application Crud Operation with SQLite in Flutter..

Code :

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(
mainAxisSize: MainAxisSize.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? descrption) async { final db = await SQLHelper.db();

final data = { 'title': title, 'description': descrption }; 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);
}

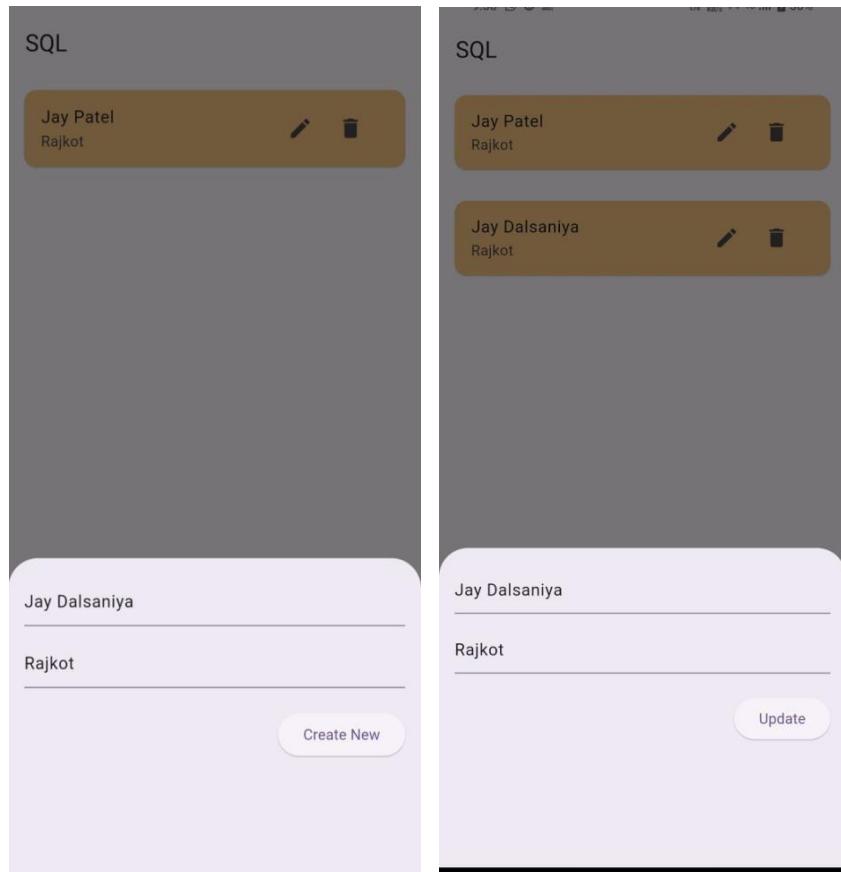
// Update an item by id
static Future<int> updateItem(
int id, String title, String? descrption) async { final db = await SQLHelper.db();

final data = { 'title': title,
'description': descrption,
'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 sqflite: ^2.0.0 path: ^1.9.0
 path_provider: any

Output:



SQL

Jay Patel
Rajkot



SQL

Jay Dalsaniya
Marwadi University



Manan Varmora
Marwadi University



Hitesh Bhandari
Marwadi University



+

+

Successfully deleted a journal!

Practical - 11

Title: Create and application Connecting to REST API in Flutter Code :

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();
},
),
),
);
}
}
}

```

Output:

| Posts | Posts |
|--|--|
| <p>Post 10:</p> <p>optio molestias id quia eum</p> <p>quo et expedita modi cum officia vel magni doloribus qui repudiandae vero nisi sit quos veniam quod sed accusamus veritatis error</p> | <p>Post 36:</p> <p>fuga nam accusamus voluptas reiciendis itaque</p> <p>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>Post 11:</p> <p>et ea vero quia laudantium autem</p> <p>delectus reiciendis molestiae occaecati non minima eveniet qui voluptatibus accusamus in eum beatae sit vel qui neque voluptates ut commodi qui incident ut animi commodi</p> | <p>Post 37:</p> <p>provident vel ut sit ratione est</p> <p>debitus 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>Post 12:</p> <p>in quibusdam tempore odit est dolorem</p> <p>itaque id aut magnam praesentium quia et ea odit et ea voluptas et sapiente quia nihil amet occaecati quia id voluptatem incident ea est distinctio odio</p> | <p>Post 38:</p> <p>explicabo et eos deleniti nostrum ab id repellendus</p> <p>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>Post 13:</p> <p>dolorum ut in voluptas mollitia et saepe quo animi</p> <p>aut dicta possimus sint mollitia voluptas commodi quo doloremque</p> | <p>Post 39:</p> <p>eos dolorem iste accusantium est eaque quam</p> <p>corporis rerum ducimus vel eum accusantium</p> |

Practical - 12

Title: Create and application Parsing JSON data from REST API in Flutter

Code :

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 incident
 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
 incident 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

Title: Create an application using Hardware Interaction in Flutter.

Code :

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(
                mainAxisAlignment: MainAxisAlignment.spaceEvenly,
                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:
