Practical 1:

Android Studio setup for Flutter development 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.

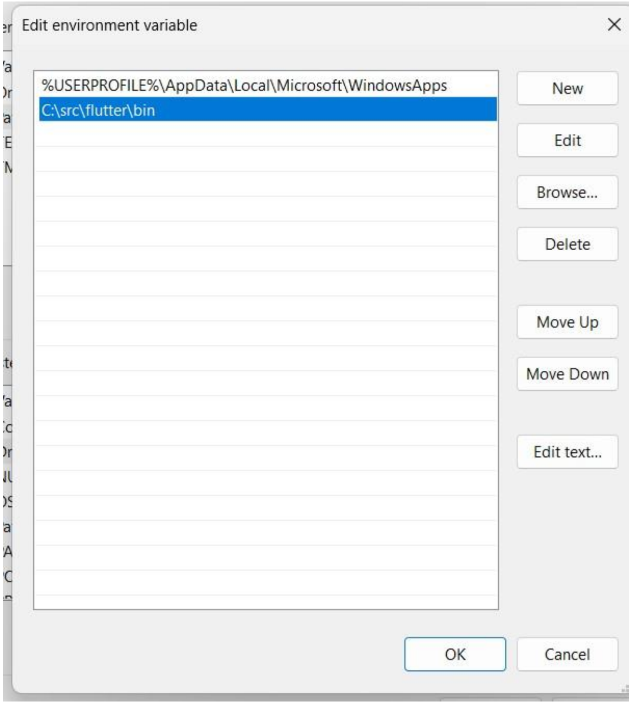
ii. **Download Flutter:**

- 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. **Set Up Environment Variables:**

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



Edit environment variable

%USERPROFILE%\AppData\Local\Microsoft\WindowsApps
C:\src\flutter\bin

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 instructions provided by flutter doctor to install any missing dependencies. This may include things like Android Studio, Xcode commandline tools, etc.

Step 2: Installing Android Studio.

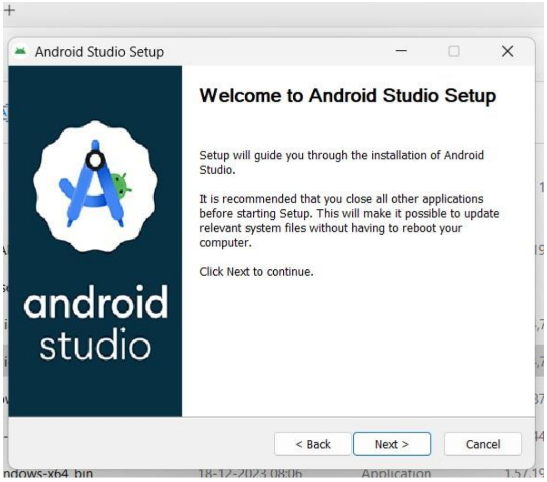
i. **Download Android Studio:** ○ Visit the Android Studio download page.

- 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:**



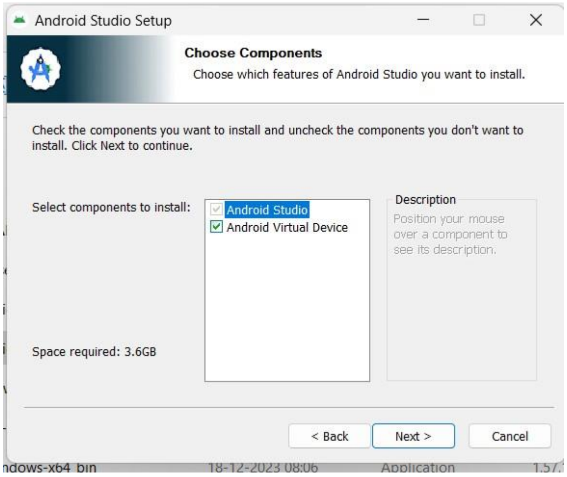
Android Studio Setup

Welcome to Android Studio Setup

Setup will guide you through the installation of Android Studio.

It is recommended that you close all other applications before starting Setup. This will make it possible to update relevant system files without having to reboot your computer.

Click Next to continue.



Android Studio Setup

Choose Components

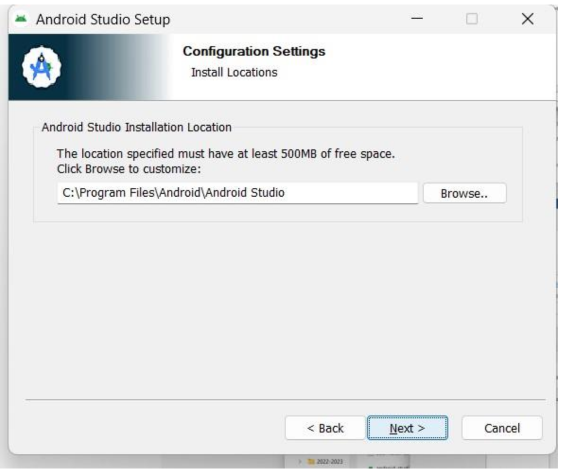
Choose which features of Android Studio you want to install.

Check the components you want to install and uncheck the components you don't want to install. Click Next to continue.

Select components to install:
☑ Android Studio
☑ Android Virtual Device

Description
Position your mouse over a component to see its description.

Space required: 3.6GB



Android Studio Setup

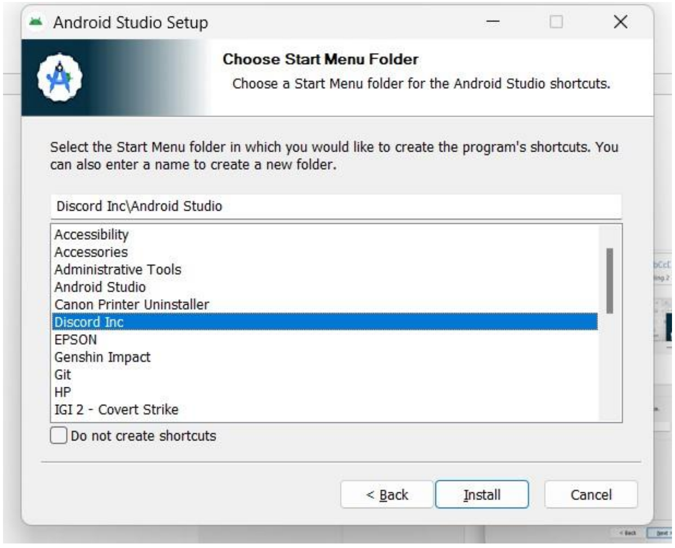
Configuration Settings

Install Locations

Android Studio Installation Location

The location specified must have at least 500MB of free space.
Click Browse to customize:

C:\Program Files\Android\Android Studio



Android Studio Setup

Choose Start Menu Folder

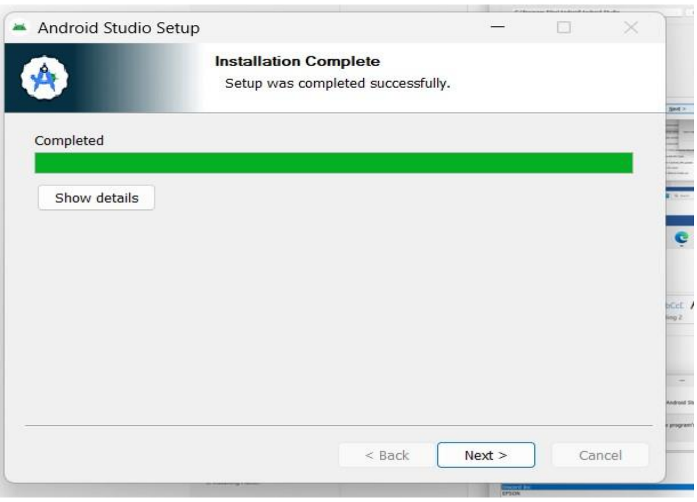
Choose a Start Menu folder for the Android Studio shortcuts.

Select the Start Menu folder in which you would like to create the program's shortcuts. You can also enter a name to create a new folder.

Discord Inc\Android Studio

Accessibility
Accessories
Administrative Tools
Android Studio
Canon Printer Uninstaller
Discord Inc
EPSON
Genshin Impact
Git
HP
IGI 2 - Covert Strike

□ Do not create shortcuts

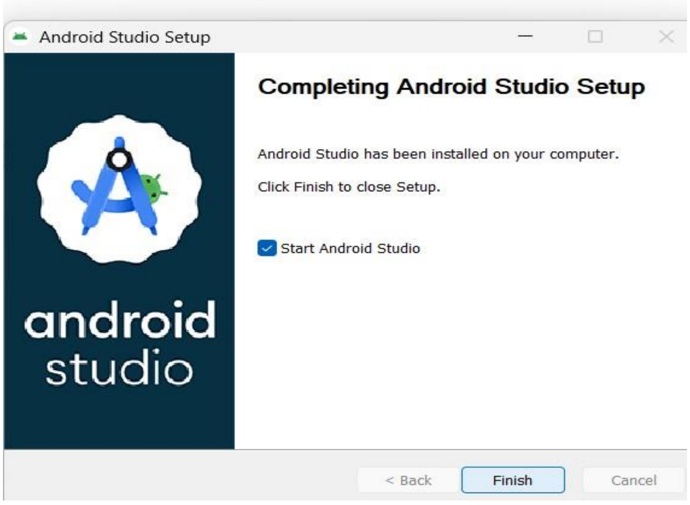


Android Studio Setup

Installation Complete

Setup was completed successfully.

Completed



Android Studio Setup

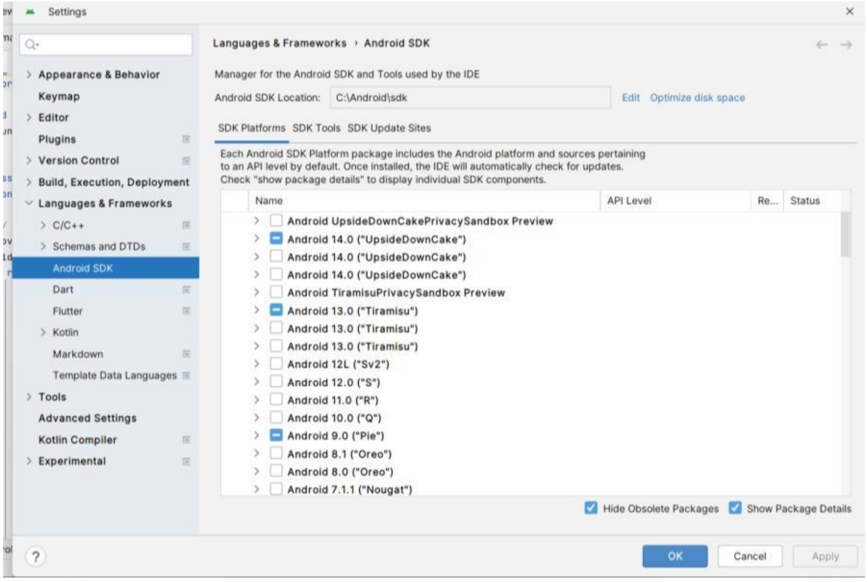
Completing Android Studio Setup

Android Studio has been installed on your computer.

Click Finish to close Setup.

☑ Start Android Studio

- **Android SDK Platforms**:



Settings

Languages & Frameworks › Android SDK

Manager for the Android SDK and Tools used by the IDE

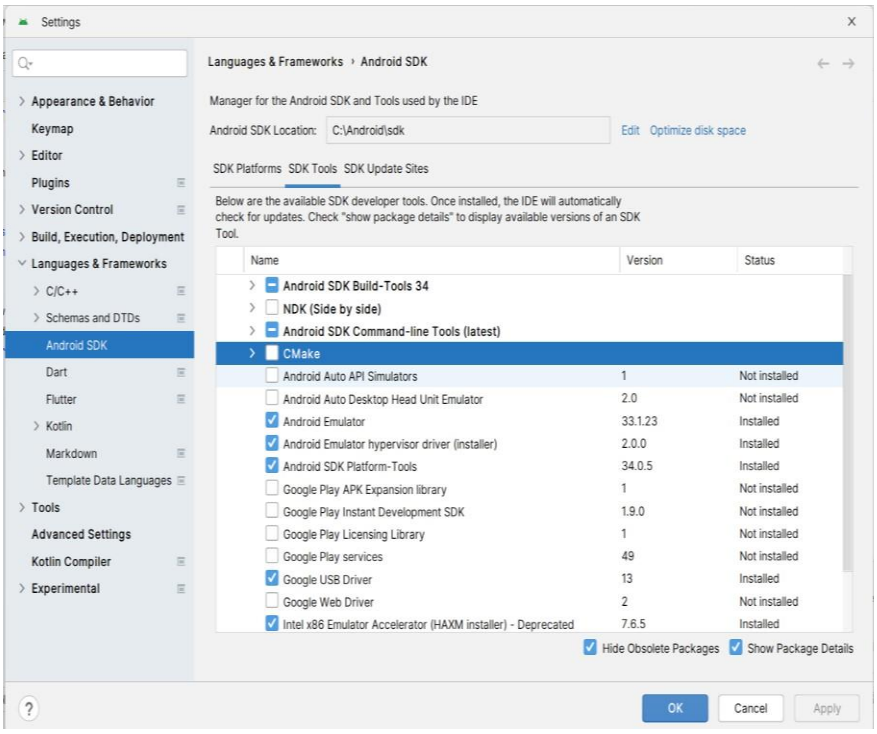
Android SDK Location: C:\Android\sdk

Each Android SDK Platform package includes the Android platform and sources pertaining to an API level by default. Once installed, the IDE will automatically check for updates. Check "show package details" to display individual SDK components.

	Name	API Level	Re...	Status
☐	Android UpsideDownCakePrivacySandbox Preview			
⊟	Android 14.0 ("UpsideDownCake")			
☐	Android 14.0 ("UpsideDownCake")			
☐	Android 14.0 ("UpsideDownCake")			
☐	Android TiramisuPrivacySandbox Preview			
⊟	Android 13.0 ("Tiramisu")			
☐	Android 13.0 ("Tiramisu")			
☐	Android 13.0 ("Tiramisu")			
☐	Android 12L ("Sv2")			
☐	Android 12.0 ("S")			
☐	Android 11.0 ("R")			
☐	Android 10.0 ("Q")			
⊟	Android 9.0 ("Pie")			
☐	Android 8.1 ("Oreo")			
☐	Android 8.0 ("Oreo")			
☐	Android 7.1.1 ("Nougat")			

☑ Hide Obsolete Packages ☑ Show Package Details

- **Android SDK Tools:**



Languages & Frameworks › Android SDK

Manager for the Android SDK and Tools used by the IDE

Android SDK Location: C:\Android\sdk Edit Optimize disk space

SDK Platforms SDK Tools SDK Update Sites

Below are the available SDK developer tools. Once installed, the IDE will automatically check for updates. Check "show package details" to display available versions of an SDK Tool.

Name	Version	Status
▣ Android SDK Build-Tools 34		
□ NDK (Side by side)		
▣ Android SDK Command-line Tools (latest)		
□ CMake		
□ Android Auto API Simulators	1	Not installed
□ Android Auto Desktop Head Unit Emulator	2.0	Not installed
☑ Android Emulator	33.1.23	Installed
☑ Android Emulator hypervisor driver (installer)	2.0.0	Installed
☑ Android SDK Platform-Tools	34.0.5	Installed
□ Google Play APK Expansion library	1	Not installed
□ Google Play Instant Development SDK	1.9.0	Not installed
□ Google Play Licensing Library	1	Not installed
□ Google Play services	49	Not installed
☑ Google USB Driver	13	Installed
□ Google Web Driver	2	Not installed
☑ Intel x86 Emulator Accelerator (HAXM installer) - Deprecated	7.6.5	Installed

☑ Hide Obsolete Packages ☑ Show Package Details

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

iv. **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**

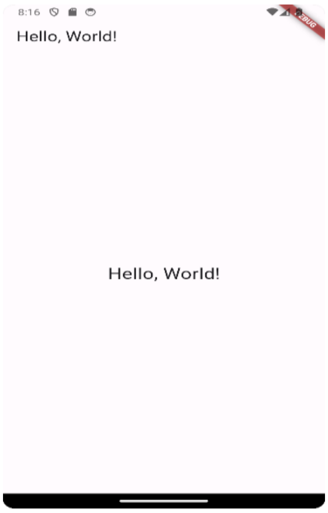
Practical 2:

Create a “Hello Flutter” application.

Solution:

```
// main.dart

import 'package:flutter/material.dart';

void main() {
runApp(MyApp());
}
class MyApp extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
return MaterialApp(      home:
Scaffold(        appBar: AppBar(
          title: Text('Hello, World!'),
        ),
        body: Center(          child: Text(
'Hello, World!',            style:
TextStyle(fontSize: 24),
          ),
        ),
      ),
    );
  }
}
```

Output:



Hello, World!

Hello, World!

PRACTICAL

AIM:

3

Create an application using Flutter Key Widgets(Row ,Text button/Outlined Button/Elevated Button, Container, Image)

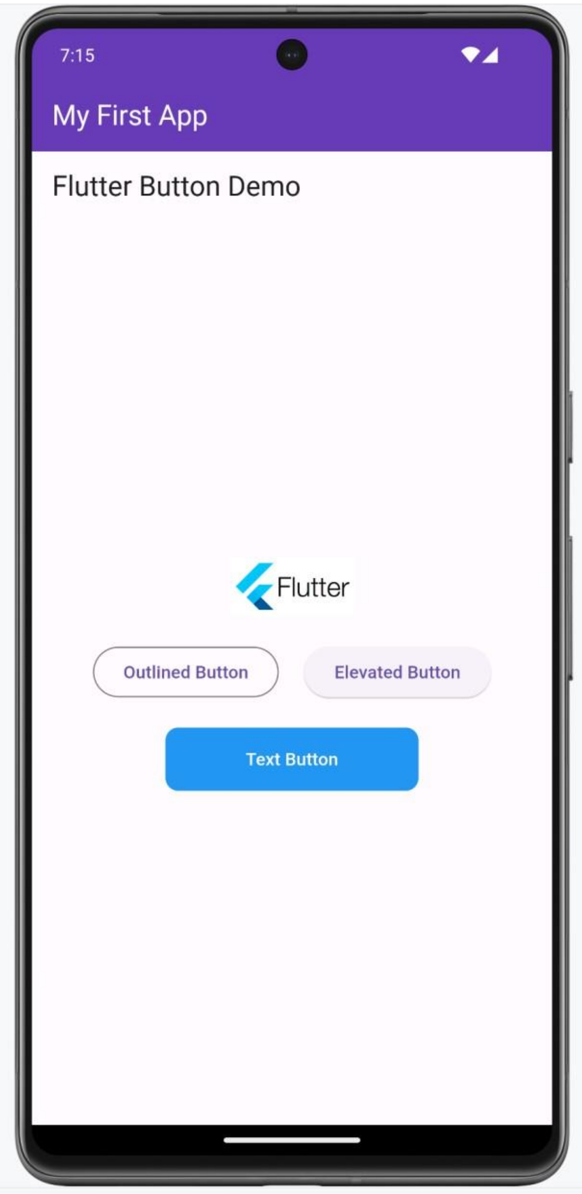
SOLUTION:

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

class Lab_3 extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
return MaterialApp(
      debugShowCheckedModeBanner: false,
home: Scaffold(         appBar: AppBar(
          title: Text('Flutter Button Demo'),
        ),         body:
Center(           child:
Column(
            mainAxisAlignment: MainAxisAlignment.center,
            children: [
Image.asset(
                'assets/flutter_logo.jpg',
width: 100,
              ),
              SizedBox(height: 20),
Row(
                mainAxisAlignment: MainAxisAlignment.center,
                children: [
                  OutlinedButton(
onPressed: () {                       //
Handle button press
                    },
                    child: Text('Outlined Button'),
                  ),
                  SizedBox(width: 20),
                  ElevatedButton(
onPressed: () {                       //
Handle button press
                    },
                    child: Text('Elevated Button'),
                  ),
                ],
              ),
              SizedBox(height: 20),
Container(                 width:
200,                 height: 50,
child: TextButton(
onPressed: () {
                    // Handle button press
                  },
child: Text(
                    'Text Button',
                    style: TextStyle(color: Colors.white),
),
                ),
                decoration: BoxDecoration(
borderRadius: BorderRadius.circular(10),
color: Colors.blue,
                ),
              ),
            ],
          ),
        ),
      ),
    );
  } }
```

OUTPUT:

PRACTICAL

AIM:



4

Create an application using Flutter Key Widgets(Column, Icon Button/Floating Button, Rich Text, Container)

SOLUTION:

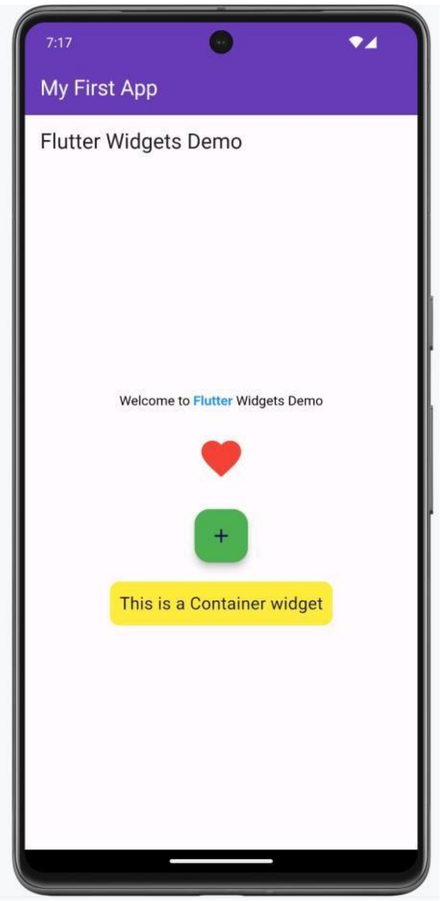
```
import 'package:flutter/material.dart';

class Lab_4 extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
return MaterialApp(
      debugShowCheckedModeBanner: false,
home: Scaffold(         appBar: AppBar(
          title: Text('Flutter Widgets Demo'),
        ),          body:
Center(           child:
Column(
            mainAxisAlignment: MainAxisAlignment.center,
            children: [
Container(
                margin: EdgeInsets.only(bottom: 20),
child: RichText(                   text: TextSpan(
text: 'Welcome to ',
                    style: TextStyle(color: Colors.black),
                    children: <TextSpan>[
TextSpan(                          text: 'Flutter ',
style: TextStyle(
fontWeight: FontWeight.bold,
color: Colors.blue,
                          ),
                        ),
                      TextSpan(text: 'Widgets Demo'),
                    ],
                  ),
                ),
              ),
Container(
                margin: EdgeInsets.only(bottom: 20),
child: IconButton(                   onPressed: () {
// Handle button press
                  },
                  icon: Icon(Icons.favorite),
color: Colors.red,                   iconSize:
50,
                ),
              ),
Container(
                margin: EdgeInsets.only(bottom: 20),
child: FloatingActionButton(
onPressed: () {                    // Handle button
press
                  },
                  child: Icon(Icons.add),
backgroundColor: Colors.green,
                ),
              ),
Container(
                padding: EdgeInsets.all(10),
decoration: BoxDecoration(
color: Colors.yellow,
                  borderRadius: BorderRadius.circular(10),
                ),
child: Text(
```

PRACTICAL

AIM:

```
                  'This is a Container widget',
style: TextStyle(fontSize: 18),
                ),
              ),
            ],
          ),
        ),
      ),
    );
  } }
```

OUTPUT:



5

Create an application using Flutter UI Components (UI Design)

SOLUTION:

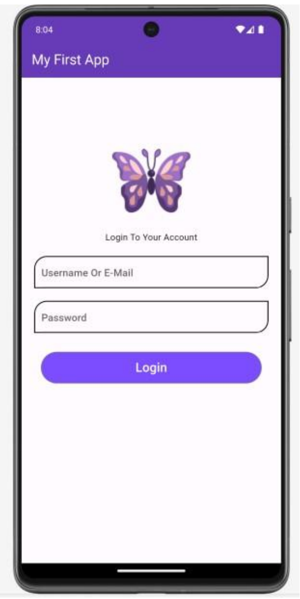
```
import 'package:flutter/material.dart';
import '../widget_learning/rich_text.dart';

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

  State<LoginPageToRichText> createState() => _LoginPageToRichTextState();
}
class _LoginPageToRichTextState extends State<LoginPageToRichText> {
  @override
  Widget build(BuildContext context) {
return Scaffold(
          body: SingleChildScrollView(
child: Column(               children:
[
                const SizedBox(height: 100),
Image.asset("assets/butterfly.png"),
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 E-Mail",
                    ),
                    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",
                    ),
                  ),
),
```

PRACTICAL

AIM:

```
                const SizedBox(height: 30),
Container(                   height: 50,
width: 350,                   child:
OutlinedButton(
onPressed: () {
                      Navigator.push(context, MaterialPageRoute(builder:
(context) => const StyledRichText()));
                    },
                    style: ButtonStyle(
foregroundColor:
MaterialStateProperty.all(Colors.white),
backgroundColor:

MaterialStateProperty.all(Colors.deepPurpleAccent),
                    ),
                    child: const Text(
                      "Login",
style: TextStyle(
fontSize: 20,
                      ),
                    ),
                  ),
                ),
              ],
            ),
          ),
        );
  } }
```

OUTPUT:



6

Create an application using Flutter UI Components that performs Custom Gesture (Page Validation)

SOLUTION:

loginpage_validation.dart

```
import 'package:flutter/material.dart'; import
'custom_gesture.dart';
 class LoginPageFormValidation extends StatefulWidget {
const LoginPageFormValidation({Key? key}) : super(key: key);

  @override
  State<LoginPageFormValidation> createState() =>
      _LoginPageFormValidationState();
}
class _LoginPageFormValidationState extends State<LoginPageFormValidation>
{
  final GlobalKey<FormState> _formKey = GlobalKey<FormState>();

  TextEditingController _username = TextEditingController();
  TextEditingController _pass = TextEditingController();

  @override
  Widget build(BuildContext context) {
return Scaffold(       body:
SingleChildScrollView(         child:
Form(           key: _formKey,
child: Column(             children: [
              const SizedBox(height: 100),
Image.asset("assets/butterfly.png"),
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: TextFormField(
controller: _username,
                  // keyboardType: TextInputType.number,
decoration: const InputDecoration(
hintText: "Username Or E-Mail",
                  ),
                  validator: (value) {
                    if (value == null || value.isEmpty) {
                      return 'Please Enter Your Username or E-Mail';
                    }
                    return null;
```

PRACTICAL

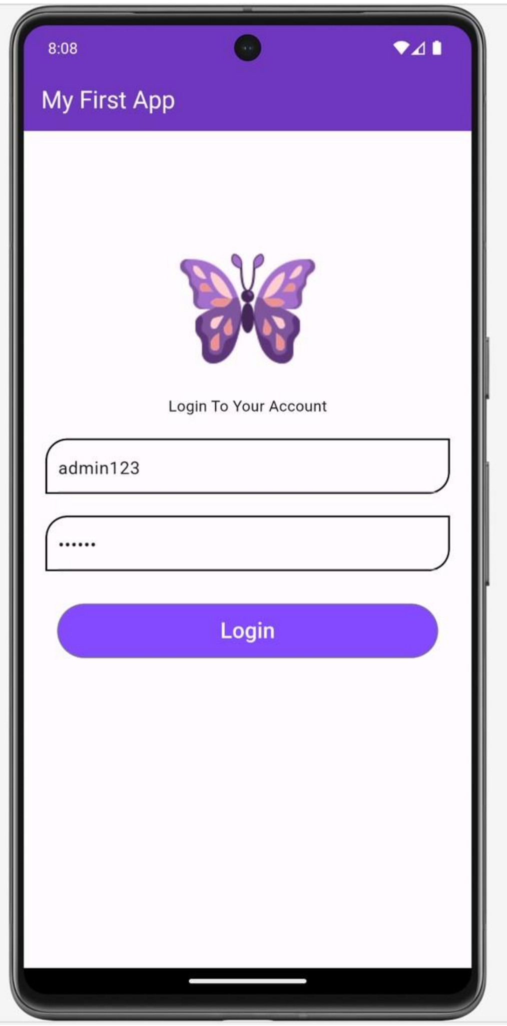
AIM:

```
                  },
                  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.red;
  String text1="Red";
  @override
  Widget build(BuildContext context) {
return Scaffold(       body:
GestureDetector(         onTap: () {
setState(() {
if(color1==Colors.red)
              {
                color1=Colors.blue;
text1="Blue";
              }
else
{
                color1=Colors.red;
text1="Red";
              }
          });       },       child:
Container(             height: 100,
width: 100,         color: color1,
child:  Center(              child:
Text(text1,            style: const
TextStyle(                   color:
Colors.white,
            )
          ),
          ),
        ),
      ),
    );
  }

}
```

OUTPUT:



8:08

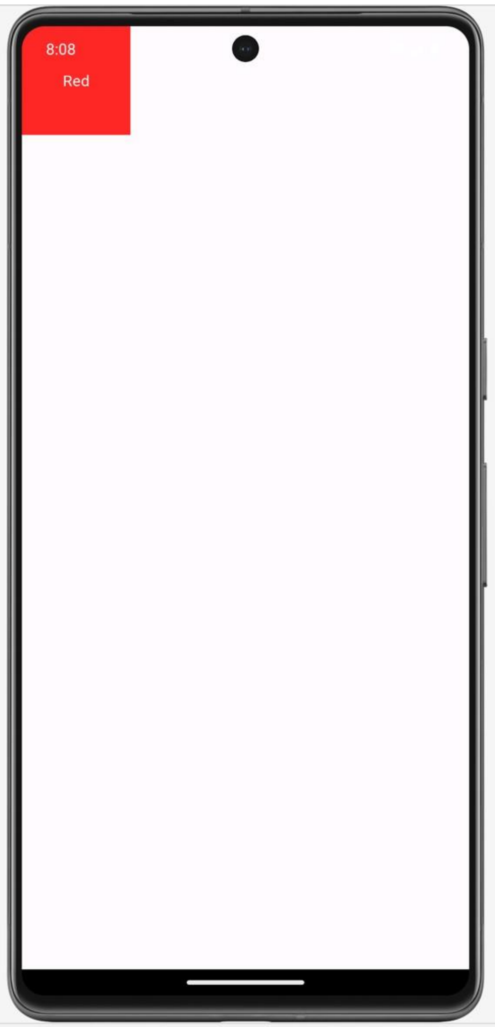
My First App

Login To Your Account

admin123

......

Login



8:08

Red

PRACTICAL 7

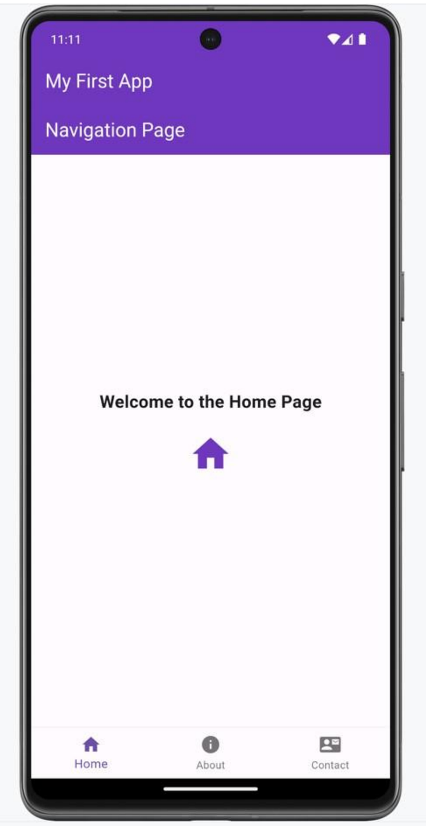
AIM: Create an application with navigation in Flutter

SOLUTION:

navigation_page.dart

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

class NavigationPage extends StatefulWidget {
  @override
  _NavigationPageState createState() => _NavigationPageState();
}
class _NavigationPageState extends State<NavigationPage> {
int _currentIndex = 0;

  final List<Widget> _pages = [
    HomePage(),
    AboutPage(),
    ContactPage(),
  ];

  @override
  Widget build(BuildContext context) {
return Scaffold(       appBar: AppBar(
        title: Text("Navigation Page"),
backgroundColor: Colors.deepPurple,
foregroundColor: Colors.white,
      ),
      body: _pages[_currentIndex],
bottomNavigationBar: BottomNavigationBar(
currentIndex: _currentIndex,         onTap:
(index) {           setState(() {
            _currentIndex = index;
          });
},
items: [
          BottomNavigationBarItem(icon: Icon(Icons.home), label: "Home"),
BottomNavigationBarItem(icon: Icon(Icons.info), label: "About"),
BottomNavigationBarItem(
              icon: Icon(Icons.contact_mail), label: "Contact"),
],
      ),
    );
  }
}  class HomePage extends
StatelessWidget {
  @override
  Widget build(BuildContext context) {
return Center(       child: Column(
        mainAxisAlignment: MainAxisAlignment.center,
        children: [
          Text(
            "Welcome to the Home Page",
            style: TextStyle(fontSize: 20, fontWeight: FontWeight.bold),
          ),
          SizedBox(height: 20),
          Icon(Icons.home, size: 50, color: Colors.deepPurple),
          SizedBox(height: 20),
        ],
      ),
    );
  }
}
class AboutPage extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
return Center(       child: Column(
        mainAxisAlignment: MainAxisAlignment.center,
children: [           Text(             "About Us",
            style: TextStyle(fontSize: 20, fontWeight: FontWeight.bold),
          ),
          SizedBox(height: 20),
          Text("Learn more about our company and mission."),
          SizedBox(height: 20),
          Text("Additional information about your company."),
          SizedBox(height: 20),
          Icon(Icons.info, size: 50, color: Colors.deepPurple),
        ],
      ),
    );
  }
}
class ContactPage extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
return Center(       child: Column(
        mainAxisAlignment: MainAxisAlignment.center,
children: [
          Text(
            "Contact Us",
            style: TextStyle(fontSize: 20, fontWeight: FontWeight.bold),
),
          SizedBox(height: 20),
          Text("Reach out to us for any inquiries or support."),
          SizedBox(height: 20),
          TextFormField(
            decoration: InputDecoration(
labelText: 'Your Name',
            ),
          ),
          SizedBox(height: 10),
          TextFormField(
            decoration: InputDecoration(
              labelText: 'Email Address',
            ),
          ),
          SizedBox(height: 20),
          ElevatedButton(
onPressed: () {
              // Add your action for the form submission
            },
            child: Text('Submit'),
          ),
          SizedBox(height: 20),
          Icon(Icons.contact_mail, size: 50, color: Colors.deepPurple),
        ],
      ),
    );
  } }
```

OUTPUT:

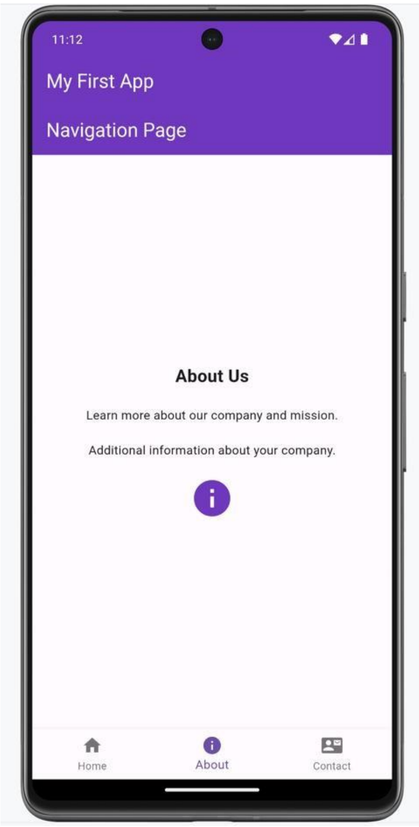


11:11

My First App

Navigation Page

Welcome to the Home Page



11:12

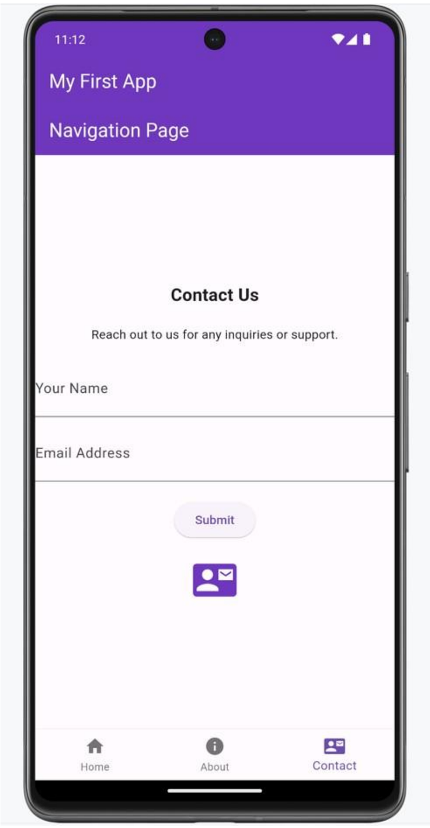
My First App

Navigation Page

About Us

Learn more about our company and mission.

Additional information about your company.



11:12

My First App

Navigation Page

Contact Us

Reach out to us for any inquiries or support.

Your Name

Email Address

PRACTICAL 8

AIM: Create an application in Flutter using List View

SOLUTION:

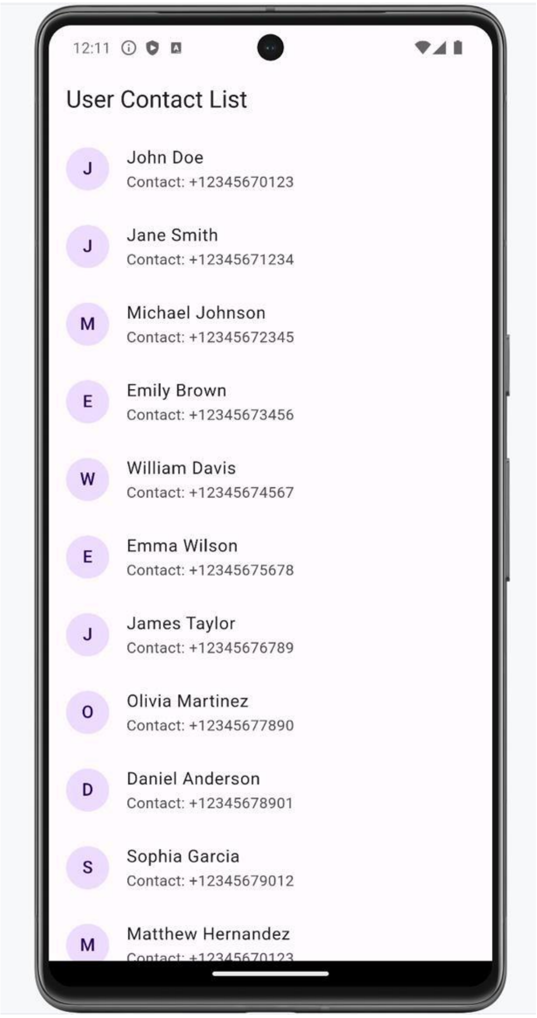
main.dart

```
import 'package:flutter/material.dart';
import 'list_view.dart'; void main() {
runApp(MyApp());
}
class MyApp extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
return MaterialApp(
      debugShowCheckedModeBanner: false,
home: MyListView (),
    );
  } }
```

list_view.dart

```
import 'package:flutter/material.dart';
 class MyListView extends StatelessWidget {
final List<Map<String, dynamic>> items = [
    {'name': 'John Doe', 'contactNumber': '+12345670123'},
    {'name': 'Jane Smith', 'contactNumber': '+12345671234'},
    {'name': 'Michael Johnson', 'contactNumber': '+12345672345'},
    {'name': 'Emily Brown', 'contactNumber': '+12345673456'},
    {'name': 'William Davis', 'contactNumber': '+12345674567'},
    {'name': 'Emma Wilson', 'contactNumber': '+12345675678'},
    {'name': 'James Taylor', 'contactNumber': '+12345676789'},
    {'name': 'Olivia Martinez', 'contactNumber': '+12345677890'},
    {'name': 'Daniel Anderson', 'contactNumber': '+12345678901'},
    {'name': 'Sophia Garcia', 'contactNumber': '+12345679012'},
    {'name': 'Matthew Hernandez', 'contactNumber': '+12345670123'},
    {'name': 'Isabella Lopez', 'contactNumber': '+12345671234'},
    {'name': 'David Gonzalez', 'contactNumber': '+12345672345'},
    {'name': 'Charlotte Perez', 'contactNumber': '+12345673456'},
    {'name': 'Andrew Wilson', 'contactNumber': '+12345674567'},
];

  @override
  Widget build(BuildContext context) {
return Scaffold(       appBar: AppBar(
        title: const Text('User Contact List'),
      ),
      body: ListView.builder(
itemCount: items.length,
itemBuilder: (context, index) {
return ListTile(
            leading: CircleAvatar(
              child: Text(items[index]['name'][0]), // Display first letter
of name              ),
            title: Text(items[index]['name']),
            subtitle: Text('Contact: ${items[index]['contactNumber']}'),
            // Add any additional widgets here
          );
        },
      ),
    );
  } }
```

OUTPUT:



User Contact List

J	John Doe Contact: +12345670123
J	Jane Smith Contact: +12345671234
M	Michael Johnson Contact: +12345672345
E	Emily Brown Contact: +12345673456
W	William Davis Contact: +12345674567
E	Emma Wilson Contact: +12345675678
J	James Taylor Contact: +12345676789
O	Olivia Martinez Contact: +12345677890
D	Daniel Anderson Contact: +12345678901
S	Sophia Garcia Contact: +12345679012
M	Matthew Hernandez Contact: +12345670123

PRACTICAL 9

AIM: Create an application in Flutter using Grid View

SOLUTION:

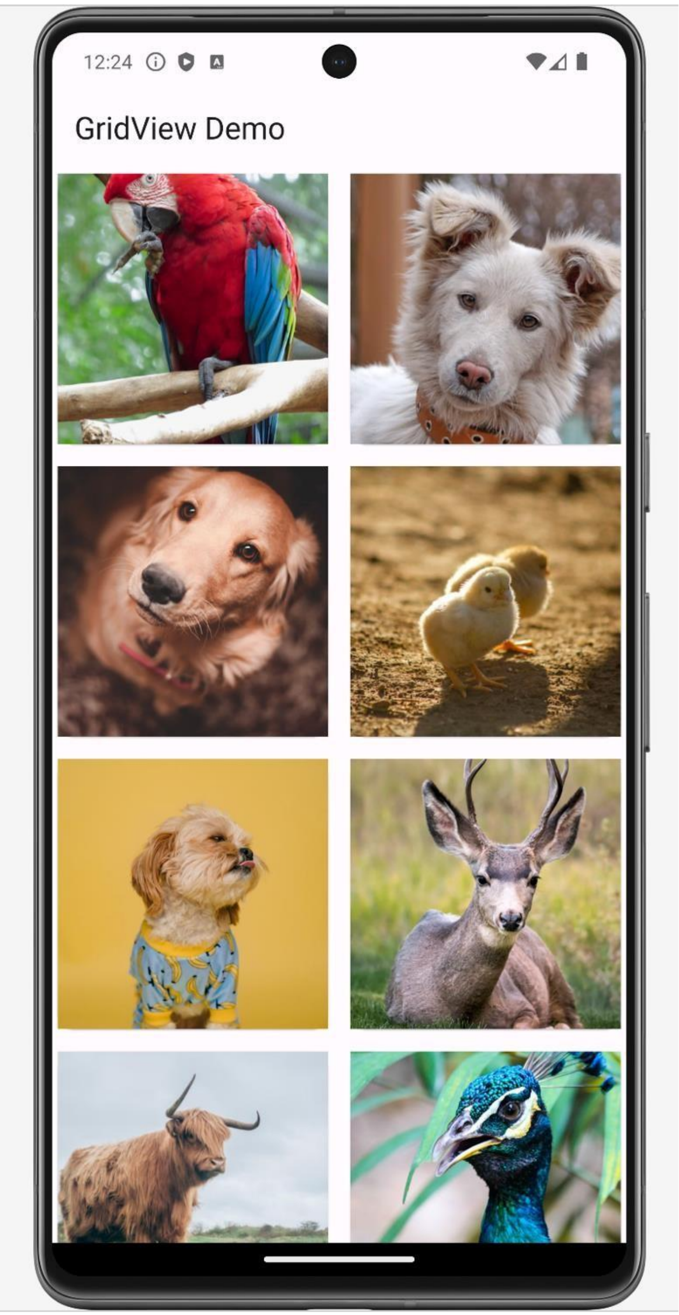
main.dart

```
import 'package:flutter/material.dart';
import 'grid_view.dart'; // Import the MyGridView class from the separate
file  void main() {    runApp(MyApp());
}
class MyApp extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
return MaterialApp(       title:
'GridView Demo',
      debugShowCheckedModeBanner: false, // Remove the debug banner
theme: ThemeData(
        primarySwatch: Colors.blue, // Use Colors for primarySwatch
      ),
      home: MyGridView(), // Use MyGridView as the home widget
    );
  } }
```

grid_view.dart

```
import 'package:flutter/material.dart';
 class MyGridView extends StatelessWidget
{   final List<String> imageUrls = [
    "https://images.unsplash.com/photo-1604457407295-
8aa34e462dcf?w=500&auto=format&fit=crop&q=60&ixlib=rb-
4.0.3&ixid=M3wxMjA3fDB8MHxleHBsb3JlLWZlZWR8OTh8fHxlbnwwfHx8fHw%3D",
"https://images.unsplash.com/photo-
1587027077233c7a2e15825cf?w=500&auto=format&fit=crop&q=60&ixlib=rb-
4.0.3&ixid=M3wxMjA3fDB8MHxleHBsb3JlLWZlZWR8MTAxfHx8ZW58MHx8fHx8",
    "https://images.unsplash.com/photo-1526489550178-
7bd5d9944f4f?w=500&auto=format&fit=crop&q=60&ixlib=rb-
4.0.3&ixid=M3wxMjA3fDB8MHxleHBsb3JlLWZlZWR8MTcyfHx8ZW58MHx8fHx8",
    "https://images.unsplash.com/photo-1546272989-
40c92939c6c2?w=500&auto=format&fit=crop&q=60&ixlib=rb-
4.0.3&ixid=M3wxMjA3fDB8MHxleHBsb3JlLWZlZWR8Njd8fHxlbnwwfHx8fHw%3D",
    "https://images.unsplash.com/photo-1583511655802-
41f2ccc2cc8f?w=500&auto=format&fit=crop&q=60&ixlib=rb-
4.0.3&ixid=M3wxMjA3fDB8MHxleHBsb3JlLWZlZWR8NDV8fHxlbnwwfHx8fHw%3D",
"https://images.unsplash.com/photo-
1476922027627aa7293e3aaa8?w=500&auto=format&fit=crop&q=60&ixlib=rb-
4.0.3&ixid=M3wxMjA3fDB8MHxleHBsb3JlLWZlZWR8NDN8fHxlbnwwfHx8fHw%3D",
    "https://images.unsplash.com/photo-
14706880900676d429c0b2600?w=500&auto=format&fit=crop&q=60&ixlib=rb-
4.0.3&ixid=M3wxMjA3fDB8MHxleHBsb3JlLWZlZWR8NDB8fHxlbnwwfHx8fHw%3D",
"https://images.unsplash.com/photo-
1594031633878c59f0c8c16fd?w=500&auto=format&fit=crop&q=60&ixlib=rb-
4.0.3&ixid=M3wxMjA3fDB8MHxleHBsb3JlLWZlZWR8Mzh8fHxlbnwwfHx8fHw%3D",
"https://images.unsplash.com/photo-
1559214369a6b1d7919865?w=500&auto=format&fit=crop&q=60&ixlib=rb-
4.0.3&ixid=M3wxMjA3fDB8MHxleHBsb3JlLWZlZWR8MTA5fHx8ZW58MHx8fHx8",
    "https://images.unsplash.com/photo-1578326626553-
39f72c545b07?w=500&auto=format&fit=crop&q=60&ixlib=rb-
4.0.3&ixid=M3wxMjA3fDB8MHxleHBsb3JlLWZlZWR8Mjl8fHxlbnwwfHx8fHw%3D",
    // Add more image URLs as needed
  ];

  @override
  Widget build(BuildContext context) {
return Scaffold(      appBar: AppBar(
        title: Text('GridView Demo'),
      ),
      body: GridView.builder(
        gridDelegate: SliverGridDelegateWithFixedCrossAxisCount(
crossAxisCount: 2,           crossAxisSpacing: 8.0,
mainAxisSpacing: 8.0,
        ),
        itemCount: imageUrls.length,
itemBuilder: (context, index) {
return GestureDetector(
onTap: () {
              // Add your onTap logic here
              print('Tapped on image at index $index');
            },             child:
Card(                child:
Image.network(
imageUrls[index],
fit: BoxFit.cover,
              ),
            ),
          );
        },
      ),
    );
  } }
```

OUTPUT:



PRACTICAL 10

AIM: Create an application in Flutter with CRUD operations using SQLite

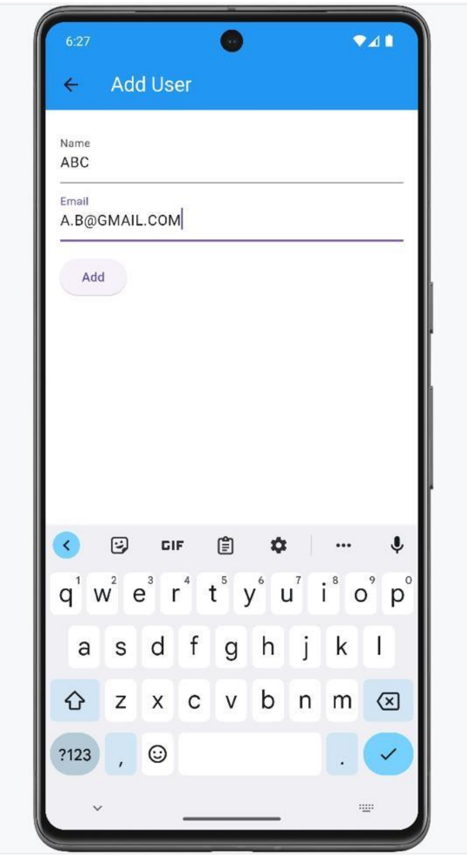
SOLUTION:

main.dart

```
import 'package:flutter/material.dart';
import 'package:sqflite/sqflite.dart'; import
'package:path/path.dart';
 void main() {
runApp(MyApp());
}
class MyApp extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
return MaterialApp(       title: 'SQL
CRUD',
      debugShowCheckedModeBanner: false, // Remove debug banner
theme: ThemeData(         primaryColor: Colors.blue,
scaffoldBackgroundColor: Colors.white,         appBarTheme:
AppBarTheme(           color: Colors.blue,
        ),
        floatingActionButtonTheme: FloatingActionButtonThemeData(
backgroundColor: Colors.blue,
        ),
),
      home: HomePage(),
    );
  }
}  class HomePage extends
StatefulWidget {
  @override
  _HomePageState createState() => _HomePageState();
}  class _HomePageState extends
State<HomePage> {   late Database _database;

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

  Future<void> _initDatabase() async {
    _database = await openDatabase(
      join(await getDatabasesPath(), 'user_database.db'),
onCreate: (db, version) {         return db.execute(
          'CREATE TABLE users(id INTEGER PRIMARY KEY, name TEXT, email
TEXT)',
        );
},       version:
1,
    );
  }

  Future<List<Map<String, dynamic>>> _getUsers() async {
return await _database.query('users');
```

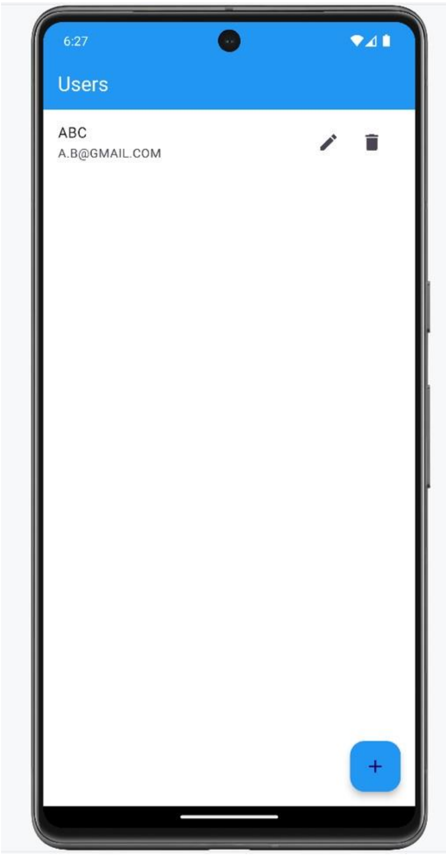
```
  }

  Future<void> _deleteUser(int id) async {
await _database.delete(
      'users',
where: 'id = ?',
whereArgs: [id],
    );
    setState(() {});
  }

  @override
  Widget build(BuildContext context) {
return Scaffold(       appBar: AppBar(
title: Text(           'Users',
          style: TextStyle(color: Colors.white),
        ),
),
      body: FutureBuilder(
future: _getUsers(),         builder:
(context, snapshot) {
          if (snapshot.connectionState == ConnectionState.waiting) {
return Center(
              child: CircularProgressIndicator(),
            );
          }

          if (snapshot.hasData && (snapshot.data as List).isNotEmpty) {
final users = snapshot.data as List<Map<String, dynamic>>?;

            return ListView.builder(
itemCount: users!.length,
itemBuilder: (context, index) {
final user = users[index];
return ListTile(                   title:
Text(user['name']),                   subtitle:
Text(user['email']),                   trailing:
Row(
                    mainAxisSize: MainAxisSize.min,
                    children: [                       IconButton(
icon: Icon(Icons.edit),                         onPressed: () {
Navigator.push(                             context,
MaterialPageRoute(                               builder: (context) =>
EditUserPage(user:
user),
                            ),
                          ).then((_) {
setState(() {});
                          });
                        },
                      ),
                      IconButton(
                        icon: Icon(Icons.delete),
onPressed: () {
                          _deleteUser(user['id']);
                        },
                      ),
                    ],
                  ),
                );
              },
            );            }
else {              return
Center(
              child: Text('No users found'),
            );
          }
        },
),
      floatingActionButton: FloatingActionButton(
        onPressed: () {
Navigator.push(
context,
            MaterialPageRoute(
              builder: (context) => AddUserPage(),
            ),
).then((_) {
setState(() {});
          });
},
        child: Icon(Icons.add),
      ),
    );
  }
}
class AddUserPage extends StatelessWidget {
  final TextEditingController _nameController = TextEditingController();
final TextEditingController _emailController = TextEditingController();
  @override
  Widget build(BuildContext context) {
return Scaffold(       appBar: AppBar(
title: Text(           'Add User',
          style: TextStyle(color: Colors.white),
        ),       ),
body: Padding(
        padding: const EdgeInsets.all(16.0),
child: Column(
          crossAxisAlignment: CrossAxisAlignment.start,
children: [
            TextField(
              controller: _nameController,
decoration: InputDecoration(
labelText: 'Name',
              ),
            ),
TextField(
              controller: _emailController,
decoration: InputDecoration(
labelText: 'Email',
              ),
            ),
            SizedBox(height: 16.0),
            ElevatedButton(
onPressed: () {
                _addUser(context);
              },
              child: Text('Add'),
            ),
          ],
        ),
      ),
    );
  }    void _addUser(BuildContext context)
async {     final name =
_nameController.text.trim();     final email =
_emailController.text.trim();
     if (name.isNotEmpty && email.isNotEmpty)
{       final database = await openDatabase(
        join(await getDatabasesPath(), 'user_database.db'),
version: 1,
      );
      await database.transaction((txn) async {
await txn.rawInsert(
          'INSERT INTO users(name, email) VALUES(?, ?)',
          [name, email],
        );
      });

      Navigator.pop(context);
    }
  }
}
class EditUserPage extends StatelessWidget {
final Map<String, dynamic> user;
  final TextEditingController _nameController = TextEditingController();
final TextEditingController _emailController = TextEditingController();
  EditUserPage({required this.user}) {
    _nameController.text = user['name'];
    _emailController.text = user['email'];
  }

  @override
  Widget build(BuildContext context) {
    return Scaffold(
appBar: AppBar(
title: Text(
'Edit User',
          style: TextStyle(color: Colors.white),
        ),       ),
body: Padding(
        padding: const EdgeInsets.all(16.0),
child: Column(
          crossAxisAlignment: CrossAxisAlignment.start,
          children: [
TextField(
              controller: _nameController,
decoration: InputDecoration(
labelText: 'Name',
              ),
            ),
TextField(
              controller: _emailController,
decoration: InputDecoration(
labelText: 'Email',
              ),
            ),
            SizedBox(height: 16.0),
            ElevatedButton(
onPressed: () {
                _updateUser(context);
              },
              child: Text('Update'),
            ),
          ],
        ),
      ),
    );
  }    void _updateUser(BuildContext context)
async {     final name =
_nameController.text.trim();     final email =
_emailController.text.trim();

    if (name.isNotEmpty && email.isNotEmpty) {       final
database = await openDatabase(         join(await
getDatabasesPath(), 'user_database.db'),         version:
1,
      );
      await database.update(
        'users',
        {'name': name, 'email': email},
where: 'id = ?',         whereArgs:
[user['id']],
      );

      Navigator.pop(context);
    }
  }
}
```

OUTPUT:

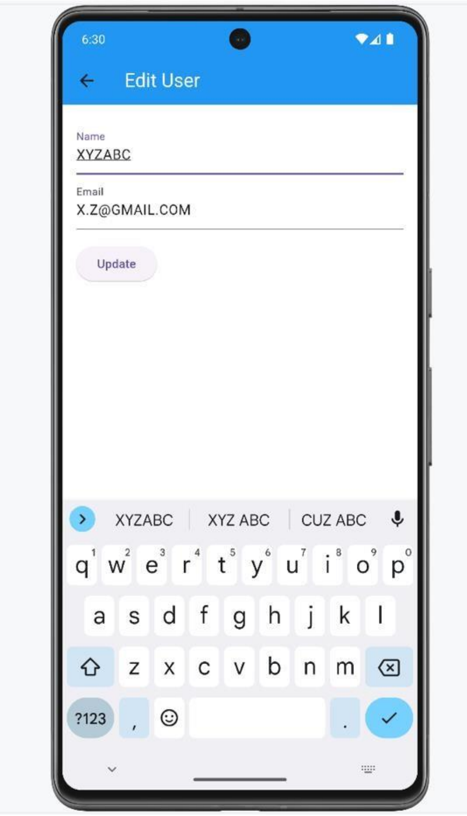


Name
ABC

Email
A.B@GMAIL.COM

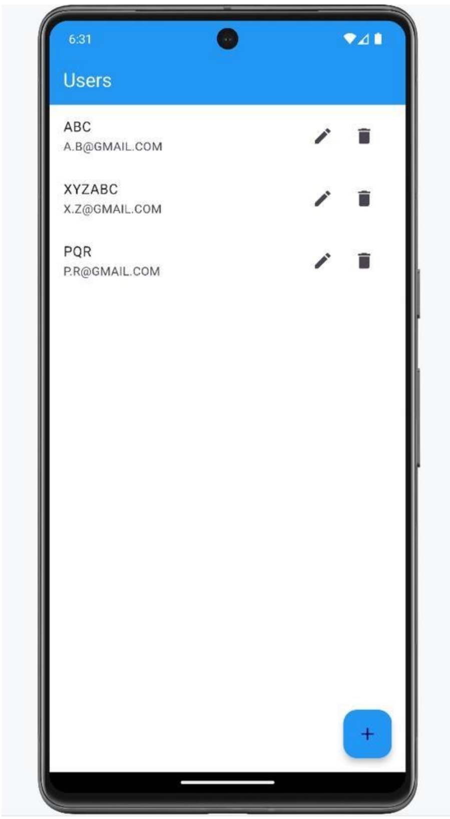


ABC
A.B@GMAIL.COM



Name
XYZABC

Email
X.Z@GMAIL.COM



ABC
A.B@GMAIL.COM

XYZABC
X.Z@GMAIL.COM

PQR
P.R@GMAIL.COM

PRACTICAL 11

AIM: Create an application in Flutter that connects to REST API

SOLUTION:

main.dart

```
import 'package:flutter/material.dart'; import
'user_list_screen.dart';
 void main() {
runApp(MyApp());
}
class MyApp extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
return MaterialApp(       title: 'REST
API',       theme: ThemeData(
        primarySwatch: Colors.blue,
      ),
      debugShowCheckedModeBanner: false, // Remove debug banner
home: UserListScreen(),
    );
  } }
```

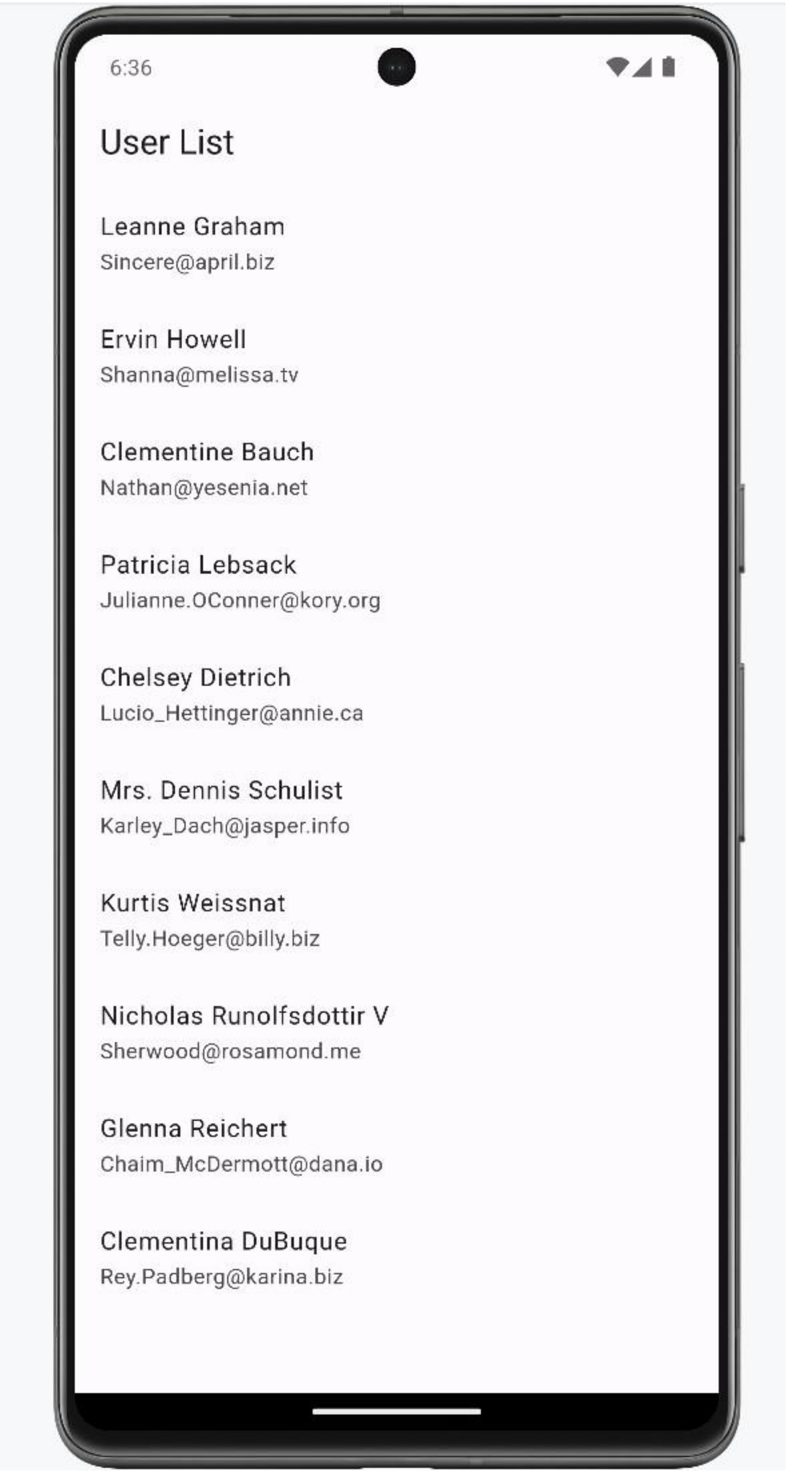
user_screnn_list.dart

```
import 'package:flutter/material.dart'; import
'dart:convert';
import 'package:http/http.dart' as http; import
'user.dart';
 class UserListScreen extends StatelessWidget
{
  @override
  Widget build(BuildContext context) {
return Scaffold(       appBar: AppBar(
        title: Text('User List'),
      ),
      body: FutureBuilder<List<User>>(
future: fetchUsers(),         builder:
(context, snapshot) {
          if (snapshot.connectionState == ConnectionState.waiting) {
return Center(
              child: CircularProgressIndicator(),
            );
          } else if (snapshot.hasError) {
return Center(
              child: Text('Error: ${snapshot.error}'),
            );
} else {
            final users = snapshot.data!;
            return ListView.builder(
itemCount: users.length,
itemBuilder: (context, index) {
final user = users[index];
return ListTile(                   title:
Text(user.name),                   subtitle:
Text(user.email),
                ); 
              },
            );
          }
        },
      ),
    );
  }
}

Future<List<User>> fetchUsers() async {
final response =       await
http.get(Uri.parse('https://jsonplaceholder.typicode.com/users'));

  if (response.statusCode == 200) {
    final List<dynamic> data = jsonDecode(response.body);
return data.map((json) => User.fromJson(json)).toList();
  } else {
    throw Exception('Failed to Load Users');
  } }
```

user.dart

```
class User {   final
int id;   final
String name;   final
String email;

  User({required this.id, required this.name, required this.email});
   factory User.fromJson(Map<String, dynamic> json)
{     return User(      id: json['id'],       name:
json['name'],       email: json['email'],
    );
  } }
```

OUTPUT:



6:36

User List

Leanne Graham
Sincere@april.biz

Ervin Howell
Shanna@melissa.tv

Clementine Bauch
Nathan@yesenia.net

Patricia Lebsack
Julianne.OConner@kory.org

Chelsey Dietrich
Lucio_Hettinger@annie.ca

Mrs. Dennis Schulist
Karley_Dach@jasper.info

Kurtis Weissnat
Telly.Hoeger@billy.biz

Nicholas Runolfsdottir V
Sherwood@rosamond.me

Glenna Reichert
Chaim_McDermott@dana.io

Clementina DuBuque
Rey.Padberg@karina.biz

PRACTICAL 12

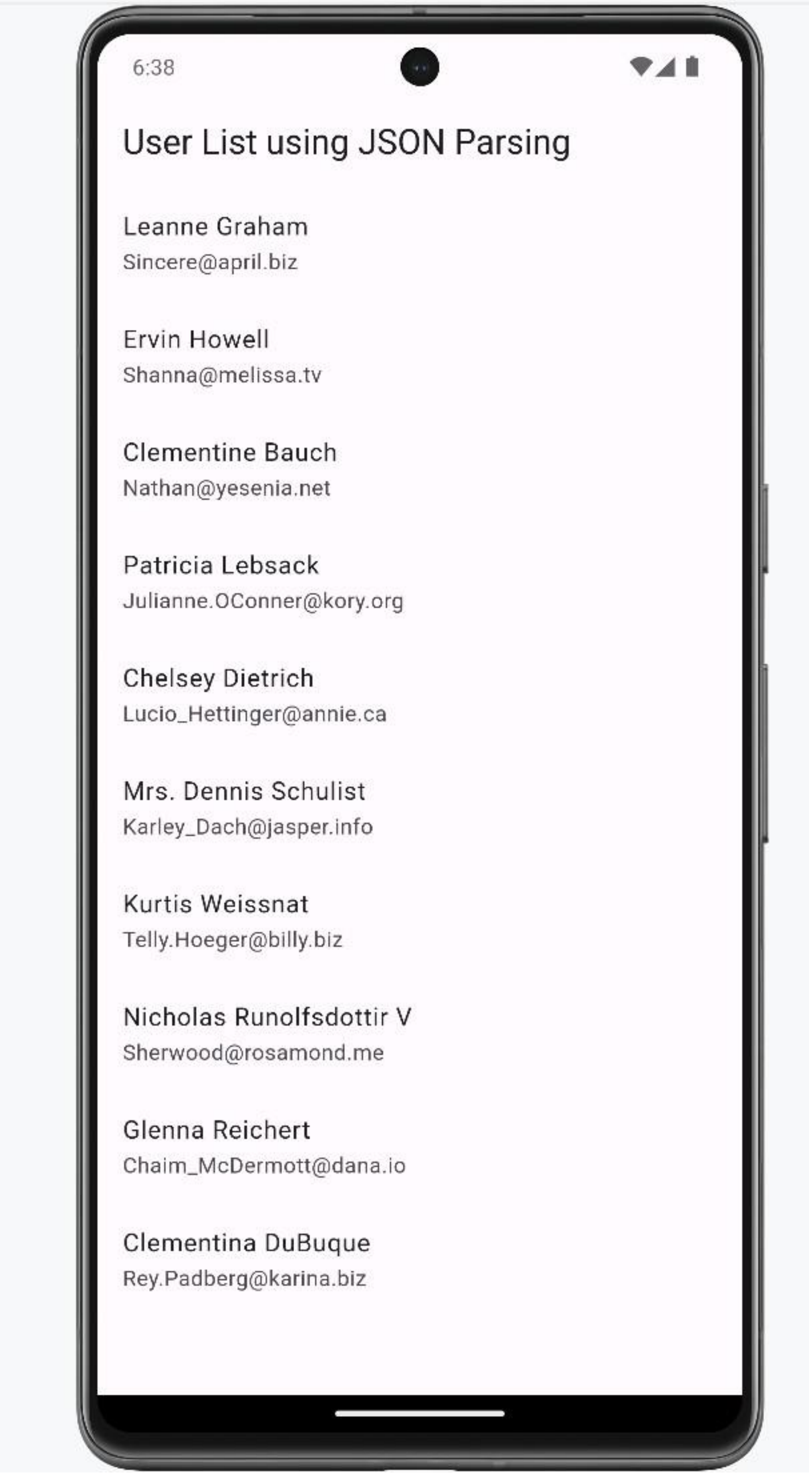
AIM: Create an application in Flutter that parses data from JSON to REST API

SOLUTION:

main.dart

```
import 'package:flutter/material.dart'; import
'dart:convert';
import 'package:http/http.dart' as http; import
'user.dart';
 void main() {
runApp(MyApp());
}
class MyApp extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
return MaterialApp(       title: 'REST
API',        theme: ThemeData(
        primarySwatch: Colors.blue,
      ),
      home: MainScreen(),
      debugShowCheckedModeBanner: false, // This line removes the debug
banner     );
  }
}
class MainScreen extends StatefulWidget {
  @override
  _MainScreenState createState() => _MainScreenState();
}  class _MainScreenState extends
State<MainScreen> {   late Future<List<User>>
_futureUsers;

  @override   void initState() {
super.initState();
_futureUsers = fetchUsers();
  }

  Future<List<User>> fetchUsers() async {
final response =         await
http.get(Uri.parse('https://jsonplaceholder.typicode.com/users'));

    if (response.statusCode == 200) {
      final List<dynamic> data = jsonDecode(response.body);
return data.map((json) => User.fromJson(json)).toList();
    } else {
      throw Exception('Failed to Load Users');
}
  }

  @override
  Widget build(BuildContext context) {
return Scaffold(       appBar: AppBar(
        title: Text('User List using JSON Parsing'),
      ),
      body: FutureBuilder<List<User>>(
future: _futureUsers,         builder:
(context, snapshot) {
          if (snapshot.connectionState == ConnectionState.waiting) {
return Center(
              child: CircularProgressIndicator(),
            );
          } else if (snapshot.hasError) {
return Center(
              child: Text('Error: ${snapshot.error}'),
            );
} else {
            final users = snapshot.data!;
return ListView.builder(
itemCount: users.length,
itemBuilder: (context, index) {
final user = users[index];
return ListTile(                   title:
Text(user.name),                   subtitle:
Text(user.email),
                );
              },
            );
          }
        },
      ),
    );
  } }
```

OUTPUT:



User List using JSON Parsing

Leanne Graham
Sincere@april.biz

Ervin Howell
Shanna@melissa.tv

Clementine Bauch
Nathan@yesenia.net

Patricia Lebsack
Julianne.OConner@kory.org

Chelsey Dietrich
Lucio_Hettinger@annie.ca

Mrs. Dennis Schulist
Karley_Dach@jasper.info

Kurtis Weissnat
Telly.Hoeger@billy.biz

Nicholas Runolfsdottir V
Sherwood@rosamond.me

Glenna Reichert
Chaim_McDermott@dana.io

Clementina DuBuque
Rey.Padberg@karina.biz

PRACTICAL 13

AIM: Create an application in Flutter that uses Hardware Acceleration

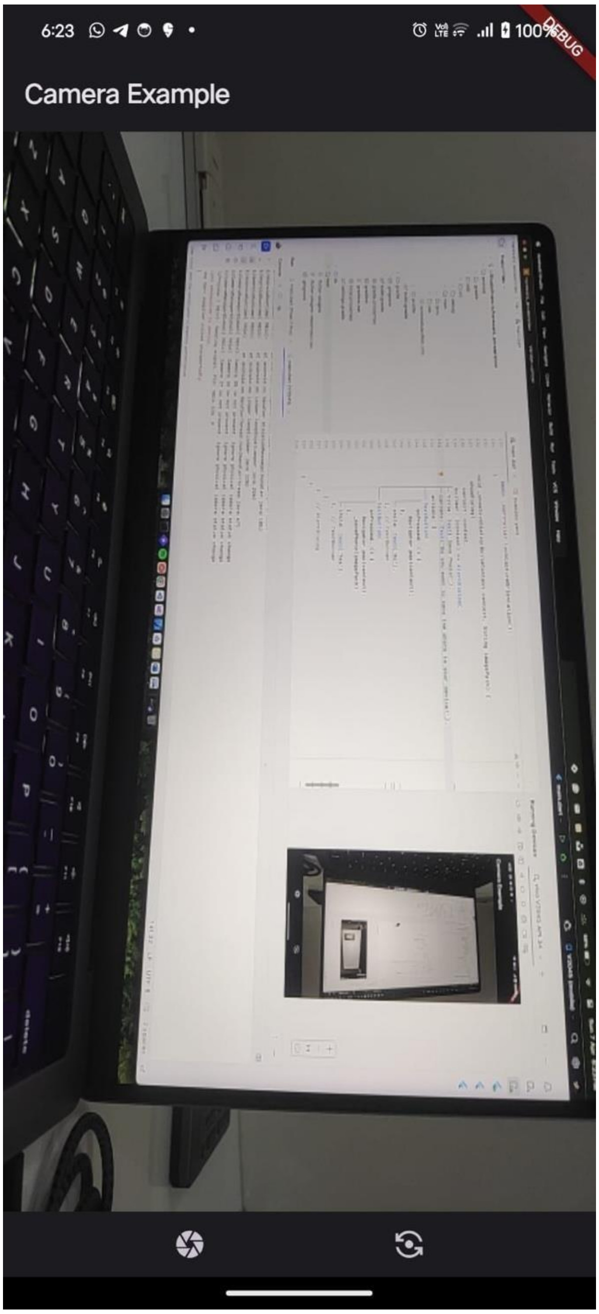
SOLUTION:

main.dart

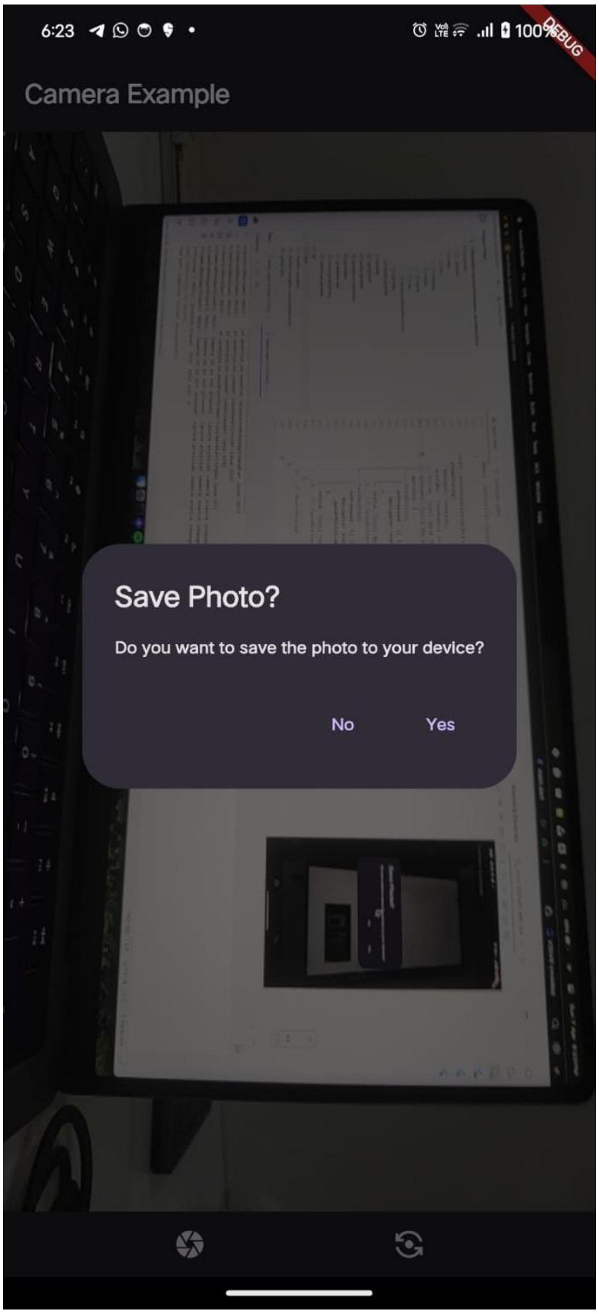
```
// main.dart
 import
'package:flutter/material.dart'; import
'package:camera/camera.dart';

void main() async {
  WidgetsFlutterBinding.ensureInitialized();
final cameras = await availableCameras();   final
firstCamera = cameras.first;   runApp(CameraApp(
      camera: firstCamera, cameras: cameras)); // Pass cameras to CameraApp
}  class CameraApp extends
StatelessWidget {   final
CameraDescription camera;
  final List<CameraDescription> cameras; // Add cameras property
   const CameraApp({
Key? key,      required
this.camera,
    required this.cameras, // Initialize cameras property
  }) : super(key: key);

  @override
  Widget build(BuildContext context) {
return MaterialApp(       theme:
ThemeData.dark(),       home:
CameraScreen(
          camera: camera, cameras: cameras), // Pass cameras to
CameraScreen
    );
  }
}  class CameraScreen extends
StatefulWidget {   final CameraDescription
camera;
  final List<CameraDescription> cameras; // Add cameras property
   const CameraScreen({
Key? key,      required
this.camera,
    required this.cameras, // Initialize cameras property
  }) : super(key: key);

  @override
  _CameraScreenState createState() => _CameraScreenState();
}
class _CameraScreenState extends State<CameraScreen> {
  late CameraController _controller;   late
Future<void> _initializeControllerFuture;   bool
_isFrontCamera = false;
```

```
  @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: FutureBuilder<void>(         future:
_initializeControllerFuture,         builder: (context,
snapshot) {
          if (snapshot.connectionState == ConnectionState.done) {
return Column(               children: [
Expanded(
                  child: CameraPreview(_controller),
                ),
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.flip_camera_android),
onPressed: () {
                        _toggleCamera();
                      },
                    ),
                  ],
                ),
              ],
            );
          } else {
            return const Center(child: CircularProgressIndicator());
          }
        },
      ),
    );
  }
  void _toggleCamera() async {
    final CameraDescription newCamera = _isFrontCamera
        ? widget.cameras.firstWhere(
            (camera) => camera.lensDirection == CameraLensDirection.back)
        : widget.cameras.firstWhere(
            (camera) => camera.lensDirection == CameraLensDirection.front);
if (!_controller.value.isInitialized) {       return;     }
    await _controller.dispose();
    _controller = CameraController(
newCamera,
      ResolutionPreset.medium,
    );
    setState(() {
      _isFrontCamera = !_isFrontCamera;
    });
    await _controller.initialize();
    await _controller.lockCaptureOrientation();
  }
  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');
  } }
```

OUTPUT:



Camera Example



Camera Example

Save Photo?

Do you want to save the photo to your device?

No Yes