**Practical 1: Android Studio setup for Flutter development along with Dart SDK.**

**Step 1: Download Flutter SDK:**

[Download](https://storage.googleapis.com/flutter_infra_release/releases/stable/windows/flutter_windows_3.13.7-stable.zip) the following installation bundle to get the latest stable release of the Flutter SDK

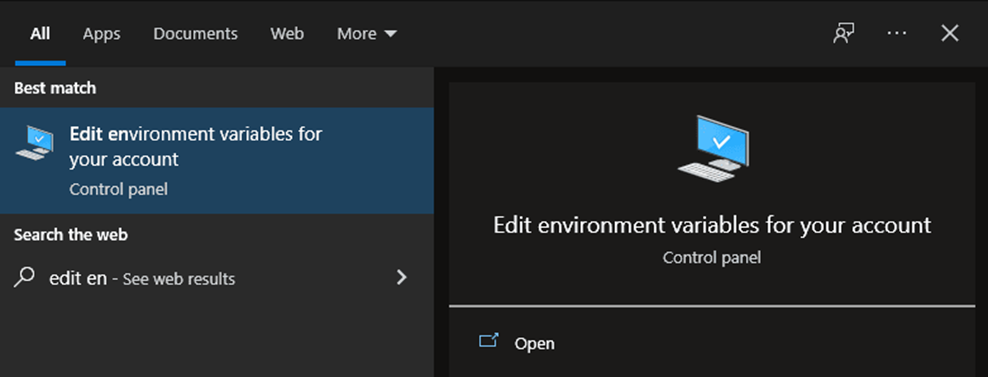
**Step 2: Extract the File: Extract the downloaded zip file and move it to the desired location where you want to install Flutter SDK**.

Do not install it in a folder or directory that requires elevated privileges, (such as *C:\Program Files\*) to ensure the program runs properly. For this tutorial, it will be stored in *C:\development\flutter*.

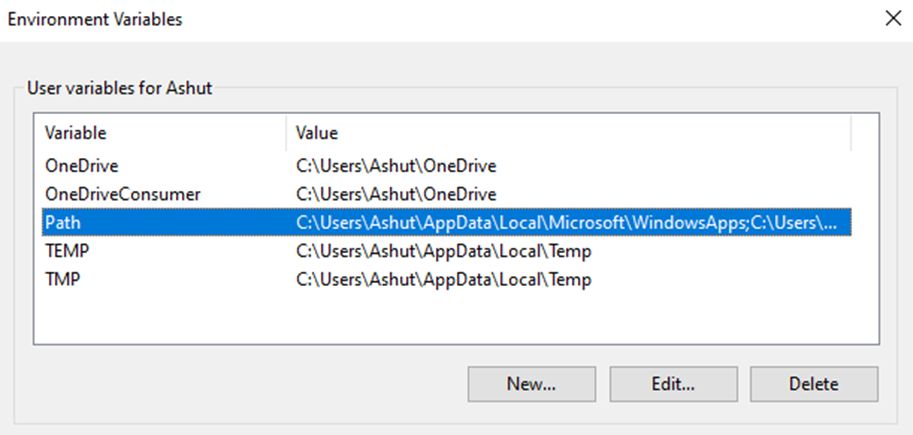
You are now ready to run Flutter commands in the Flutter Console.

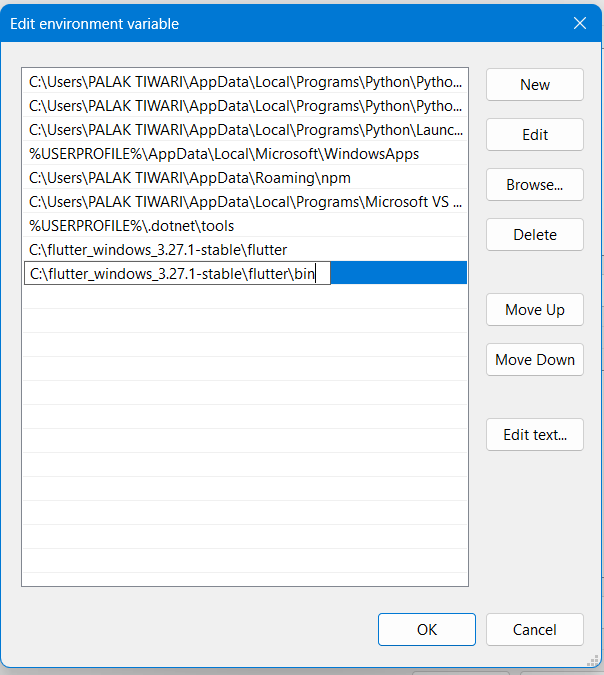
**Step 3: Update Path Variable for Windows PowerShell**

If you wish to run Flutter commands in the regular Windows console, take these steps to add Flutter to the PATH environment variable:From the Start search bar, enter ‘env’ and select Edit environment variables for your account.



* Under User variables check if there is an entry called Path:
* If the entry exists, append the full path to flutter\bin using; as a separator from existing values.

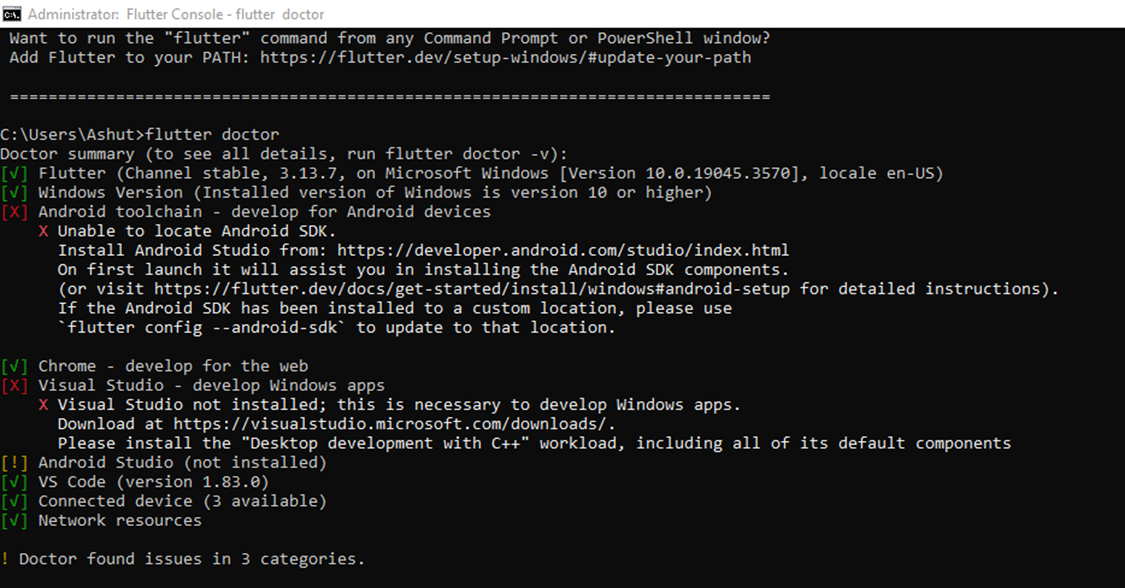


On the next screen, click New and add the full path to your *flutter\bin* directory. For this guide, it is shown below. Click OK on both windows to enable running Flutter commands in Windows consoles.

* If the entry doesn’t exist, create a new user variable named Path with the full path to flutter\bin as its value.

**Step 4: Confirm Installed Tools for Running Flutter**

In CMD, run the *flutter doctor*command to confirm the installed tools along with brief descriptions.

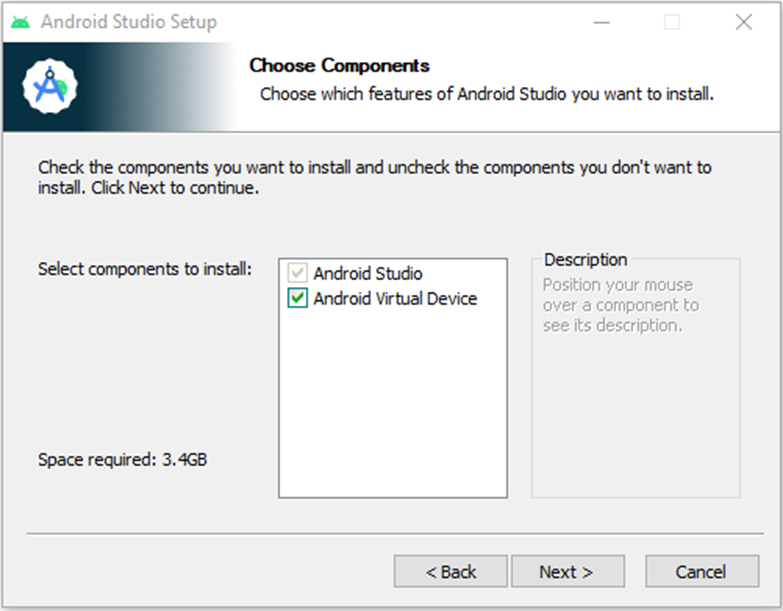


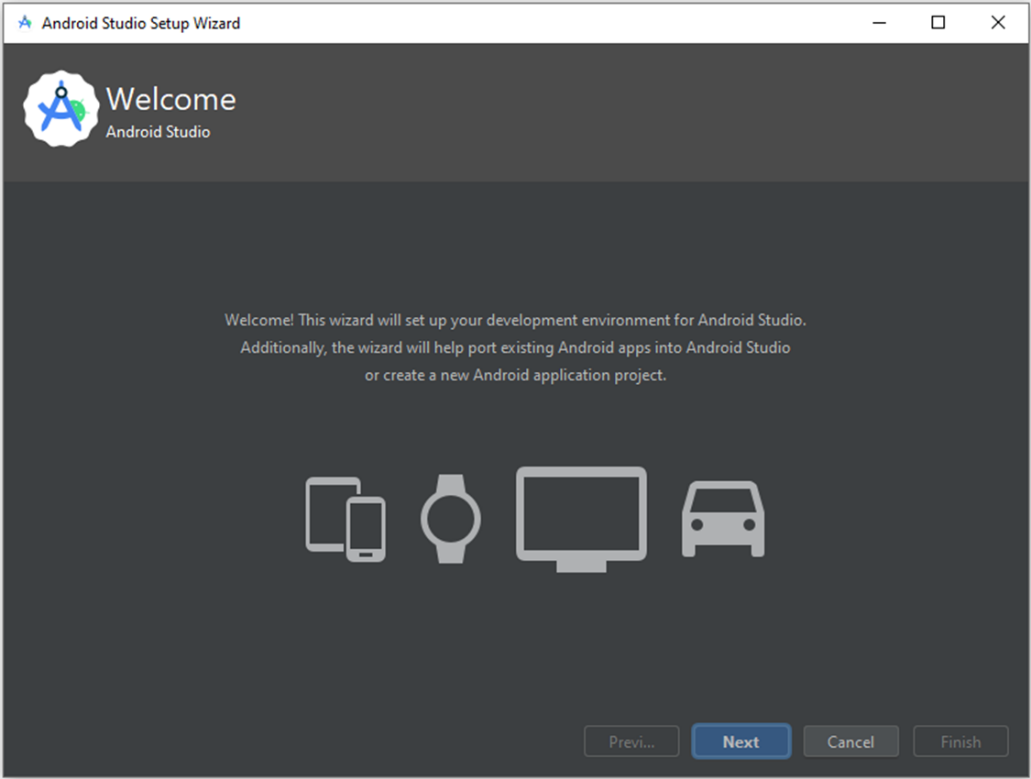
As visible, several components still need to be installed to complete the installation.

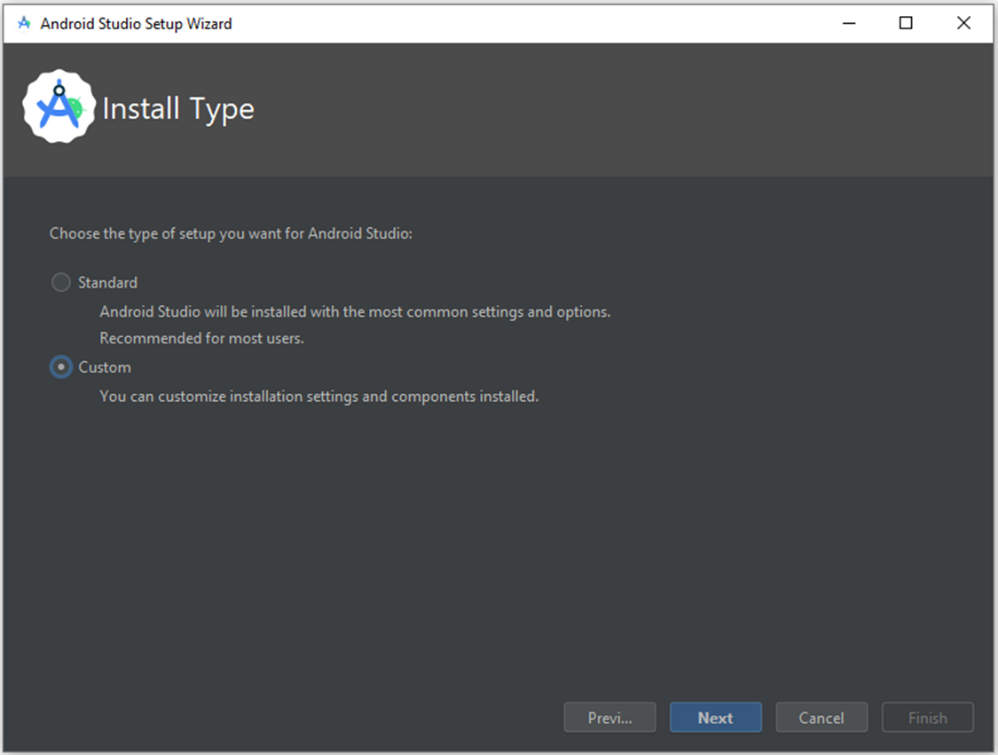
**Step 5: Download and Install Android Studio**

Download Android Studio:

* Visit the official Android Studio download page at <https://developer.android.com/studio>.
* Click on the “Download Android Studio” button.

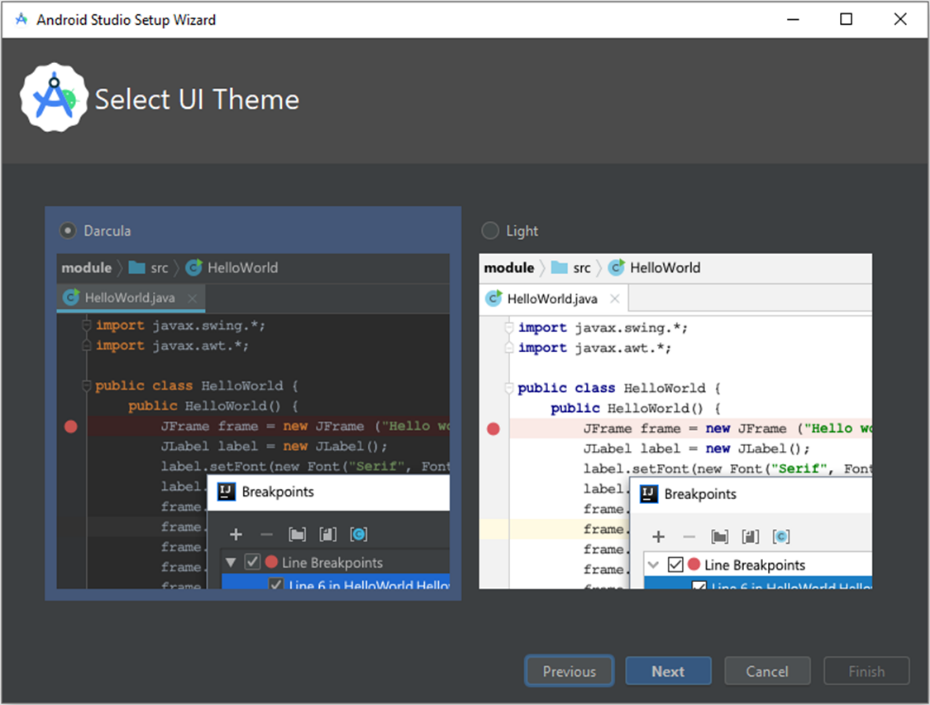
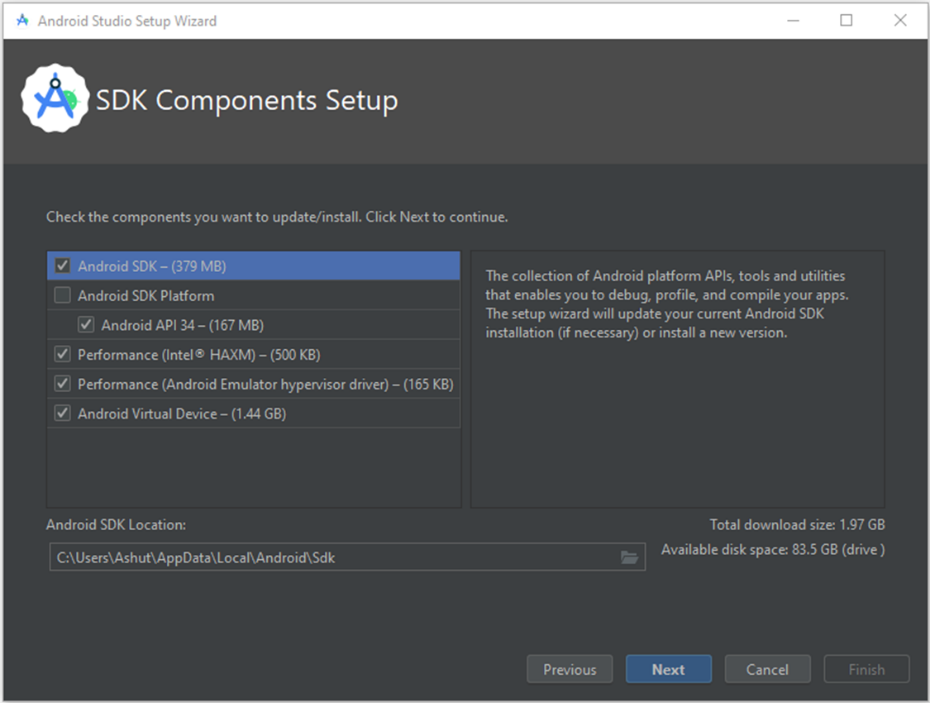
Next, proceed by downloading Android Studio. During the setup, unless you have unique requirements, simply click “Next” on all screens to keep the default settings. On the “Choose Components” screen, be sure to select the “Android Virtual Device” option to enable an Android emulator for your app development needs.

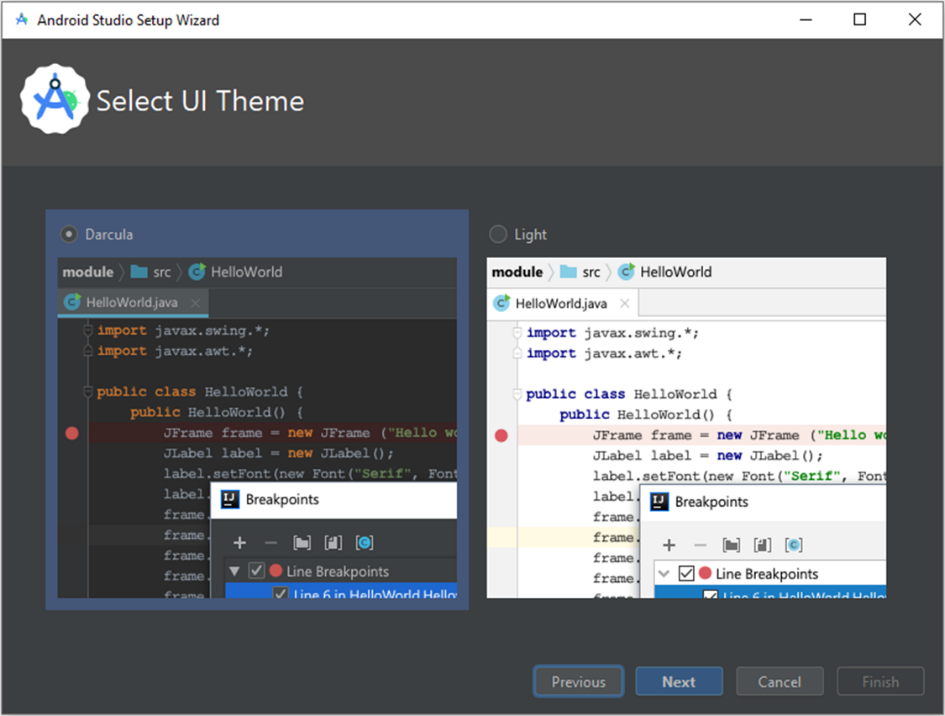
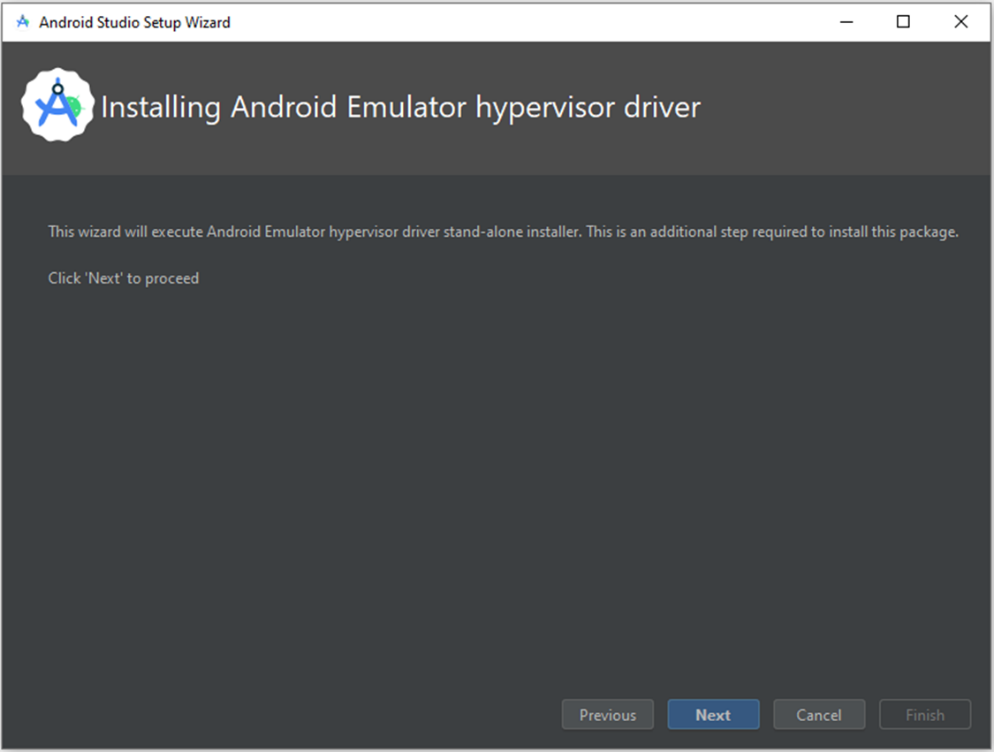
Afterward, The Android Studio Setup Wizard will start and you can proceed by clicking Next.

On the Install Type screen, select Custom and click Next

Select the installation location or leave the default path and click Next.

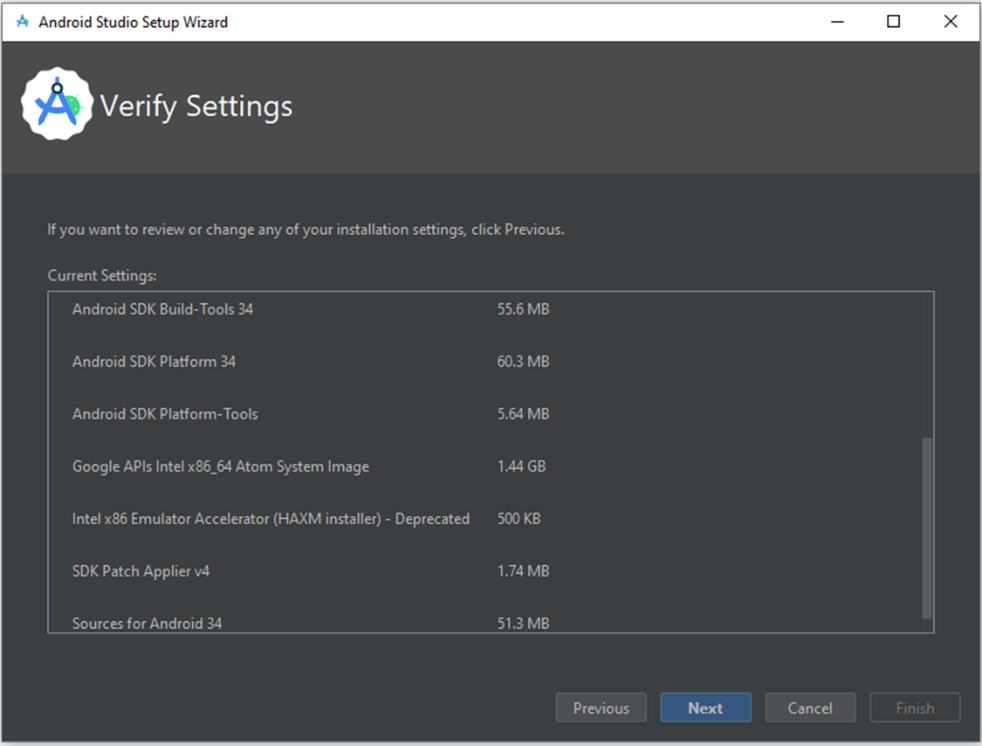
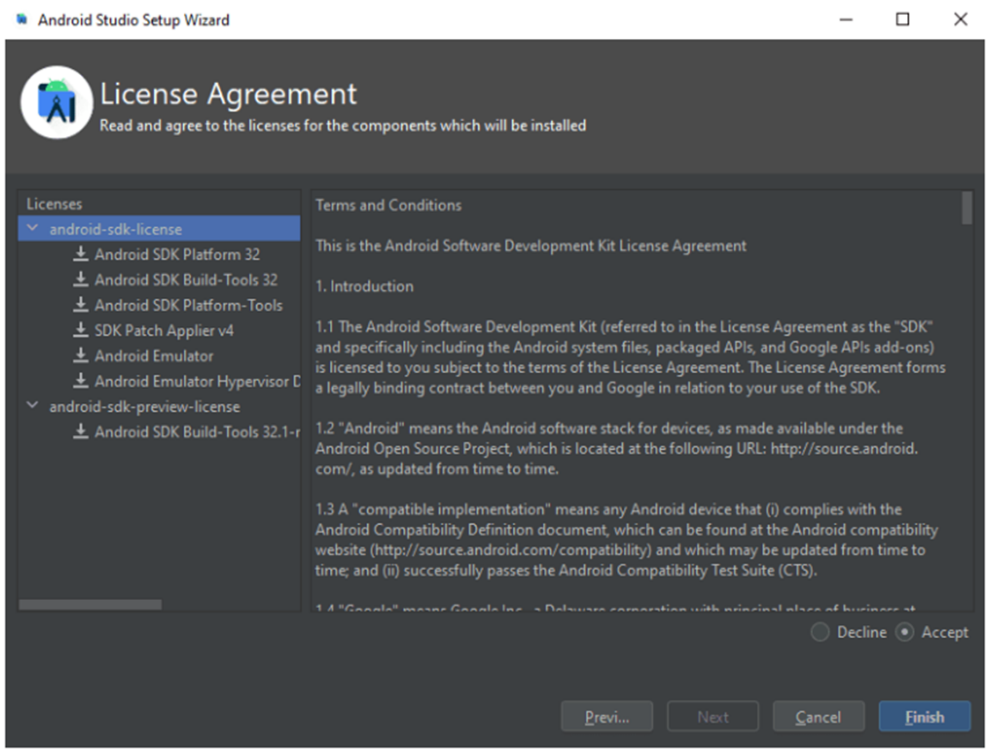
Select your UI theme and click Next.





Verify the selections and click Next.

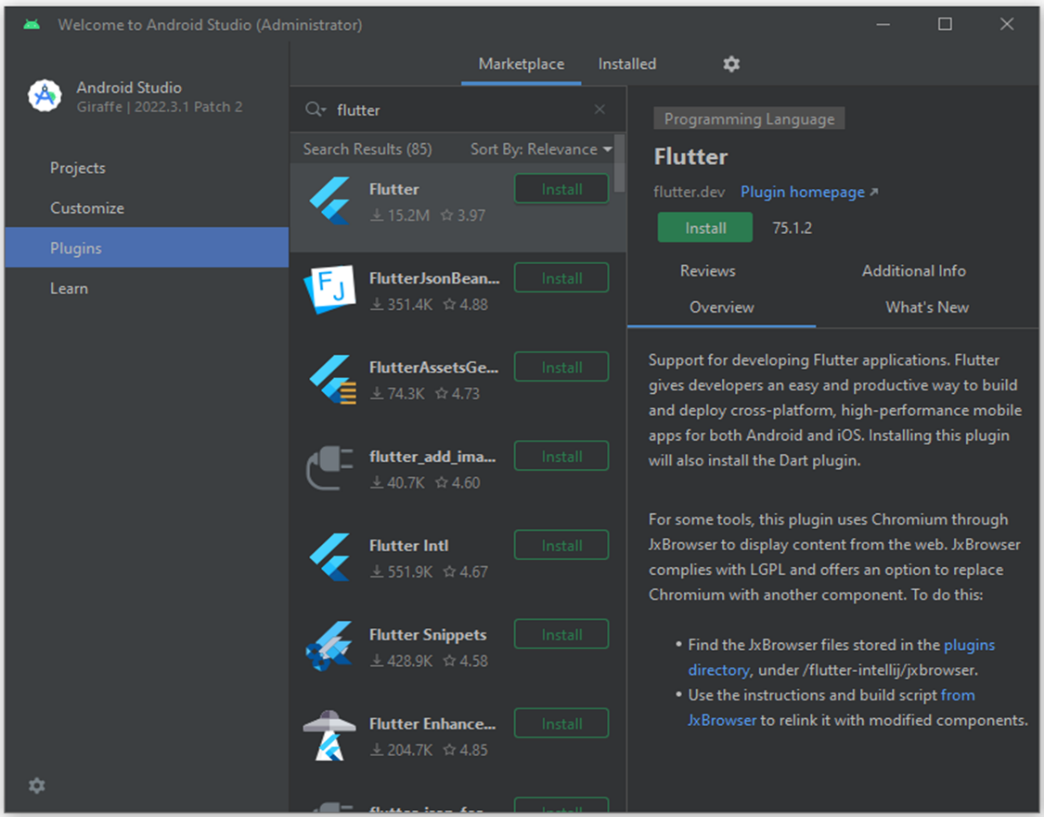
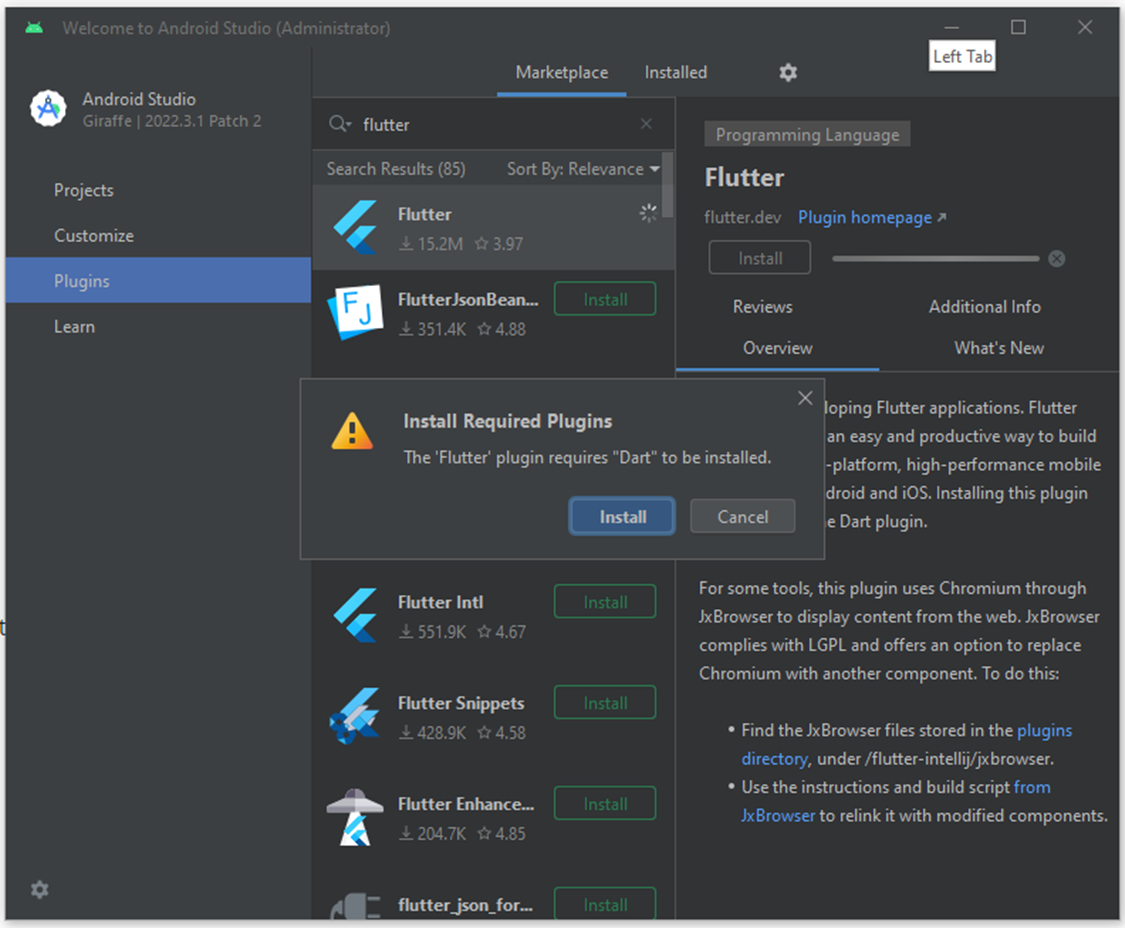
On the next screen, accept the License Agreement and click Finish.



The download of the components will start and Android Studio install. Once completed, click Finish.

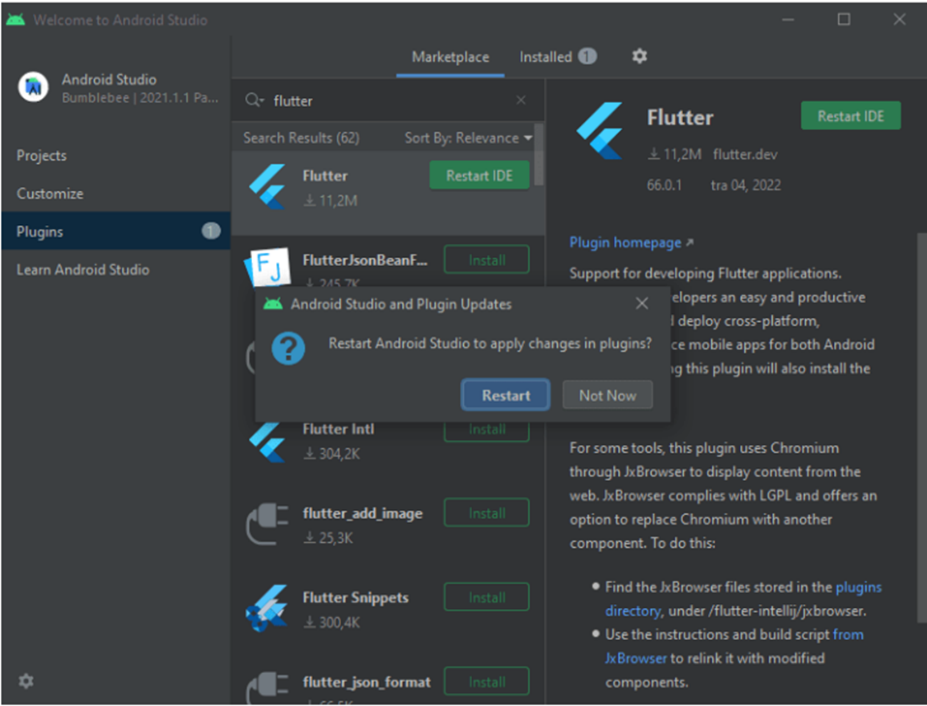
After the installation, start Android Studio. On the left side, click Plugins. Search for Flutter and click Install to install the Flutter plugin.

It will also prompt you to install Dart, a programming language used to create Flutter apps. Click Install at the prompt.



Finally, click Restart IDE so that the plugin changes are applied. Click Restart at the prompt

to confirm this action.

Afterward, run the *flutter doctor*command in CMD to confirm the Android Studio installation.

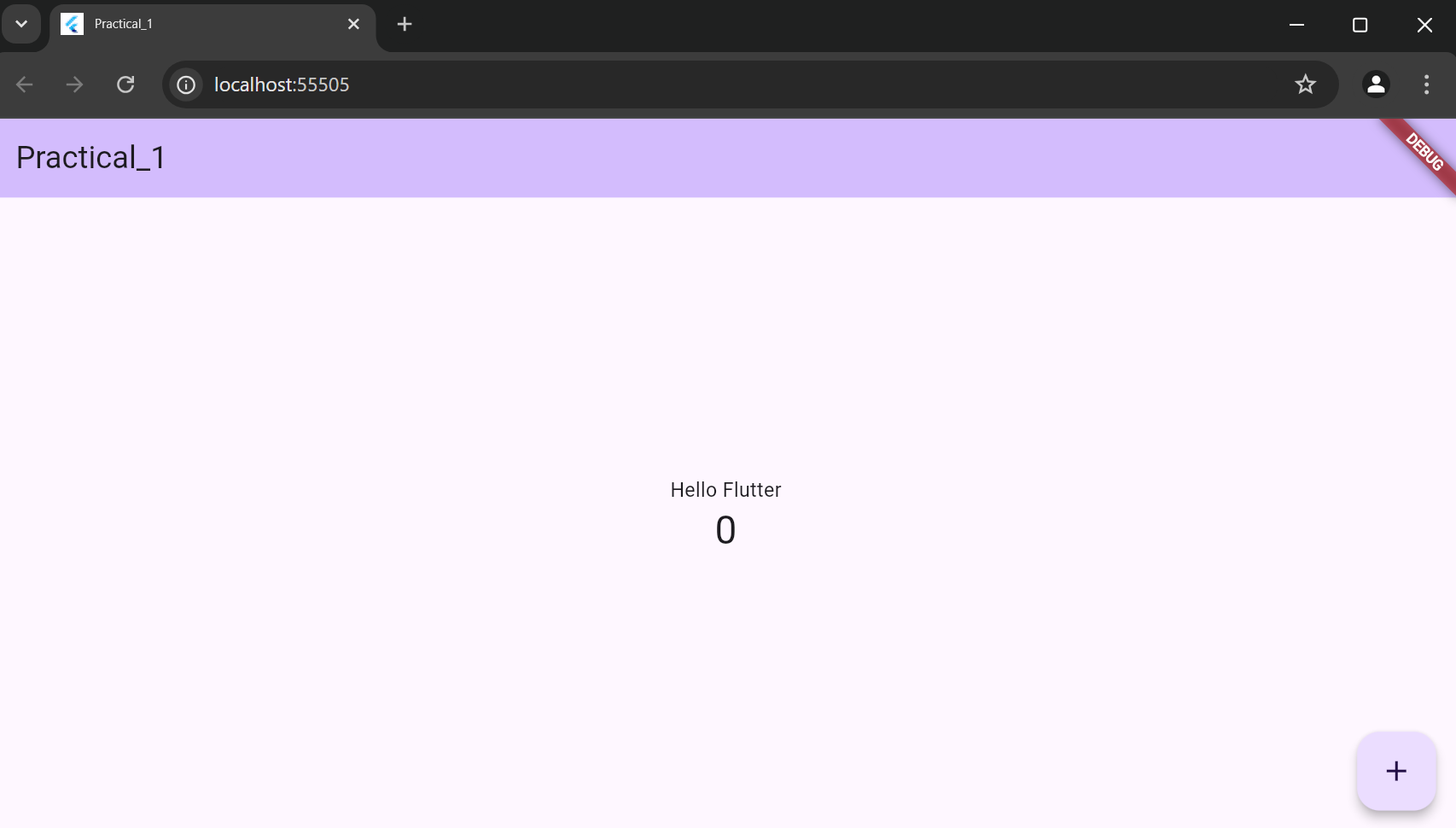
**Practical 2: Create a ‘Hello Flutter’ application.**

**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(  
 title: 'Practical\_1',  
 theme: ThemeData(  
  
 colorScheme: ColorScheme.fromSeed(seedColor: Colors.deepPurple),  
 useMaterial3: true,  
 ),  
 home: const MyHomePage(title: 'Practical\_1'),  
 );  
 }  
}  
  
class MyHomePage extends StatefulWidget {  
 const MyHomePage({super.key, required this.title});  
  
 final String title;  
  
 @override  
 State<MyHomePage> createState() => \_MyHomePageState();  
}  
  
class \_MyHomePageState extends State<MyHomePage> {  
 int \_counter = 0;  
  
 void \_incrementCounter() {  
 setState(() {  
  
 \_counter++;  
 });  
 }  
  
 @override  
 Widget build(BuildContext context) {  
  
 return Scaffold(  
 appBar: AppBar(  
 backgroundColor: Theme.of(context).colorScheme.inversePrimary,  
 title: Text(widget.title),

),  
 body: Center(  
 child: Column(  
 mainAxisAlignment: MainAxisAlignment.center,  
 children: <Widget>[  
 const Text(  
 'Hello Flutter',  
 ),  
 Text(  
 '$\_counter',  
 style: Theme.of(context).textTheme.headlineMedium,  
 ),  
 ],  
 ),  
 ),  
 floatingActionButton: FloatingActionButton(  
 onPressed: \_incrementCounter,  
 tooltip: 'Increment',  
 child: const Icon(Icons.add),  
 ),  
 );  
 }  
}

**Output**

****

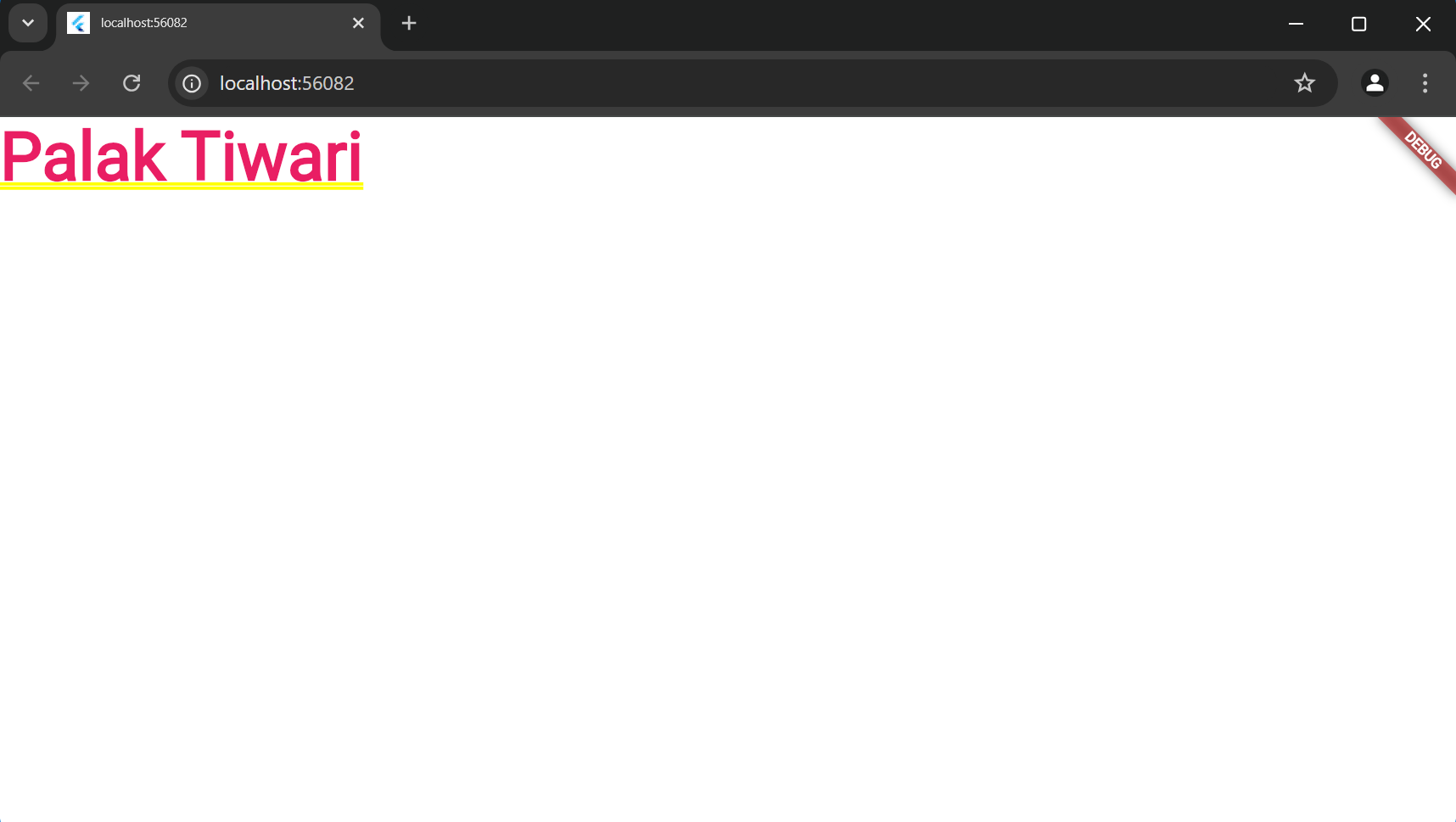
**CODE:**

import 'package:flutter/material.dart';  
  
void main() {  
 runApp(  
 MaterialApp(  
 home: Text("Palak Tiwari", style: TextStyle(color: Colors.pink ),)  
 )  
 );  
  
}



**Output:**



****

**Practical 4: Create and application using Flutter Key Widgets:**

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(

title: 'Flutter P3', theme: ThemeData(

colorScheme: ColorScheme.fromSeed(seedColor: Colors.deepPurple), useMaterial3: true,

),

home: const MyHomePage(title: 'Flutter Row and Column'),

);

}

}

class MyHomePage extends StatefulWidget {

const MyHomePage({super.key, required this.title}); final String title;

@override

State<MyHomePage> createState() => \_MyHomePageState();

}

class \_MyHomePageState extends State<MyHomePage> { @override

Widget build(BuildContext context) { return Scaffold(

appBar: AppBar(

backgroundColor: Theme.of(context).colorScheme.inversePrimary, title: Text(widget.title),

),

body: Column( children: <Widget>[

Text('Row',style: TextStyle(fontSize: 25),), Center(

child: Row(

mainAxisAlignment: MainAxisAlignment.spaceEvenly, children: <Widget>[

Container(

margin: EdgeInsets.all(12.0), padding: EdgeInsets.all(8.0), decoration:BoxDecoration(

borderRadius:BorderRadius.circular(8), color:Colors.lightBlue

),

child: Text(

'Flutter',

style: TextStyle( color: Colors.red, fontSize: 20

),

),

),

Container(

margin: EdgeInsets.all(15.0), padding: EdgeInsets.all(8.0), decoration:BoxDecoration(

borderRadius:BorderRadius.circular(8), color:Colors.green

),

child: Text("Android",style: TextStyle(color:Colors.orange,fontSize:25),),

),

Container(

margin: EdgeInsets.all(12.0), padding: EdgeInsets.all(8.0), decoration:BoxDecoration(

borderRadius:BorderRadius.circular(8), color:Colors.yellow

),

child: Text("Firebase",style: TextStyle(color:Colors.blue,fontSize:25),),

)

],

),

),

Column(

children: <Widget>[

Text('Column',style: TextStyle(fontSize: 25),), Container(

margin: EdgeInsets.all(20.0), padding: EdgeInsets.all(12.0), decoration: BoxDecoration(

borderRadius: BorderRadius.circular(8), color: Colors.red,

),

child: Text( "React.js",

style: TextStyle(color: Colors.yellowAccent, fontSize: 20),

),

),

Container(

margin: EdgeInsets.all(20.0), padding: EdgeInsets.all(12.0), decoration: BoxDecoration(

borderRadius: BorderRadius.circular(8), color: Colors.red,

),

child: Text( "Flutter",

style: TextStyle(color: Colors.yellowAccent, fontSize: 20),

),

),

Container(

margin: EdgeInsets.all(20.0), padding: EdgeInsets.all(12.0), decoration: BoxDecoration(

borderRadius: BorderRadius.circular(8), color: Colors.red,

),

child: Text( "MySQL",

style: TextStyle(color: Colors.yellowAccent, fontSize: 20),

),

),

],

),

],

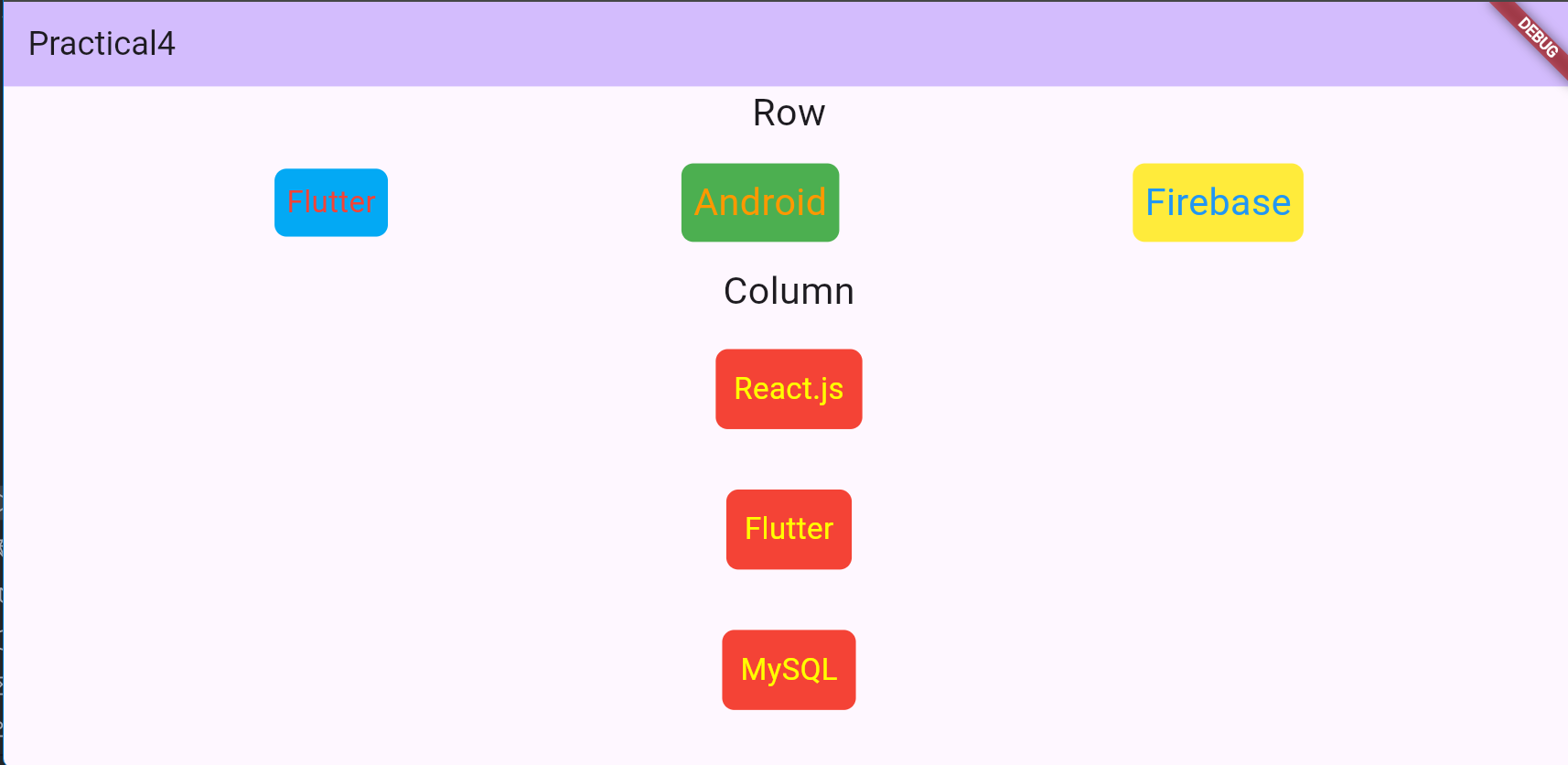
),

);

}

}

**Output:**

****

**Practical 5: Create and application with Flutter UI Components**.

**CODE:**

import 'package:flutter/material.dart'; void main() {

runApp(MyApp());

}

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

@override

Widget build(BuildContext context) { return MaterialApp( debugShowCheckedModeBanner: false, theme: ThemeData(

primaryColor: Colors.*blueAccent*,

scaffoldBackgroundColor: Colors.*grey*[200], // Background color inputDecorationTheme: InputDecorationTheme(

filled: true,

fillColor: Colors.*white*, border: OutlineInputBorder(

borderRadius: BorderRadius.circular(10.0), borderSide: BorderSide.*none*, // Remove default border

),

enabledBorder: OutlineInputBorder( borderRadius: BorderRadius.circular(10.0),

borderSide: BorderSide(color: Colors.*blueAccent*, width: 1),

),

focusedBorder: OutlineInputBorder( borderRadius: BorderRadius.circular(10.0),

borderSide: BorderSide(color: Colors.*blue*, width: 2),

),

),

),

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: const EdgeInsets.all(20.0), child: Container(

width: 400, // Reduce the width to make the box smaller padding: const EdgeInsets.all(20.0),

decoration: BoxDecoration( color: Colors.*white*,

borderRadius: BorderRadius.circular(15.0), boxShadow: [

BoxShadow(

color: Colors.*grey*.withOpacity(0.2), spreadRadius: 5,

blurRadius: 7,

offset: Offset(0, 3), // Shadow position

),

],

),

child: Column(

mainAxisSize: MainAxisSize.min, children: [

const Text( "Login",

style: TextStyle( fontSize: 24,

fontWeight: FontWeight.*bold*, color: Colors.*blueAccent*,

),

),

const SizedBox(height: 20), TextField(

controller: emailController, decoration:

InputDecoration(

labelText: 'Email',

prefixIcon: Icon(Icons.*email*, color: Colors.*blueAccent*),

),

keyboardType: TextInputType.*emailAddress*,

),

const SizedBox(height: 10), TextField(

controller: passwordController, decoration: InputDecoration(

prefixIcon: Icon(Icons.*lock*, color: Colors.*blueAccent*),

),

obscureText: true,

),

const SizedBox(height: 20), SizedBox(

width: double.*infinity*, child: ElevatedButton( onPressed: () {

String email = emailController.text;

String password = passwordController.text; print('Email: $email, Password: $password');

},

style: ElevatedButton.*styleFrom*(

padding: const EdgeInsets.symmetric(vertical: 12), shape: RoundedRectangleBorder(

borderRadius: BorderRadius.circular(10),

),

backgroundColor: Colors.*blueAccent*,

),

child: const Text( 'Login',

style: TextStyle(fontSize: 18, color: Colors.*white*),

),

),

),

const SizedBox(height: 20),

// Optional: Add footer text (like a website) Align(

alignment: Alignment.*center*, child: TextButton( onPressed: () {},

labelText: 'Password',

child: const Text( "Forgot Password?",

style: TextStyle(color: Colors.*blueAccent*),

),

),

),

],

),

),

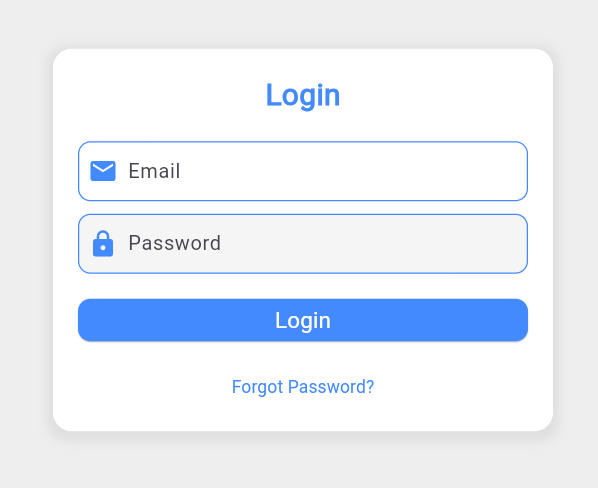
),

),

);

}

}

**OUTPUT:**

**Practical 6: Create and application with Flutter UI Components**. **CODE:**

import 'package:flutter/material.dart'; void main() {

runApp(MyApp());

}

class MyApp extends StatelessWidget { @override

Widget build(BuildContext context) { return MaterialApp( debugShowCheckedModeBanner: false, theme: ThemeData(

primaryColor: Colors.*blueAccent*,

visualDensity: VisualDensity.*adaptivePlatformDensity*,

),

home: SignupForm(),

);

}

}

class SignupForm extends StatefulWidget { @override

\_SignupFormState createState() => \_SignupFormState();

}

class \_SignupFormState extends State<SignupForm> { final \_formKey = GlobalKey<FormState>();

final TextEditingController \_nameController = TextEditingController(); final TextEditingController \_emailController = TextEditingController(); final TextEditingController \_passwordController = TextEditingController(); final TextEditingController \_phoneController = TextEditingController(); final TextEditingController \_addressController = TextEditingController(); String \_selectedGender = 'Male';

bool \_isAgreed = false;

void \_submitForm() {

if (\_formKey.currentState?.validate() ??

false) { if (\_isAgreed) {

ScaffoldMessenger.*of*(context).showSnackBar( SnackBar(content: Text('Signup Successful')));

} else { ScaffoldMessenger.*of*(context).showSnackBar(

SnackBar(content: Text('You must agree to the terms')));

}

}

}

@override

Widget build(BuildContext context) { return Scaffold(

body: Center( child: Padding(

padding: EdgeInsets.symmetric(horizontal: 20.0), child: SingleChildScrollView(

child: Container(

width: MediaQuery.*of*(context).size.width < 600 ? double.*infinity* : 600, padding: EdgeInsets.all(20),

decoration: BoxDecoration(

borderRadius: BorderRadius.circular(10),

border: Border.all(color: Colors.*blueAccent*, width: 2), boxShadow: [

BoxShadow(

color: Colors.*black26*, blurRadius: 10,

offset: Offset(0, 4),

),

],

color: Colors.*white*,

),

child: Column(

crossAxisAlignment: CrossAxisAlignment.start, children: [

Center( child: Text(

'Signup Form', style: TextStyle( fontSize: 28,

fontWeight: FontWeight.*bold*, color: Colors.*blueAccent*,

),

),

),

SizedBox(height: 20), TextFormField(

controller: \_nameController, decoration: InputDecoration( labelText: 'Full Name',

hintText: 'Enter your full name', border: OutlineInputBorder(),

),

validator: (value) {

if (value == null || value.isEmpty) { return 'Please enter your full name.';

}

return null;

},

),

SizedBox(height: 20), TextFormField(

controller: \_emailController, decoration: InputDecoration( labelText: 'Email Address',

hintText: 'Enter your email address', border: OutlineInputBorder(),

),

validator: (value) {

if (value == null || value.isEmpty) { return 'Please enter your email address.';

}

if (!RegExp(r'\S+@\S+\.\S+').hasMatch(value)) { return 'Please enter a valid email address.';

}

return null;

},

),

SizedBox(height: 20), TextFormField(

controller: \_phoneController, decoration: InputDecoration( labelText: 'Phone Number',

hintText: 'Enter your phone number', border: OutlineInputBorder(),

),

validator: (value) {

if (value == null || value.isEmpty) {

return 'Please enter your phone number.';

}

return null;

},

),

SizedBox(height: 20), TextFormField(

controller: \_addressController, decoration: InputDecoration( labelText: 'Address',

hintText: 'Enter your address',

border: OutlineInputBorder(),

),

validator: (value) {

if (value == null || value.isEmpty) { return 'Please enter your address.';

}

return null;

},

),

SizedBox(height: 20), TextFormField(

controller: \_passwordController, decoration: InputDecoration( labelText: 'Password', hintText: 'Enter your password', border: OutlineInputBorder(),

),

obscureText: true, validator: (value) {

if (value == null || value.isEmpty) { return 'Please enter your password.';

}

if (value.length < 6) {

return 'Password must be at least 6 characters.';

}

return null;

},

),

SizedBox(height: 20), DropdownButtonFormField<String>( value: \_selectedGender,

decoration: InputDecoration( labelText: 'Gender',

border: OutlineInputBorder(),

),

onChanged: (String? newValue) { setState(() {

\_selectedGender = newValue!;

});

},

items: ['Male', 'Female', 'Other'].map((gender) { return DropdownMenuItem<String>(

value: gender, child: Text(gender),

);

}).toList(),

),

SizedBox(height: 20), Row(

children: [ Checkbox(

value: \_isAgreed, onChanged: (bool? value) { setState(() { isAgreed = value!;

});

},

),

Text('I agree to the terms and conditions'),

],

),

SizedBox(height: 20), ElevatedButton( onPressed: \_submitForm, child: Text('Signup'),

style: ElevatedButton.*styleFrom*(

padding: EdgeInsets.symmetric(vertical: 15), shape: RoundedRectangleBorder(

borderRadius: BorderRadius.circular(10),

),

),

),

],

),

),

),

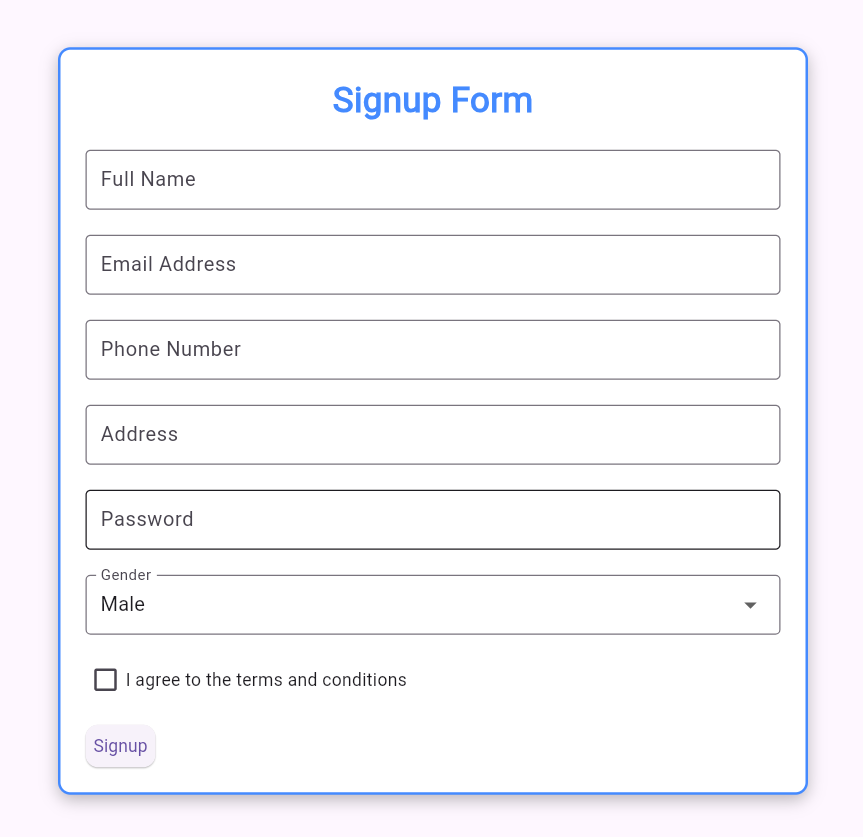
),

),

);

}

}

Output :

**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\_geasture.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\_geasture())); 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\_geasture.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), 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: 20), TextFormField(

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'](mailto:%27admin@gmail.com) && passwordController.text == 'admin123!@#') {

Navigator.pushReplacement(context, MaterialPageRoute(builder: (context)=> const custom\_geasture()));

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( icn, size: 50,

color: Colors.white,), Text(displayText, style: TextStyle( fontSize: 50,

color: Colors.white,

),),

Container(

)

],

),

),

),

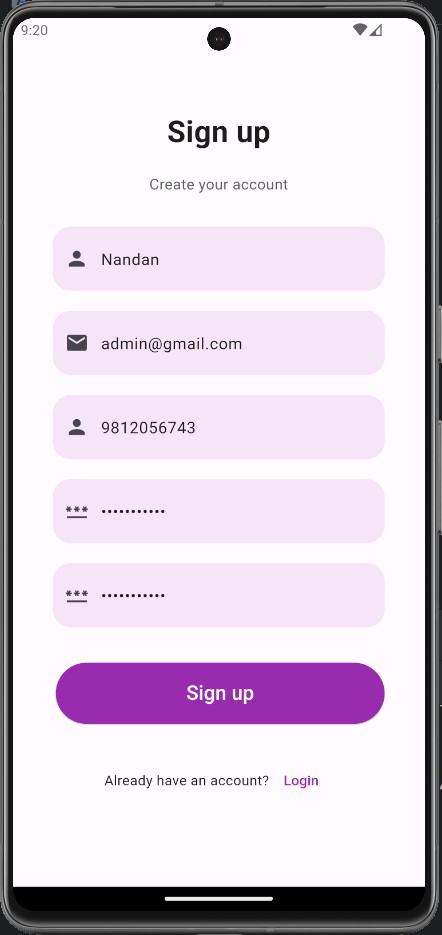
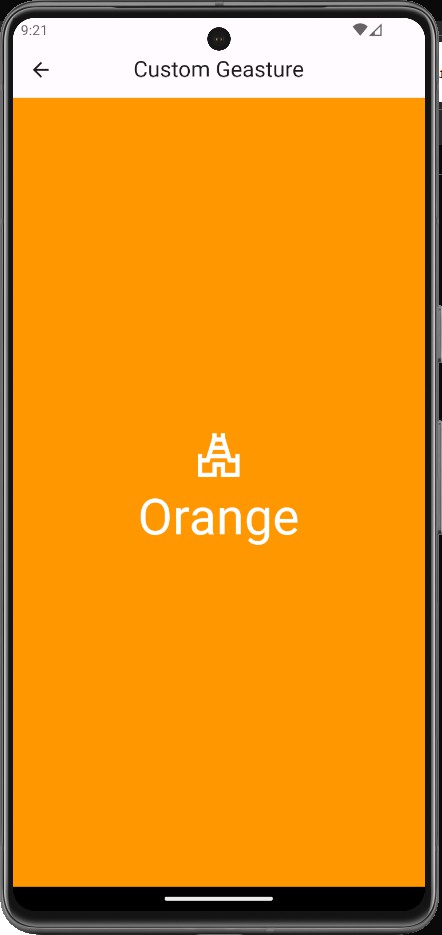
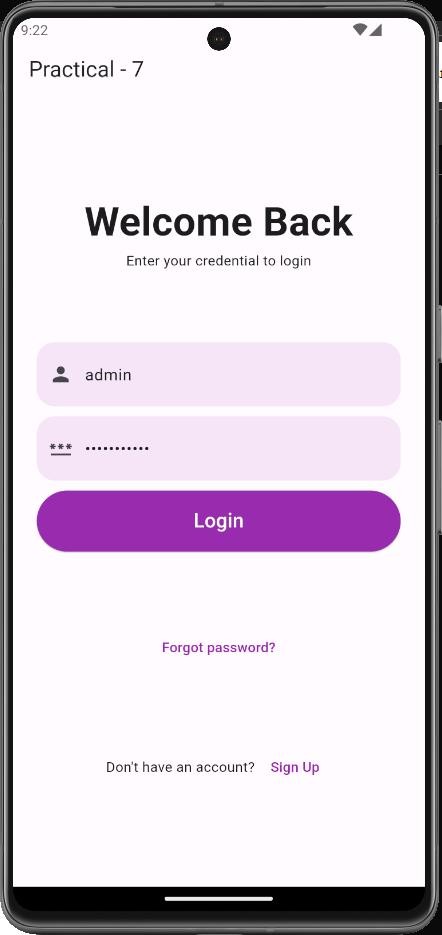
),

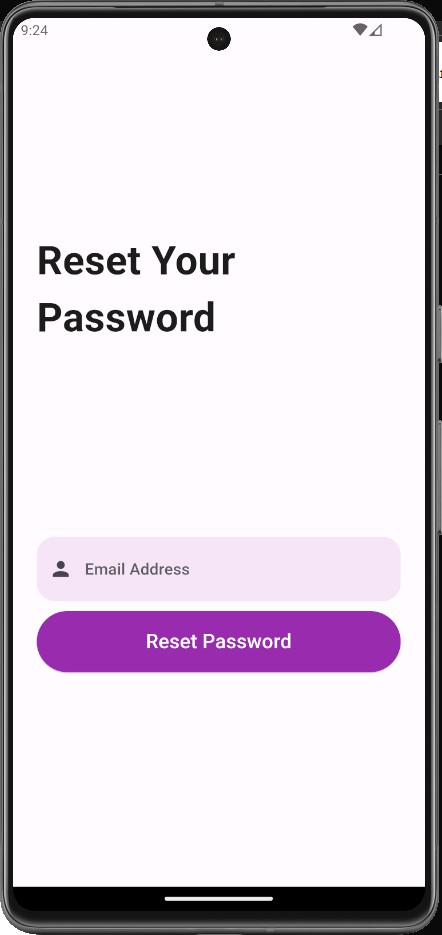
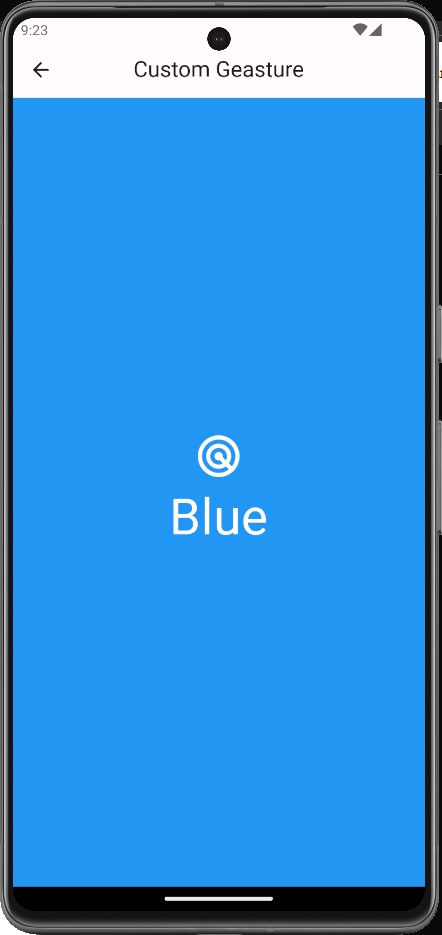
),

);

}

}

**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),),

),

],

),

),

),

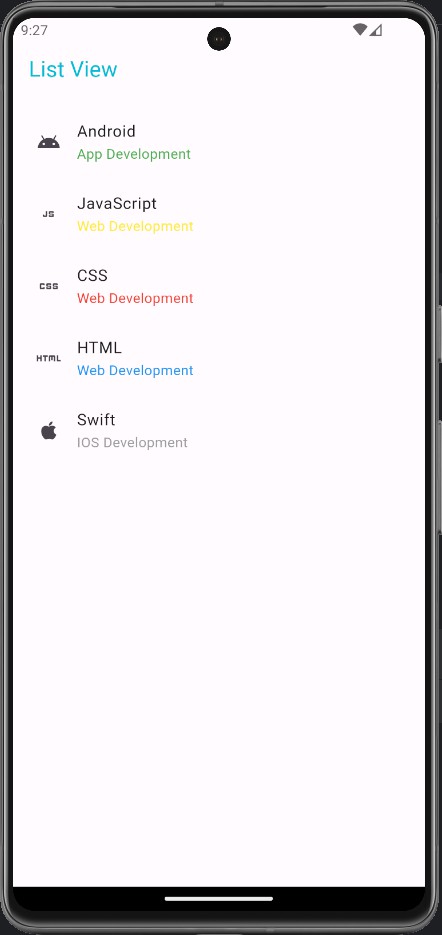
),

);

}

}

**Output:**



**Practical 9: Create and 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),),),

);

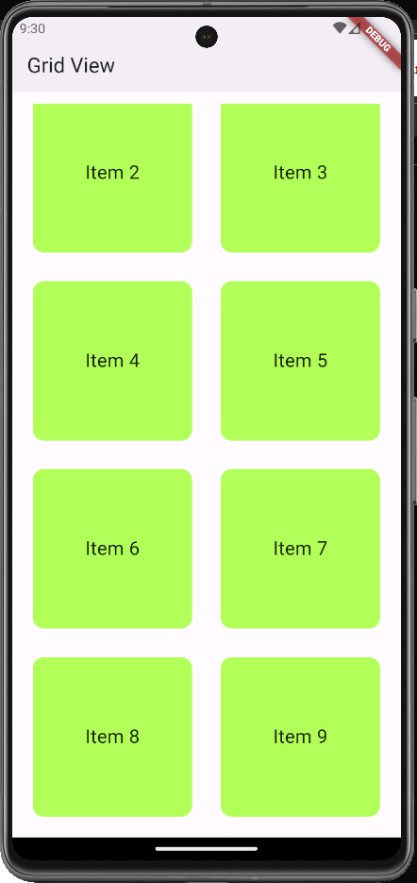
}),

), ), ),

);

}

}



**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 {

if (id == null) { await \_addItem();

}

if (id != null) {

await \_updateItem(id);

}

\_titleController.text = '';

\_descriptionController.text = '';

)

],

),

));

}

},

child: Text(id == null ? 'Create New' : 'Update'),

<void> \_addItem() async { await SQLHelper.createItem(

\_titleController.text, \_descriptionController.text);

\_refreshJournals();

}

{ await SQLHelper.updateItem( id, \_titleController.text, \_descriptionController.text);

\_refreshJournals();

}

void \_deleteItem(int id) async { await SQLHelper.deleteItem(id); ScaffoldMessenger.of(context).showSnackBar(const SnackBar( content: Text('Successfully deleted a journal!'),

));

\_refreshJournals();

}

@override

Widget build(BuildContext context) { return Scaffold( appBar: AppBar(

title: const Text('SQL'),

),

body: \_isLoading

? const Center(

child: CircularProgressIndicator(),

)

: ListView.builder( itemCount: \_journals.length,

itemBuilder: (context, index) => Card( color: Colors.orange[200], margin: const EdgeInsets.all(15), child: ListTile(

title: Text(\_journals[index]['title']),

subtitle: Text(\_journals[index]['description']), trailing: SizedBox( width: 100, child: Row( children: [

IconButton(

icon: const Icon(Icons.edit),

onPressed: () => \_showForm(\_journals[index]['id']),

),

IconButton(

icon: const Icon(Icons.delete), onPressed: () =>

\_deleteItem(\_journals[index]['id']),

),

],

),

)),

),

),

floatingActionButton: FloatingActionButton( child: const Icon(Icons.add), onPressed: () => \_showForm(null),

),

);

}

}

**sqlHelper.dart:**

import 'package:flutter/foundation.dart'; import 'package:sqflite/sqflite.dart' as sql; class SQLHelper {

static Future<void> createTables(sql.Database database) async { await database.execute("""CREATE

TABLE items(

id INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,

title TEXT, description TEXT,

createdAt TIMESTAMP NOT NULL DEFAULT CURRENT\_TIMESTAMP

)

""");

}

static Future<sql.Database> db() async { return sql.openDatabase( 'dbtech.db', version: 1,

onCreate: (sql.Database database, int version) async { await createTables(database);

},

);

}

static Future<int> createItem(String title, String? descrption) async { final db = await SQLHelper.db();

final data = {'title': title, 'description': descrption}; final id = await db.insert('items', data, conflictAlgorithm: sql.ConflictAlgorithm.replace); return id;

}

static Future<List<Map<String, dynamic>>> getItems() async { final db = await SQLHelper.db(); return db.query('items', orderBy: "id");

}

Future<List<Map<String, dynamic>>> getItem(int id) async {

final db = await SQLHelper.db();

return db.query('items', where: "id = ?", whereArgs: [id], limit: 1);

}

static Future<int> updateItem(

int id, String title, String? descrption) async { final db = await SQLHelper.db();

final data = { 'title': title, 'description': descrption,

'createdAt': DateTime.now().toString()

};

final result =

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

}

// Delete

static Future<void> deleteItem(int id) async { final db = await SQLHelper.db(); try {

await db.delete("items", where: "id = ?", whereArgs: [id]);

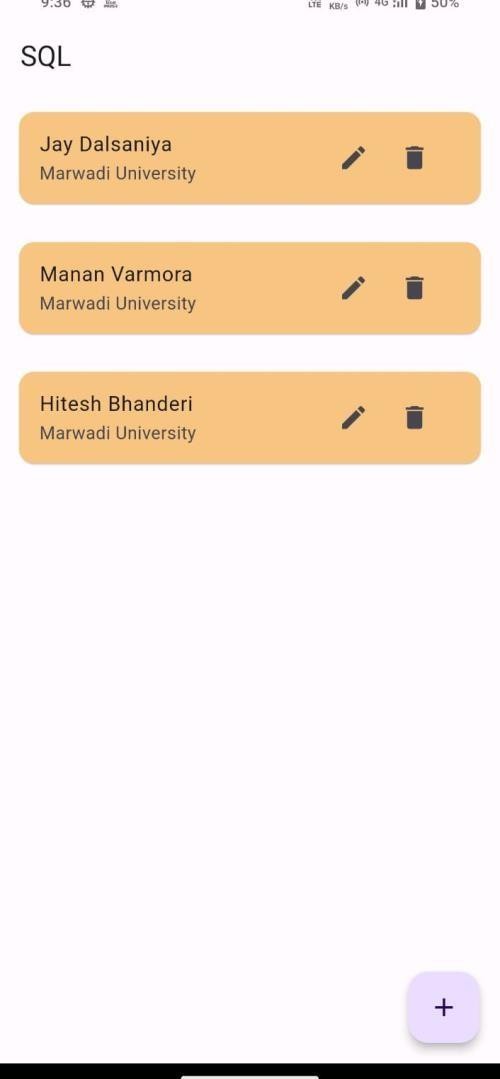
} catch (err) {

debugPrint("Something went wrong when deleting an item: $err");

dependencies: flutter:

sdk: flutter sqflite: ^2.0.0 path: ^1.9.0 path\_provider: any

**Output:**



**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}");

}

},

),

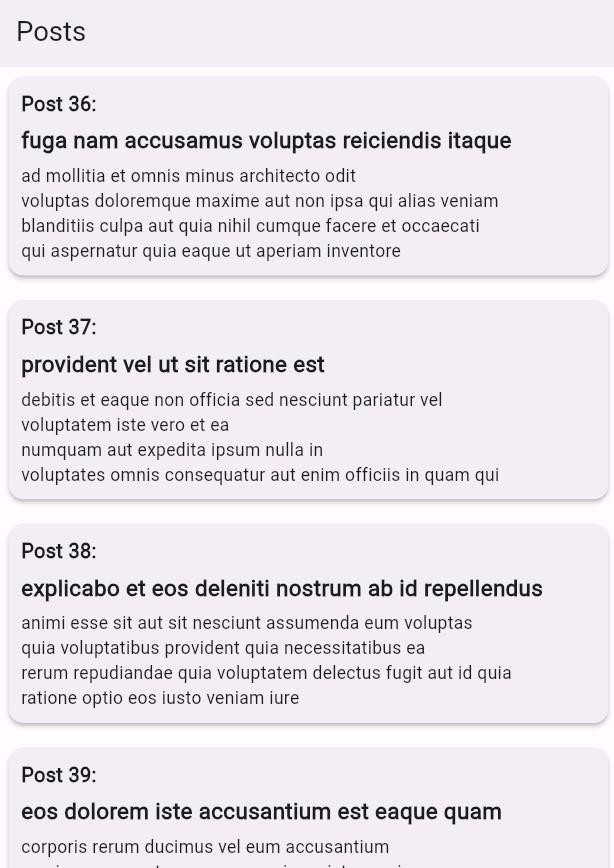
),

);

}

}



**Output:**

**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'],

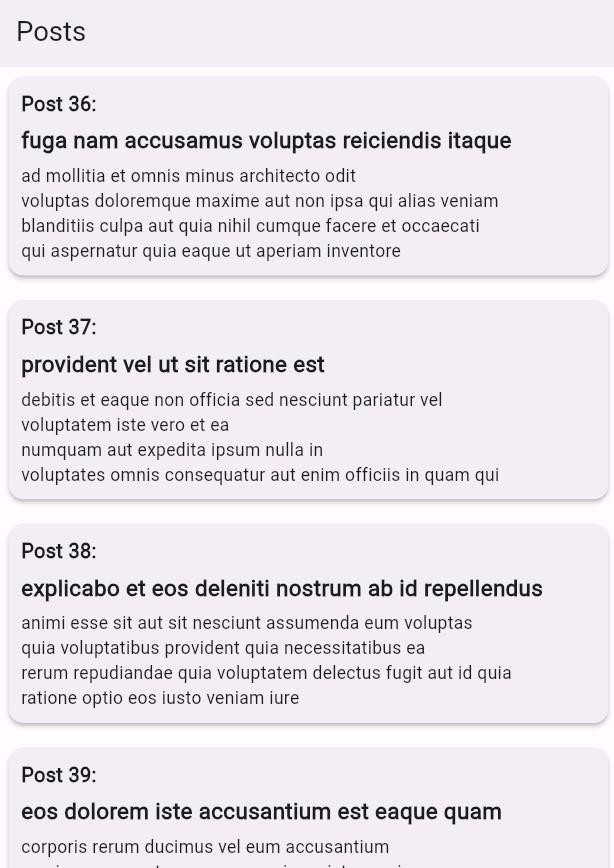
);

}

}

dev\_dependencies: flutter\_test: sdk: flutter http: ^0.13.3

**Output:**



**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(

crossAxisAlignment: CrossAxisAlignment.stretch, 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:**