



Index

Name, Ayush Kumar
 Subject, ADF
 Semester, 6th

Year, 2025
 Class, EII - A
 Roll No, 92201703008

No.	Date	Experiment Description / Subject	Page	Marks	Faculty's Signature
01	19/1	Practical - 1	1-10	20	JL
02	21/1	Practical - 2	8	20	JL
03	28/1	Practical - 3	9-12	20	JL
04	11/2	Practical - 4	13-14	20	JL
05	18/2	Practical - 5	15-16	20	JL
06	25/2	Practical - 6	17-19	20	JL
07	11/3	Practical - 7	20-30	20	JL
08	20/3	Practical - 8	31-32	20	JL
09	27/3	Practical - 9	33	20	JL
10	11/4	Practical - 10	34-35	20	JL
11	8/4	Practical - 11	40-42	20	JL
12	15/4	Practical - 12	43-46	20	JL
13	22/4	Practical - 13	47-49	20	JL

Practical 1

Aim: 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:**

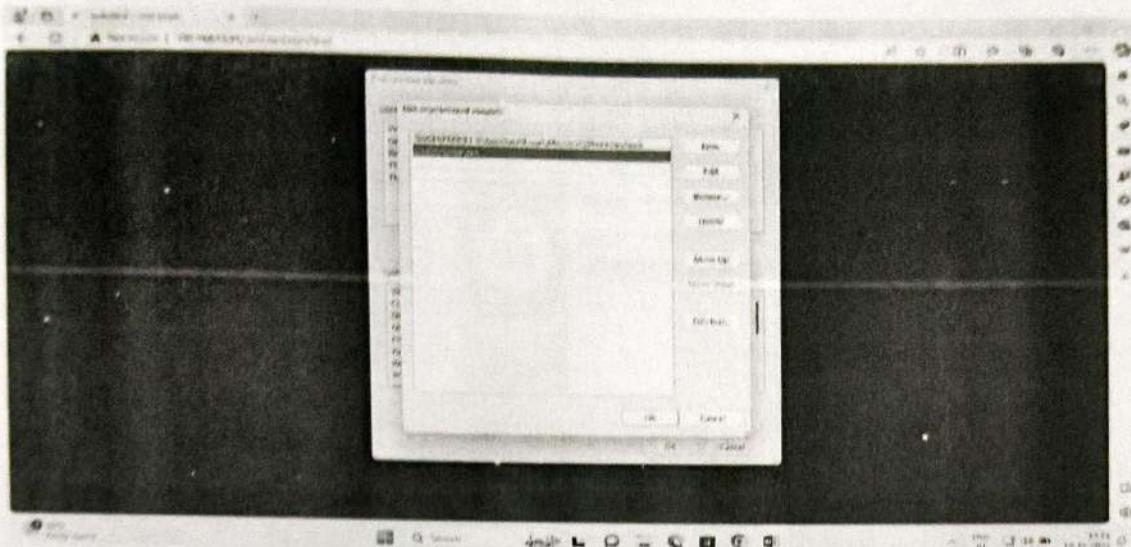
- Visit Flutter Website for Installation of Flutter -> <https://docs.flutter.dev/get-started/install>.

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.



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 command-line tools, etc.

Step 2: Installing Android Studio.

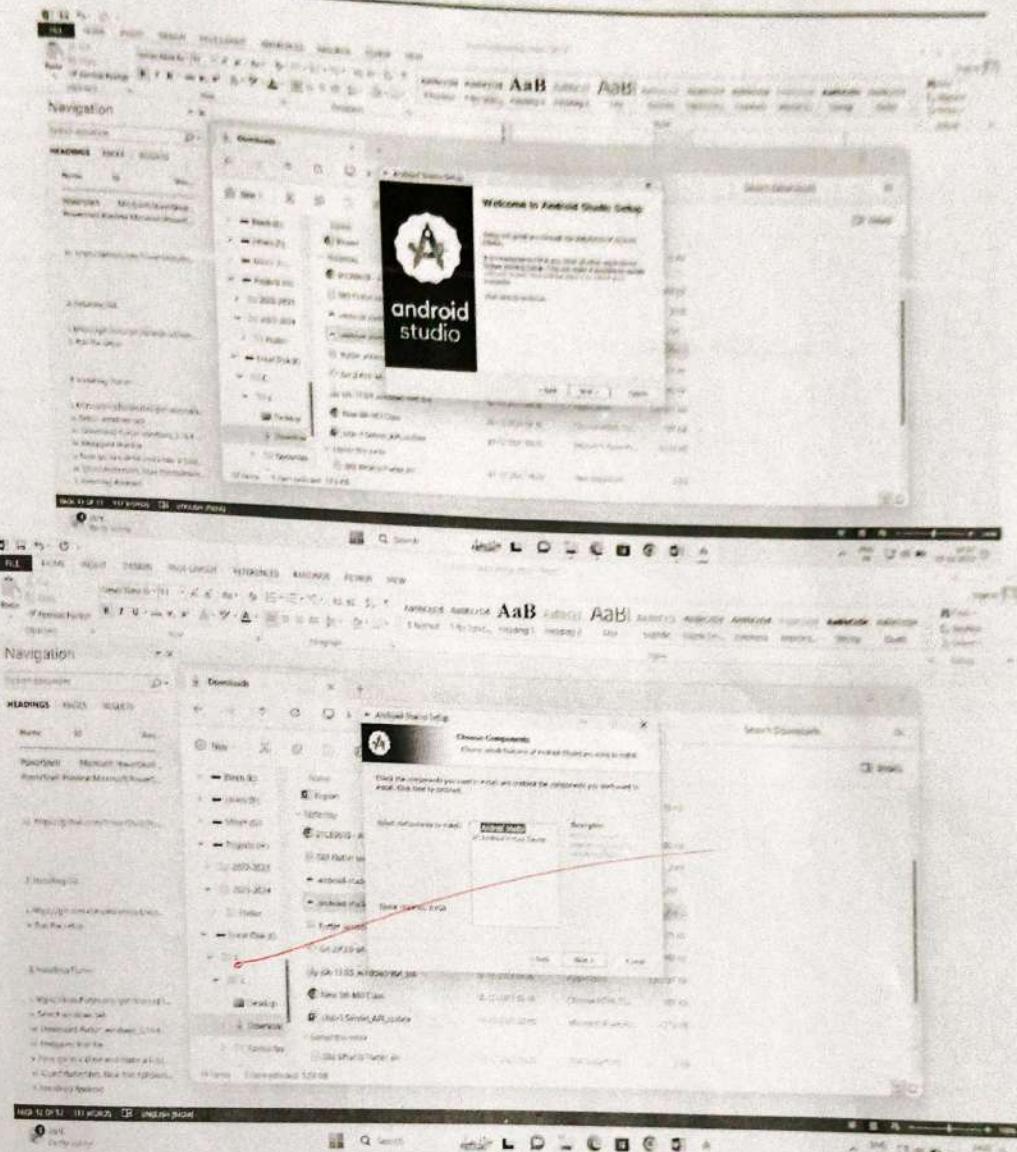
i. **Download Android Studio:**

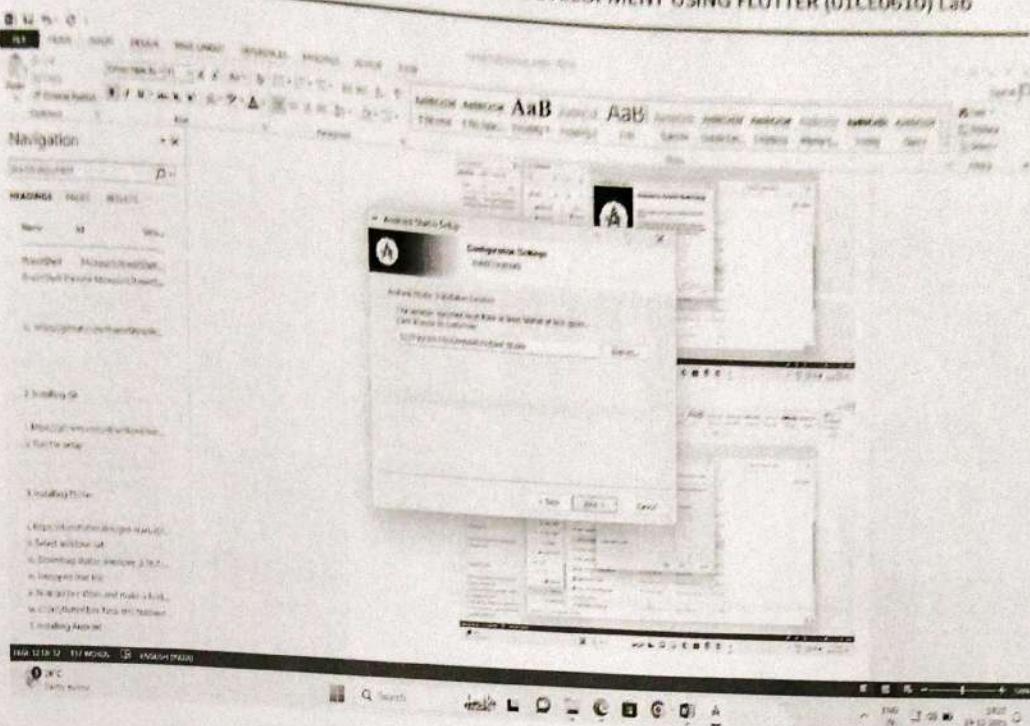
- Visit the [Android Studio download page](#).
- Click on the "Download" button and download the Windows version.

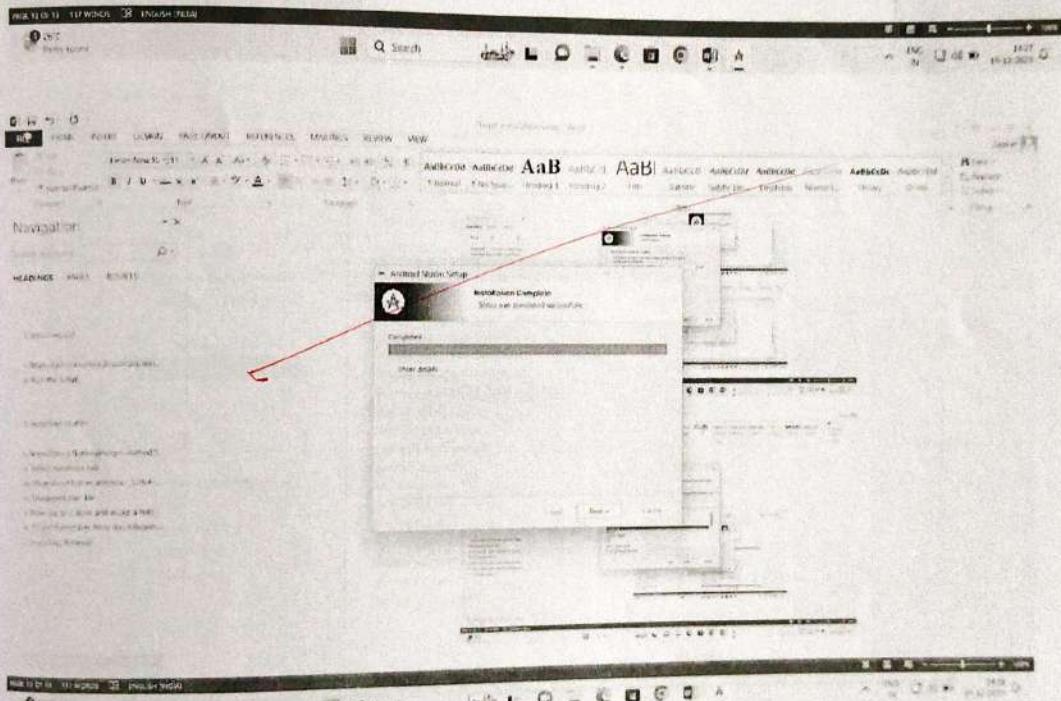
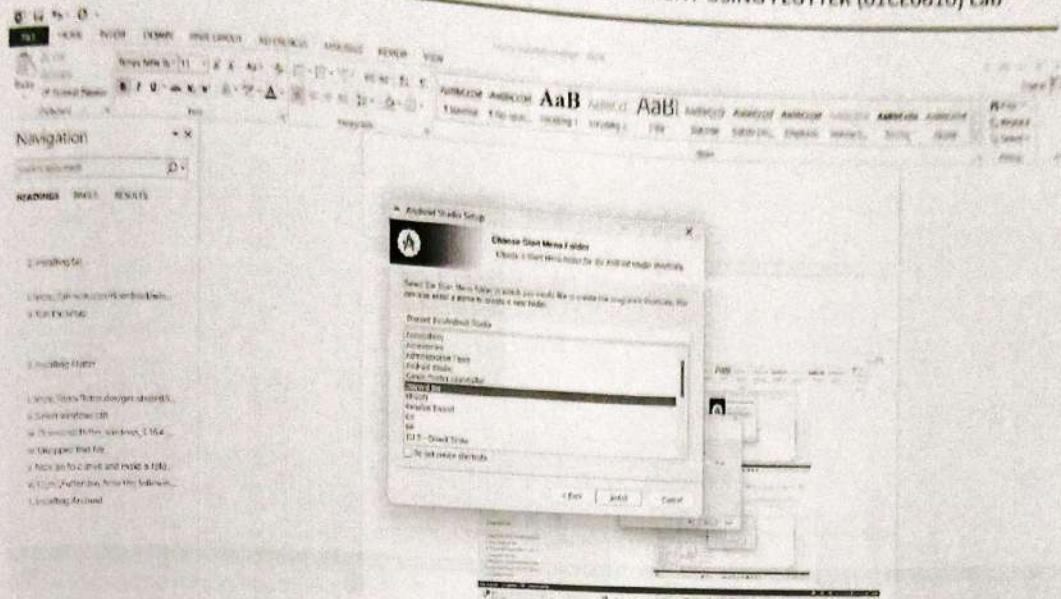
ii. **Run the Installer:**

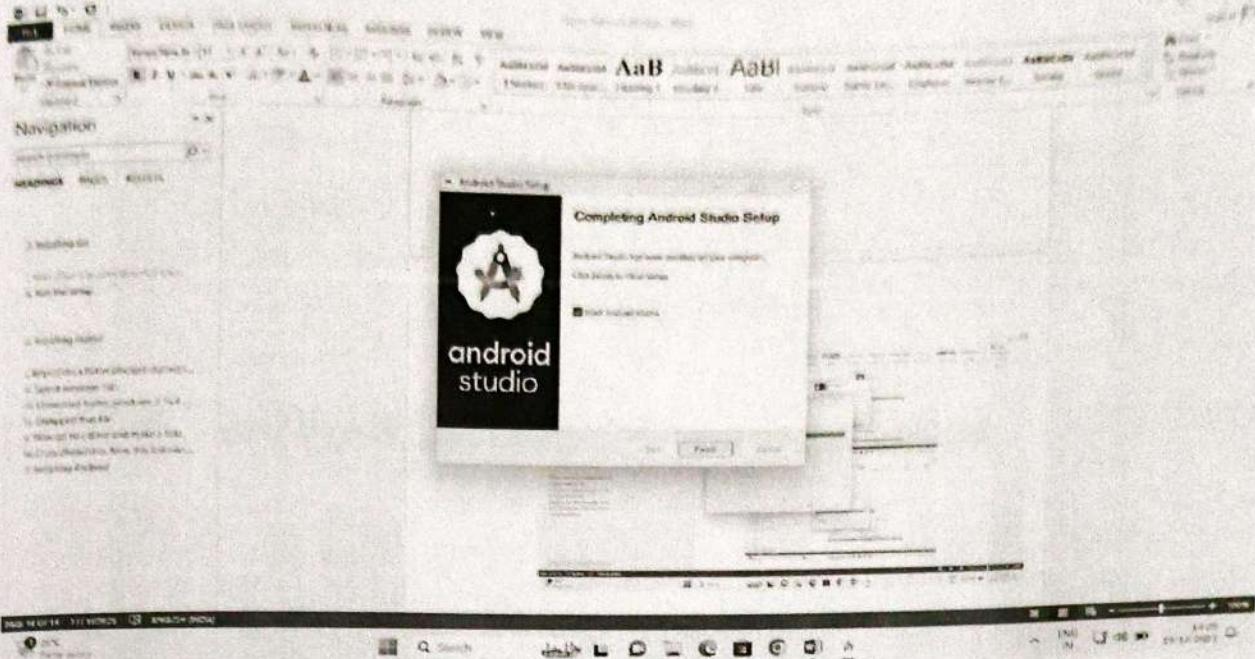
- Once the download is complete, run the installer executable (.exe) file.

iii. **Follow Installation Wizard:**

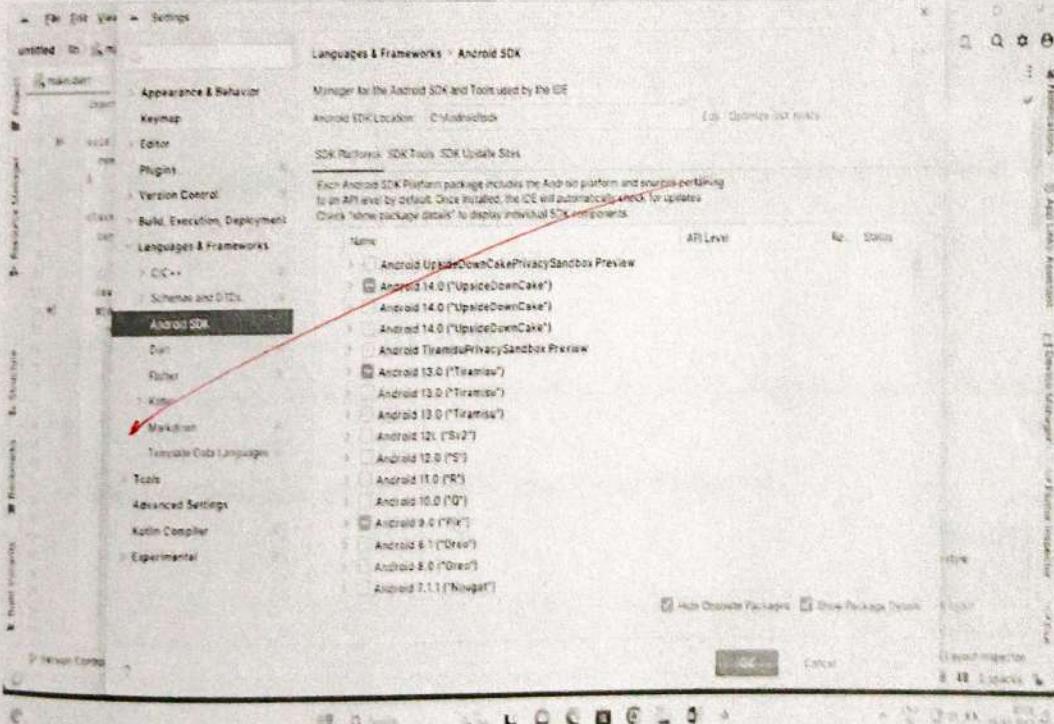




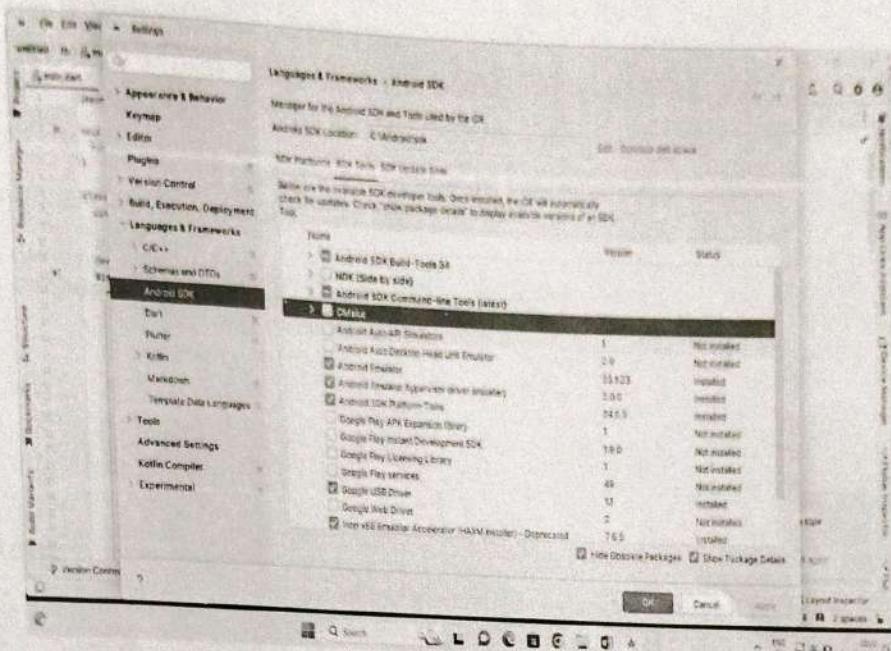




- **Android SDK Platforms:**



Android SDK Tools:



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

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

JL

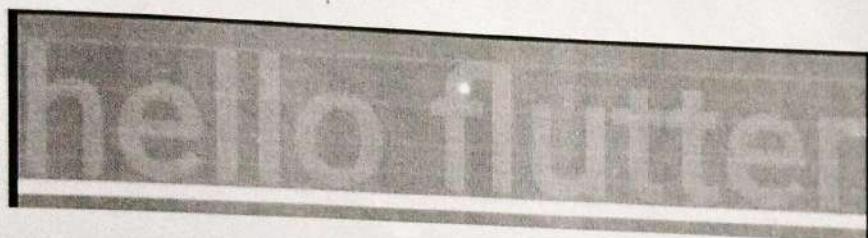
Practical 2

Aim: Create a "Hello Flutter" application.

Solution:

```
import 'package:flutter/material.dart'; void main(){  
runApp(MaterialApp(  
home: Text("hello flutter",style: TextStyle( color: Colors.green  
,fontSize: 85,backgroundColor: Colors.blueAccent))  
)); }
```

Output:





Practical 3

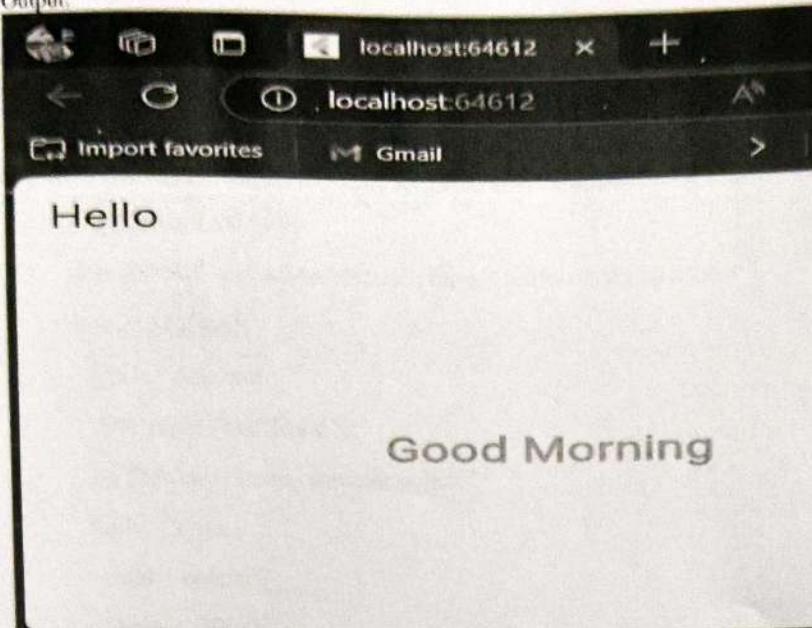
Aim: Create and application using Flutter Key Widgets

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

```
void main() {  
  runApp(const MyApp());  
}  
  
class MyApp extends StatelessWidget {  
  const MyApp({super.key});  
  @override  
  Widget build(BuildContext context) {  
    return MaterialApp(  
      debugShowCheckedModeBanner: false, // Removes debug banner  
      home: Scaffold(  
        appBar: AppBar(  
          title: const Text('Hello'),  
        ),  
        body: const Center(  
          child: Text(  
            'Good Morning',  
            style: TextStyle(  
              fontSize: 24.0,  
              fontWeight: FontWeight.bold,  
              color: Colors.blue,  
            ),  
            overflow: TextOverflow.ellipsis,  
          ),  
        ),  
      ),  
    );  
  }  
}
```



Output:



JK

Code:

Practical 3 – B

```
import 'package:flutter/material.dart';
void main() {
  runApp(MaterialApp(
    debugShowCheckedModeBanner: false, // Removes debug banner
    home: Scaffold(
      appBar: AppBar(
        title: const Text("Rohit"),
      ), // **Added missing parenthesis here**
      body: Center(
        child: Container(
          height: 200.0,
          width: 150.0,
          margin: const EdgeInsets.all(16.0),
          padding: const EdgeInsets.symmetric(horizontal: 40.0, vertical: 60.0),
          decoration: BoxDecoration(
            color: Colors.blue,
            borderRadius: BorderRadius.circular(20.0), // **Added rounded corners**
          ),
          child: const Center(
            child: Text(
              "Hitman",
              style: TextStyle(color: Colors.white, fontSize: 18.0),
              textAlign: TextAlign.center,
            ),
          ),
        ),
      ),
    ),
  );
}
```



Output:



JL

Practical 4

Aim: Create and application using Flutter Key Widgets.

Code:

```
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('Row, Column, RichText & Icons Example'),
        ),
        body: Center(
          child: Column(
            mainAxisAlignment: MainAxisAlignment.center,
            children: <Widget>[
              // Example of a Row with Icons
              Row(
                mainAxisAlignment: MainAxisAlignment.center,
                children: <Widget>[
                  Icon(Icons.star, color: Colors.yellow, size: 40),
                  SizedBox(width: 10), // Spacer
                  Icon(Icons.favorite, color: Colors.red, size: 40),
                  SizedBox(width: 10), // Spacer
                  Icon(Icons.thumb_up, color: Colors.blue, size: 40),
                ],
              ),
              SizedBox(height: 20), // Spacer
              // Example of a Column with Icons
              Column(
                mainAxisAlignment: MainAxisAlignment.center,
                children: <Widget>[
                  Icon(Icons.home, color: Colors.green, size: 40),
                  SizedBox(height: 10), // Spacer
                  Icon(Icons.settings, color: Colors.orange, size: 40),
                  SizedBox(height: 10), // Spacer
                  Icon(Icons.person, color: Colors.purple, size: 40),
                ],
              ),
            ],
          ),
        ),
      ),
    );
  }
}
```

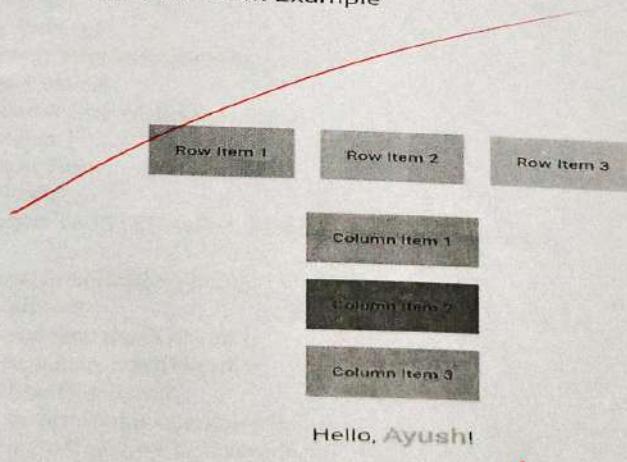


SizedBox(height: 20), // Spacer

```
// Example of RichText with Icons
RichText(
  text: TextSpan(
    style: DefaultTextStyle.of(context).style,
    children: <TextSpan>[
      TextSpan(
        text: 'Hello, ',
        style: TextStyle(color: Colors.black, fontSize: 20),
      ),
      WidgetSpan(
        child: Icon(Icons.flutter_dash, color: Colors.blue, size: 24),
      ),
      TextSpan(
        text: ' Ayush',
        style: TextStyle(color: Colors.black, fontSize: 20),
      ),
    ],
  ),
),
),
),
),
),
),
);
}
}
```

Output :

Row, Column & RichText Example





Practical 5

Create and application with Flutter UI Components.

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

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

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

  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      debugShowCheckedModeBanner: false,
      home: LoginScreen(),
    );
  }
}

class LoginScreen extends StatelessWidget {
  final TextEditingController emailController = TextEditingController();
  final TextEditingController passwordController = TextEditingController();

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      body: Center(
        child: Padding(
          padding: EdgeInsets.all(20.0),
          child: Column(
            mainAxisAlignment: MainAxisAlignment.min,
            children: [
              const Text(
                'Login',
                style: TextStyle(fontSize: 28, fontWeight: FontWeight.bold),
              ),
              const SizedBox(height: 20),
              TextField(
                controller: emailController,
                decoration: InputDecoration(
                  labelText: 'Email',
                  border: OutlineInputBorder(),
                  prefixIcon: Icon(Icons.email),
                ),
                keyboardType: TextInputType.emailAddress,
              ),
            ],
          ),
        ),
      ),
    );
  }
}
```

```
const SizedBox(height: 10),  
TextField(  
  controller: passwordController,  
  decoration: InputDecoration(  
    labelText: 'Password',  
    border: OutlineInputBorder(),  
    prefixIcon: Icon(Icons.lock),  
>,  
  obscureText: true,  
>,  
const SizedBox(height: 20),  
ElevatedButton(  
  onPressed: () {  
    String email = emailController.text;  
    String password = passwordController.text;  
    // Add login logic here  
    print('Email: $email, Password: $password');  
>,  
  child: const Text('Login'),  
  style: ElevatedButton.styleFrom(  
    minimumSize: Size(double.infinity, 50),  
>,  
>,  
>,  
>,  
>,  
>);  
>);  
>};  
>}  
>}
```

Output





Practical 6

Create and application with Flutter UI Components.

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

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

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

  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      debugShowCheckedModeBanner: false,
      home: SignUpScreen(),
    );
  }
}

class SignUpScreen extends StatelessWidget {
  final TextEditingController nameController = TextEditingController();
  final TextEditingController emailController = TextEditingController();
  final TextEditingController passwordController = TextEditingController();
  final TextEditingController confirmPasswordController = TextEditingController();

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      body: Center(
        child: Padding(
          padding: const EdgeInsets.all(20.0),
          child: Column(
            mainAxisAlignment: MainAxisAlignment.min,
            children: [
              const Text(
                'Sign Up',
                style: TextStyle(fontSize: 28, fontWeight: FontWeight.bold),
              ),
              const SizedBox(height: 20),
              TextField(
                controller: nameController,
                decoration: const InputDecoration(
                  labelText: 'Full Name',
                  border: OutlineInputBorder(),
                  prefixIcon: Icon(Icons.person),
                ),
              ),
            ],
          ),
        ),
      ),
    );
  }
}
```

```

const SizedBox(height: 10),
TextField(
  controller: emailController,
  decoration: const InputDecoration(
    labelText: 'Email',
    border: OutlineInputBorder(),
    prefixIcon: Icon(Icons.email),
  ),
  keyboardType: TextInputType.emailAddress,
),
const SizedBox(height: 10),
TextField(
  controller: passwordController,
  decoration: const InputDecoration(
    labelText: 'Password',
    border: OutlineInputBorder(),
    prefixIcon: Icon(Icons.lock),
  ),
  obscureText: true,
),
const SizedBox(height: 10),
TextField(
  controller: confirmPasswordController,
  decoration: const InputDecoration(
    labelText: 'Confirm Password',
    border: OutlineInputBorder(),
    prefixIcon: Icon(Icons.lock),
  ),
  obscureText: true,
),
const SizedBox(height: 20),
ElevatedButton(
  onPressed: () {
    String name = nameController.text;
    String email = emailController.text;
    String password = passwordController.text;
    String confirmPassword = confirmPasswordController.text;

    // Add sign-up logic here
    if (password == confirmPassword) {
      print('Name: $name, Email: $email, Password: $password');
    } else {
      print('Passwords do not match');
    }
  },
  child: const Text('Sign Up'),
  style: ElevatedButton.styleFrom(
    minimumSize: const Size(double.infinity, 50),
  ),
).

```

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

Output:

Sign Up

— Full Name —

— Email —

— Password —

— Confirm Password —

Sign Up



JL

Practical 7: Create and application with Navigation in Flutter.

main.dart:

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

void main() {
  runApp(MaterialApp(
    debugShowCheckedModeBanner: false,
    home: Scaffold(
      appBar: AppBar(
        title: const Text("Practical - 7"),
        foregroundColor: Colors.black87,
      ),
      body: LoginPage(),
    ),
  );
}
```

login.dart:

```
import 'package:flutter/material.dart';
import 'package:practical7/custom_gesture.dart';
import 'package:practical7/forpass.dart';
import 'package:practical7/signup.dart';

class LoginPage extends StatelessWidget {

  LoginPage({super.key});

  final TextEditingController usernameController = TextEditingController();
  final TextEditingController passwordController = TextEditingController();

  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      debugShowCheckedModeBanner: false,
      home: Scaffold(
        body: Container(
          margin: const EdgeInsets.all(24),
          child: Column(
            mainAxisAlignment: MainAxisAlignment.spaceEvenly,
            children: [
              header(context),
              inputField(context),
              forgotPassword(context),
              signup(context),
            ],
          ),
        ),
      );
  }
}
```

```
header(context) {
    return const Column(
        children: [
            Text(
                "Welcome Back",
                style: TextStyle(fontSize: 40, fontWeight: FontWeight.bold),
            ),
            Text("Enter your credential to login"),
        ],
    );
}

inputField(context) {
    return Column(
        crossAxisAlignment: CrossAxisAlignment.stretch,
        children: [
            TextFormField(
                controller: usernameController,
                decoration: InputDecoration(
                    hintText: "Username",
                    border: OutlineInputBorder(
                        borderRadius: BorderRadius.circular(18),
                        borderSide: BorderSide.none
                    ),
                    fillColor: Colors.purple.withOpacity(0.1),
                    filled: true,
                    prefixIcon: const Icon(Icons.person),
                ),
                validator: (value) {
                    if (value!.isEmpty) {
                        return 'Please enter your username';
                    }
                    return null;
                },
            ),
            const SizedBox(height: 10),
            TextFormField(
                controller: passwordController,
                decoration: InputDecoration(
                    hintText: "Password",
                    border: OutlineInputBorder(
                        borderRadius: BorderRadius.circular(18),
                        borderSide: BorderSide.none),
                    fillColor: Colors.purple.withOpacity(0.1),
                    filled: true,
                    prefixIcon: const Icon(Icons.password),
                ),
                obscureText: true,
                validator: (value) {
                    if (value!.isEmpty) {
                        return 'Please enter your password';
                    }
                    return null;
                },
            ),
            const SizedBox(height: 10),
            ElevatedButton(

```

```
onPressed: () {
    if (usernameController.text == 'admin' &&
        passwordController.text == 'admin123!@#') {
        Navigator.push(context, MaterialPageRoute(builder: (context)=> const custom_gesture()));
        print('Login Successful');
    } else {
        print('Invalid Credentials');
    }
},
style: ElevatedButton.styleFrom(
    shape: const StadiumBorder(),
    padding: const EdgeInsets.symmetric(vertical: 16),
    backgroundColor: Colors.purple,
),
child: const Text(
    "Login",
    style: TextStyle(fontSize: 20, color: Colors.white),
),
),
),
),
);
}

_forgotPassword(context) {
    return TextButton(
        onPressed: () {
            Navigator.pushReplacement(context, MaterialPageRoute(builder: (context)=> const ForPassPage()));
        },
        child: const Text("Forgot password?",
            style: TextStyle(color: Colors.purple),
        ),
    );
}

signup(context) {
    return Row(
        mainAxisAlignment: MainAxisAlignment.center,
        children: [
            const Text("Don't have an account? "),
            TextButton(
                onPressed: () {
                    Navigator.push(context, MaterialPageRoute(builder: (context)=> SignupPage()));
                },
                child: const Text("Sign Up", style: TextStyle(color: Colors.purple)),
            )
        ],
    );
}
```

signup.dart:

```

import 'package:flutter/material.dart';
import 'package:flutter/services.dart';
import 'package:practical7/login.dart';

import 'custom_gesture.dart';

class SignupPage extends StatelessWidget {
    SignupPage({super.key});

    final TextEditingController emailController = TextEditingController();
    final TextEditingController usernameController = TextEditingController();
    final TextEditingController passwordController = TextEditingController();
    final TextEditingController conpasswordController = TextEditingController();
    final TextEditingController mobileController = TextEditingController();
}

```

@override

Widget build(BuildContext context) {

```

    return MaterialApp(
        debugShowCheckedModeBanner: false,
        home: Scaffold(
            body: SingleChildScrollView(
                child: Container(
                    padding: const EdgeInsets.symmetric(horizontal: 40),
                    height: MediaQuery.of(context).size.height - 50,
                    width: double.infinity,
                    child: Column(
                        mainAxisAlignment: MainAxisAlignment.spaceEvenly,
                        crossAxisAlignment: CrossAxisAlignment.stretch,
                        children: <Widget>[
                            Column(
                                children: <Widget>[
                                    const SizedBox(height: 60.0),
                                    const Text(
                                        "Sign up",
                                        style: TextStyle(
                                            fontSize: 30,
                                            fontWeight: FontWeight.bold,
                                        ),
                                    ),
                                    const SizedBox(height: 20.),
                                    Text(
                                        "Create your account",
                                        style: TextStyle(fontSize: 15, color: Colors.grey[700]),
                                    )
                                ],
                            ),
                            Column(
                                children: <Widget>[
                                    TextFormField(

```

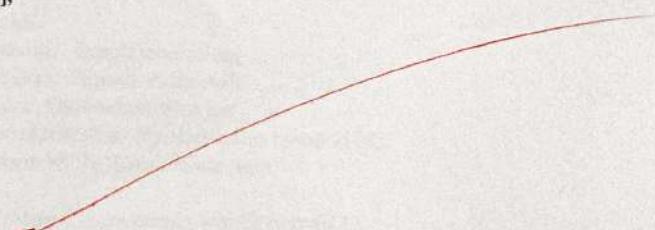
```
decoration: InputDecoration(  
    hintText: "Username",  
    border: OutlineInputBorder(  
        borderRadius: BorderRadius.circular(18),  
        borderSide: BorderSide.none),  
    fillColor: Colors.purple.withOpacity(0.1),  
    filled: true,  
    prefixIcon: const Icon(Icons.person)),  
validator: (value) {  
    if (value!.isEmpty) {  
        return 'Please enter your username';  
    }  
    return null;  
},  
,  
const SizedBox(height: 20),  
 TextFormField(  
    controller: emailController,  
    decoration: InputDecoration(  
        hintText: "Email",  
        border: OutlineInputBorder(  
            borderRadius: BorderRadius.circular(18),  
            borderSide: BorderSide.none),  
        fillColor: Colors.purple.withOpacity(0.1),  
        filled: true,  
        prefixIcon: const Icon(Icons.email)),  
    validator: (value) {  
        if (value!.isEmpty) {  
            return 'Please enter your email address';  
        }  
        return null;  
},  
,  
const SizedBox(height: 20),  
 TextFormField(  
    controller: mobileController,  
    decoration: InputDecoration(  
        hintText: "Mobile",  
        border: OutlineInputBorder(  
            borderRadius: BorderRadius.circular(18),  
            borderSide: BorderSide.none),  
        fillColor: Colors.purple.withOpacity(0.1),  
        filled: true,  
        prefixIcon: const Icon(Icons.person)),  
    keyboardType: TextInputType.number,  
    inputFormatters: <TextInputFormatter>[  
        FilteringTextInputFormatter.digitsOnly  
    ],  
    validator: (value) {  
        if (value!.isEmpty) {  
            return 'Please enter your Mobile Number';  
        }  
    },
```

```
        }
        return null;
    },
),
const SizedBox(height: 20),
TextField(
    controller: passwordController,
    decoration: InputDecoration(
        hintText: "Password",
        border: OutlineInputBorder(
            borderRadius: BorderRadius.circular(18),
            borderSide: BorderSide.none),
        fillColor: Colors.purple.withOpacity(0.1),
        filled: true,
        prefixIcon: const Icon(Icons.password),
    ),
    obscureText: true,
    validator: (value) {
        if (value!.isEmpty) {
            return 'Please enter your password';
        }
        return null;
    },
),
const SizedBox(height: 20),
TextField(
    controller: conpasswordController,
    decoration: InputDecoration(
        hintText: "Confirm Password",
        border: OutlineInputBorder(
            borderRadius: BorderRadius.circular(18),
            borderSide: BorderSide.none),
        fillColor: Colors.purple.withOpacity(0.1),
        filled: true,
        prefixIcon: const Icon(Icons.password),
    ),
    obscureText: true,
    validator: (value) {
        if(value != passwordController.text)
        return "Password Doesn't Match";
        if (value!.isEmpty) {
            return 'Please enter your password';
        }
        return null;
    },
),
],
),
Container(
    padding: const EdgeInsets.only(top: 3, left: 3),
    child: ElevatedButton(
```

```

onPressed: () {
    if (emailController.text == 'admin@gmail.com' &&
        passwordController.text == 'admin123!@#') {
        Navigator.pushReplacement(context, MaterialPageRoute(builder: (context)=> const
custom_gesture()));
        print('Login Successful');
    } else {
        print('Invalid Credentials');
    }
},
child: const Text(
    "Sign up",
    style: TextStyle(fontSize: 20, color: Colors.white),
),
style: ElevatedButton.styleFrom(
    shape: const StadiumBorder(),
    padding: const EdgeInsets.symmetric(vertical: 16),
    backgroundColor: Colors.purple,
),
)
),
),
Row(
mainAxisAlignment: MainAxisAlignment.center,
children: <Widget>[
    const Text("Already have an account?"),
    TextButton(
        onPressed: () {
            Navigator.push(context, MaterialPageRoute(builder: (context)=> LoginPage()));
        },
        child: const Text("Login", style: TextStyle(color: Colors.purple),)
    ),
],
),
),
),
),
);
}
}

```



forpass.dart:

```

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

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

@Override
Widget build(BuildContext context) {

```

```
return MaterialApp(  
  debugShowCheckedModeBanner: false,  
  home: Scaffold(  
    body: Container(  
      margin: const EdgeInsets.all(24),  
      child: Column(  
        mainAxisAlignment: MainAxisAlignment.spaceEvenly,  
        children: [  
          _header(context),  
          _inputField(context),  
        ],  
      ),  
    ),  
  ),  
);  
}  
  
_header(context) {  
  return const Column(  
    children: [  
      Text(  
        "Reset Your Password",  
        style: TextStyle(fontSize: 40, fontWeight: FontWeight.bold),  
      ),  
    ],  
  );  
}  
  
_inputField(context) {  
  return Column(  
    crossAxisAlignment: CrossAxisAlignment.stretch,  
    children: [  
      TextField(  
        decoration: InputDecoration(  
          hintText: "Email Address",  
          border: OutlineInputBorder(  
            borderRadius: BorderRadius.circular(18),  
            borderSide: BorderSide.none  
          ),  
          fillColor: Colors.purple.withOpacity(0.1),  
          filled: true,  
          prefixIcon: const Icon(Icons.person)),  
        ),  
      const SizedBox(height: 10),  
      ElevatedButton(  
        onPressed: () {  
          Navigator.push(context, MaterialPageRoute(builder: (context) => LoginPage()));  
        },  
        style: ElevatedButton.styleFrom(  
          shape: const StadiumBorder(),  
          padding: const EdgeInsets.symmetric(vertical: 16),  
        ),  
      ),  
    ],  
  );  
}
```

```
backgroundColor: Colors.purple,
),
child: const Text(
"Reset Password",
style: TextStyle(fontSize: 20, color: Colors.white),
),
),
],
);
}
}
```

custom_geasture.dart:

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

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

@Override
State<custom_geasture> createState() => _custom_geastureState();
}

class _custom_geastureState extends State<custom_geasture> {

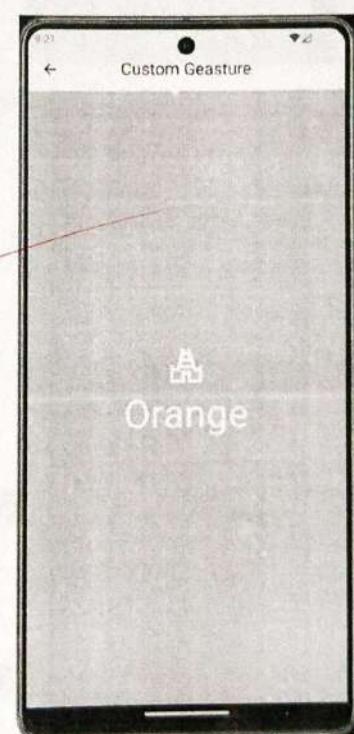
Color color1=Colors.orange;
String displayText = 'Orange';
IconData icn = Icons.temple_hindu_outlined;

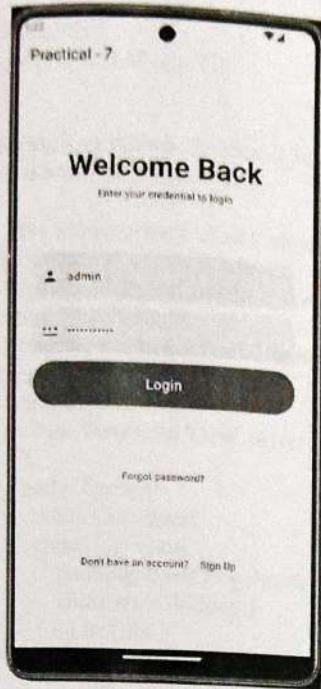
@Override
Widget build(BuildContext context) {

return Scaffold(
appBar: AppBar(
centerTitle: true,
title: Text("Custom Geasture"),
),
body: GestureDetector(
onTap: () {
setState(() {
if(color1==Colors.orange) {
color1 = Colors.blue;
displayText = 'Blue';
icn = Icons.radar;
} else if(color1==Colors.blue) {
color1=Colors.green;
displayText = 'Green';
icn = Icons.add_business;
} else {
color1=Colors.orange;
displayText = 'Orange';
icn = Icons.temple_hindu_rounded;
}
})
}
)
}
}
```

```
},
),
child: Center(
child: Container(
height: 1000,
width: 1000,
color: color1,
child: Center(
child: Column(
mainAxisAlignment: MainAxisAlignment.center,
children: [
Icon(
icon,
size: 50,
color: Colors.white,),
Text(displayText,
style: TextStyle(
fontSize: 50,
color: Colors.white,
)),
Container(
),
],
),
),
),
),
),
);
}
```

Output:





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

```

import 'package:flutter/material.dart';

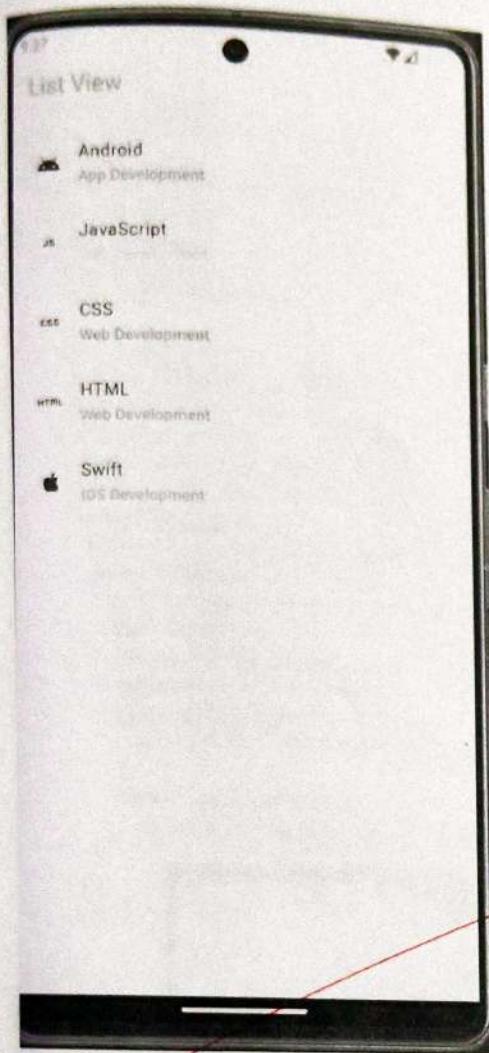
void main() {
  runApp(const MyApp());
}

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:



JH

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

```
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("Grid View"),
        ),
        body: Container(
          padding: EdgeInsets.all(12.0),
          child: GridView.count(
            crossAxisCount: 2,
            crossAxisSpacing: 10.0,
            mainAxisSpacing: 10.0,
            shrinkWrap: true,
            children: List.generate(10, (index) {
              return Padding(
                padding: const EdgeInsets.all(10.0),
                child: Container(
                  alignment: Alignment.center,
                  decoration: BoxDecoration(
                    color: Colors.lightGreenAccent,
                    borderRadius: BorderRadius.circular(12.0),
                  ),
                  child: Text('Item $index',
                    style: TextStyle(fontSize: 20, color: Colors.black87)),
                ),
              );
            })),
      ),
    );
  }
}
```

tput:



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

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("""
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? description) async { final db = await SQLHelper.db();
final data = {'title': title, 'description': description}; 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? description) async {
    final db = await SQLHelper.db();

    final data = { 'title': title,
        'description': description,
        'createdAt': DateTime.now().toString()
    };

    final result =
        await db.update('items', data, where: "id = ?", whereArgs: [id]);
    return result;
}

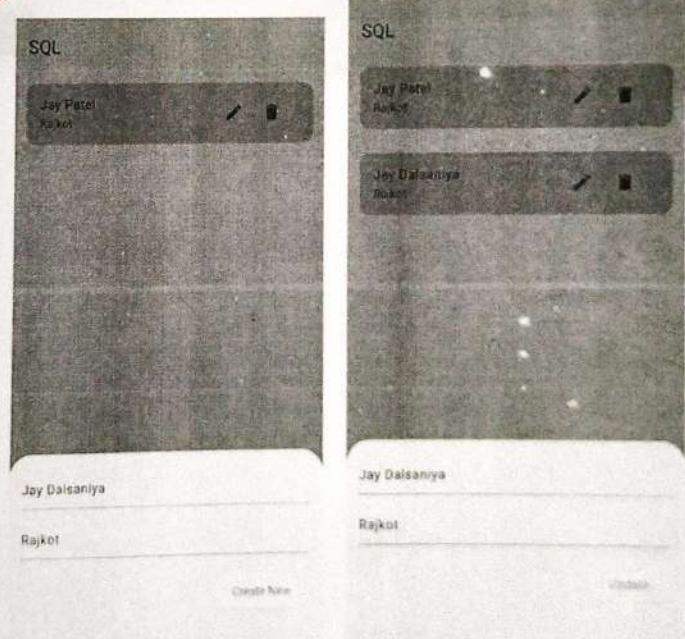
// Delete
static Future<void> deleteItem(int id) async {
    final db = await SQLHelper.db();
    try {
        await db.delete("items", where: "id = ?", whereArgs: [id]);
    } catch (err) {
        debugPrint("Something went wrong when deleting an item: $err");
    }
}
}

dependencies: flutter:
  sdk: flutter
  sqflite: ^2.0.0
  path_provider: any

```

Output:

JL



SQL

Jay Patel
marked



SQL

Jay Datsarliya
Marwadi University

View Details



Manan Varmora
Marwadi University

View Details



Hitesh Bhandari
Marwadi University

View Details



Successfully deleted a journal



jl

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

main.dart:

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

void main() { runApp(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

Post 10:

optio molestias id quia eum

qui et expedita modi cum officia vel magni
doloribus qui repudiandae

vero nisi sit

Quis 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 beatiae sit

vel qui neque voluptas ut commodi qui incident

ut animi commodi.

Post 12:

in quibusdam tempore odit est dolorem

laque id aut magnam

presentium 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 molitlia et saepe quo animi

ut dicta possimus sint molitlia 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 nisi 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 officia 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 quem

corporis rerum ducimus vel eum accusantium

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

```
import 'package:flutter/material.dart';
import '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
quo veniam quod sed accusamus veritatis error

Post 11:

et ea vero quia laudantium eutem
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 magnum
praesentium quis et ea odit et ea voluptas et
sapiente quis 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
blanditis culpa aut qua 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
qua voluptatibus provident quis necessitatibus ea
rerum repudiandae qua voluptatem delectus fugit aut id quia
ratione optio eos iusto veniam iure

Post 39:

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

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

main.dart:

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

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

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

homescreen.dart:

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

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

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

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

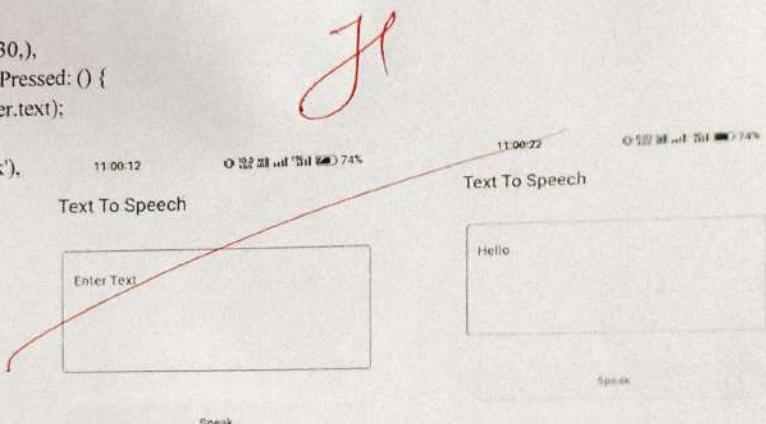
  @override
  void dispose() {
    textController.dispose();
    super.dispose();
  }
}
```

```

Future<void> speak(String text) async{
    await flutterTts.setLanguage('en-US');
    await flutterTts.setPitch(1.0);
    await flutterTts.setSpeechRate(0.5);
    await flutterTts.speak(text);
}

Widget build(BuildContext context) {
    return Scaffold(
        appBar: AppBar(
            title: Text("Text To Speech"),
        ),
        body: Padding(
            padding: EdgeInsets.all(20),
            child: Column(
                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:

Speak