Experiment 1

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

# Installation Steps :-

1. Download Android Studio:

Visit the official Android Studio website.

Click on the “Download Android Studio” button. Save the downloaded file

1. Install Android Studio:

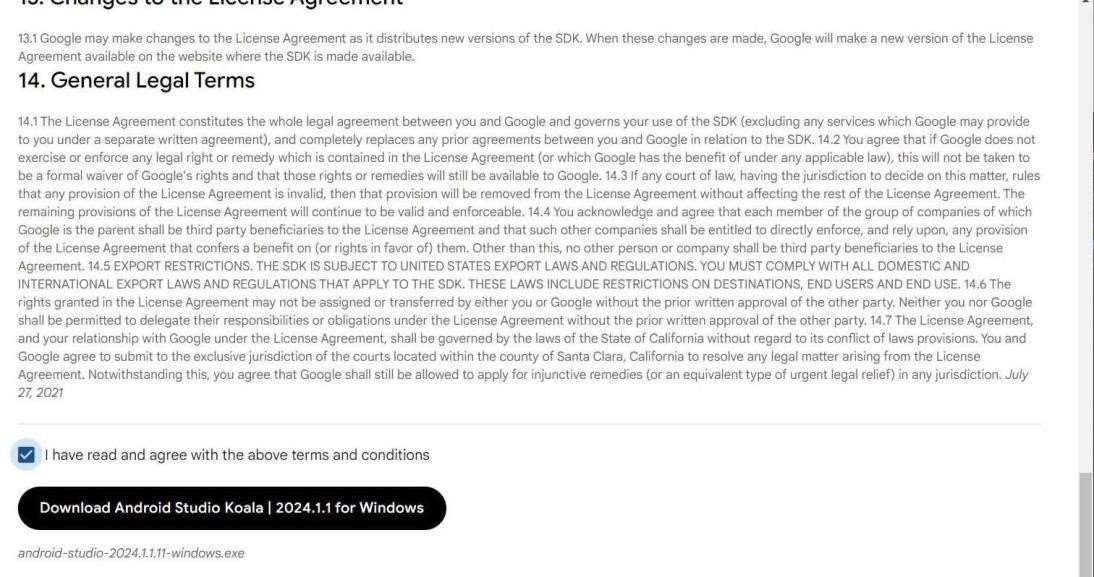
Double-click the downloaded .exe file to launch the installer.

1. Follow the Setup Wizard:

The Android Studio Setup Wizard will guide you through the installation process. Install any recommended SDK packages.

# Screenshots:

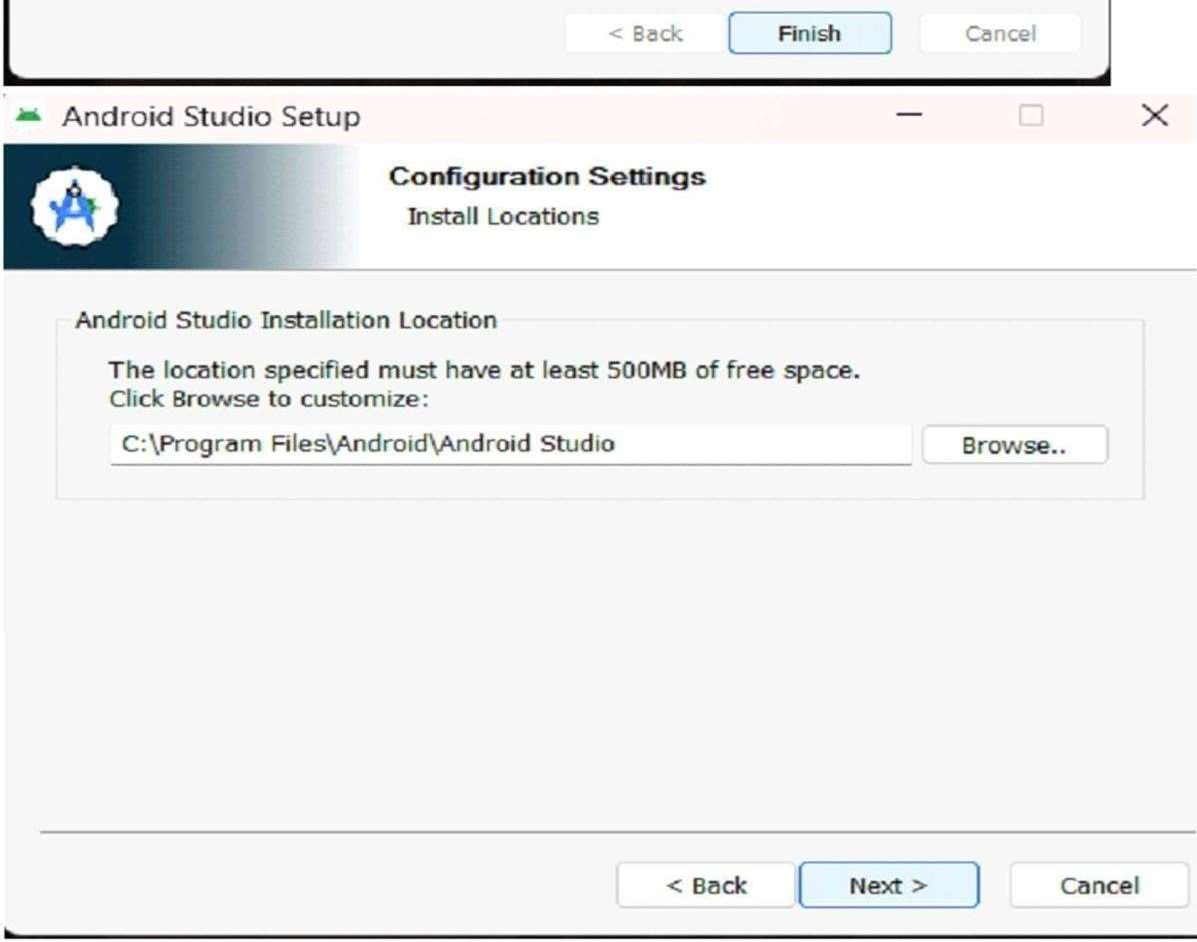
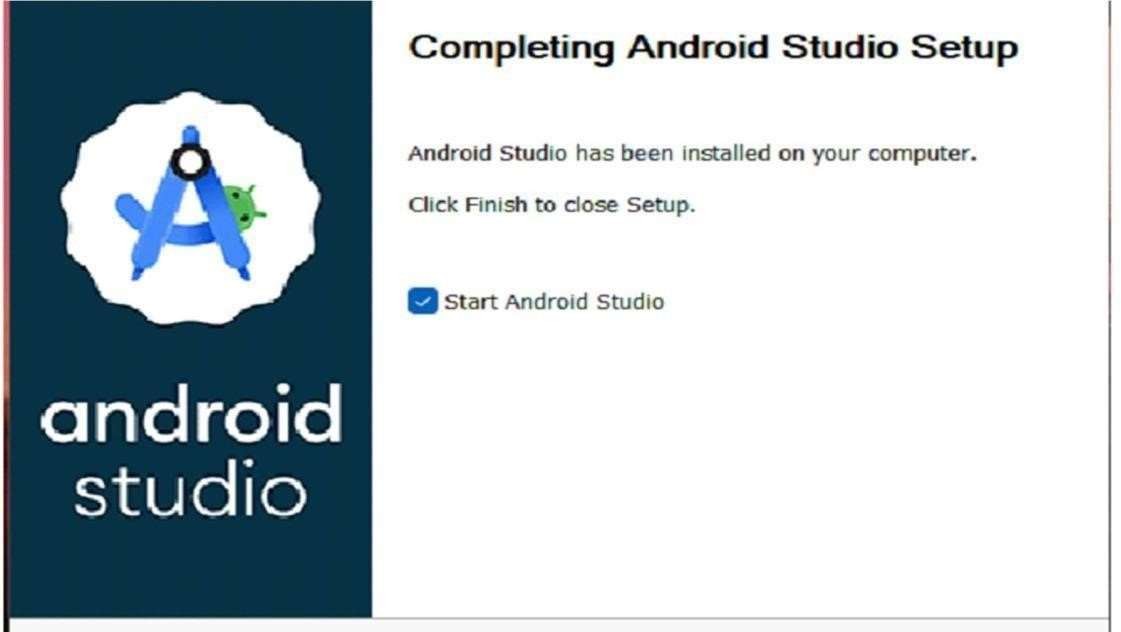
Accept terms&condtions and click download button

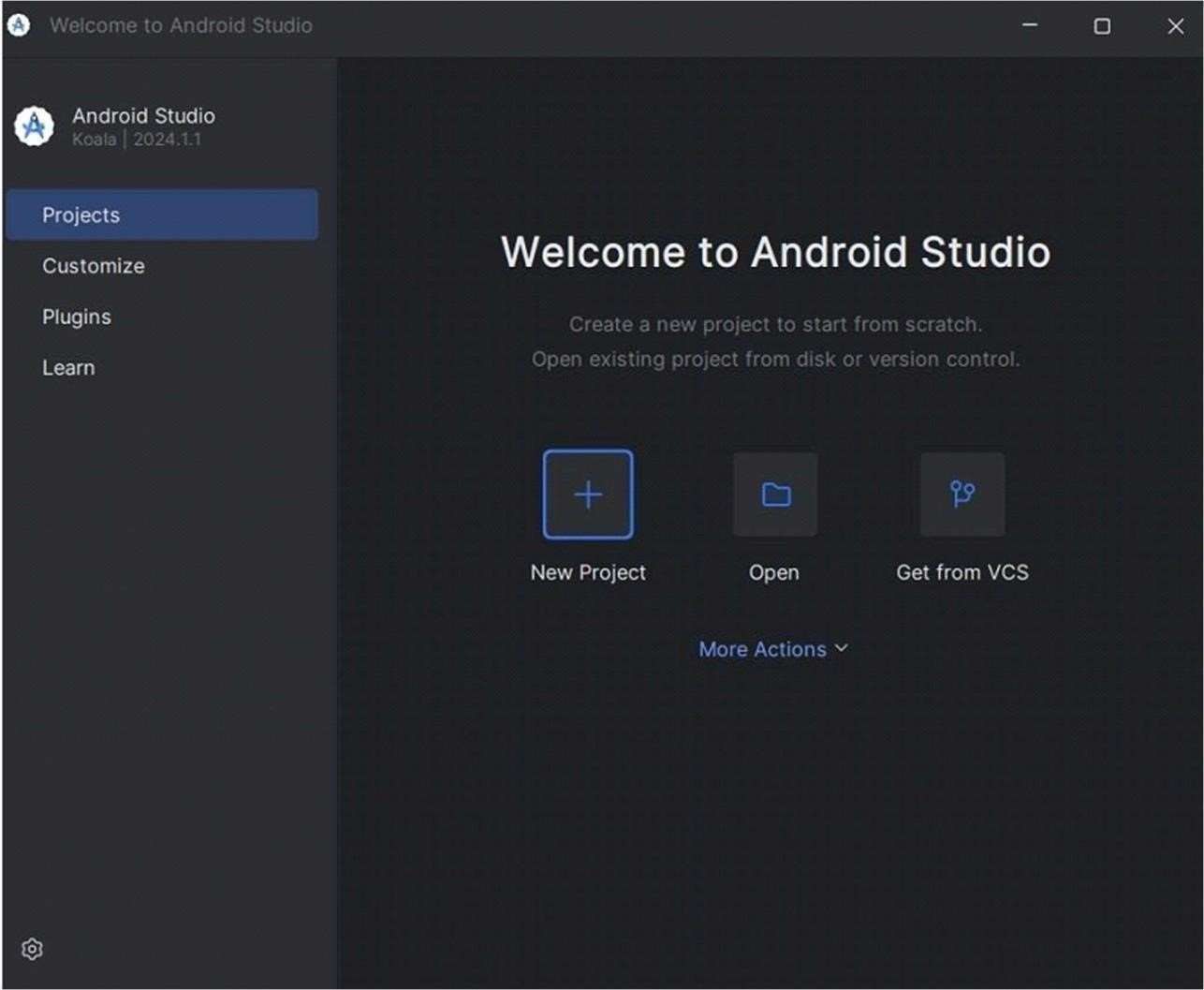


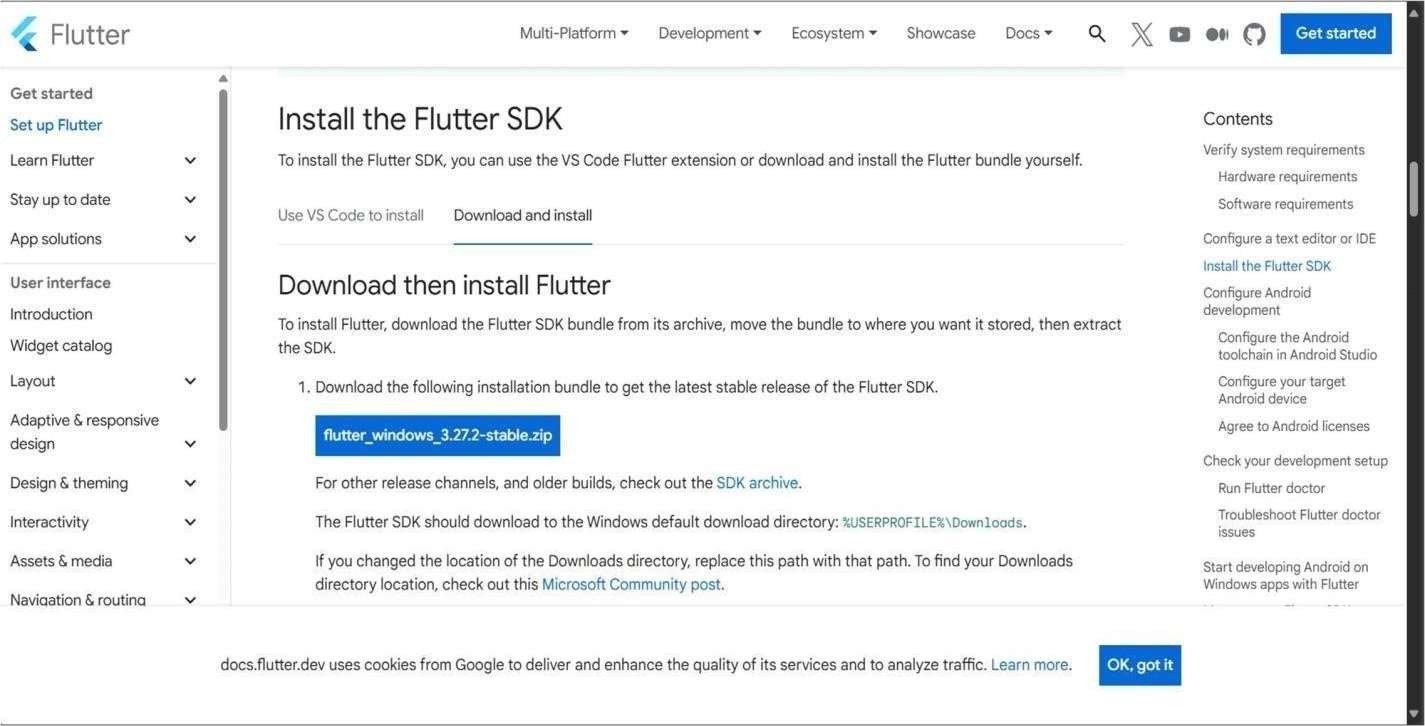
Click on next button



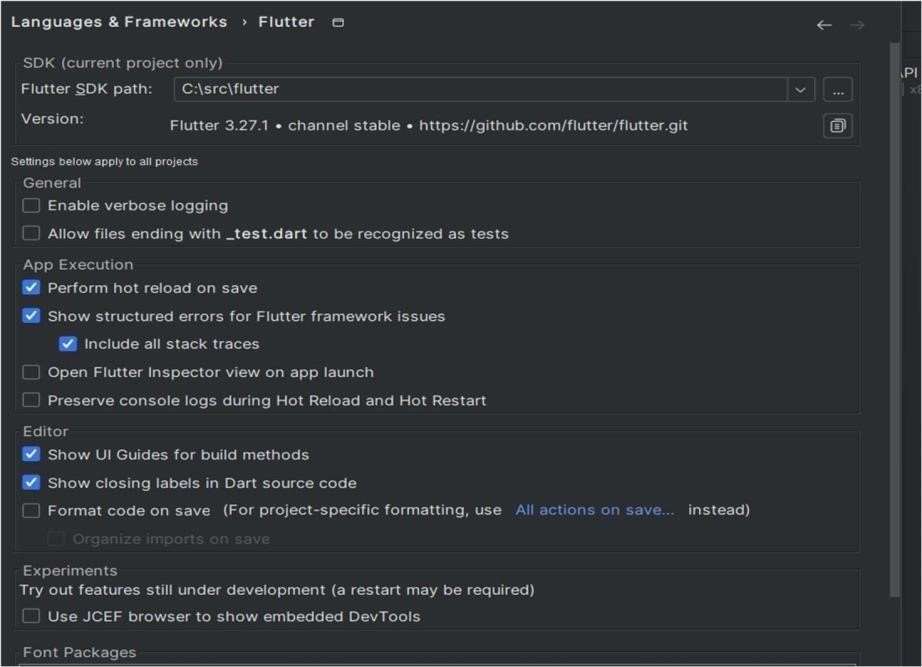
Click on finish button



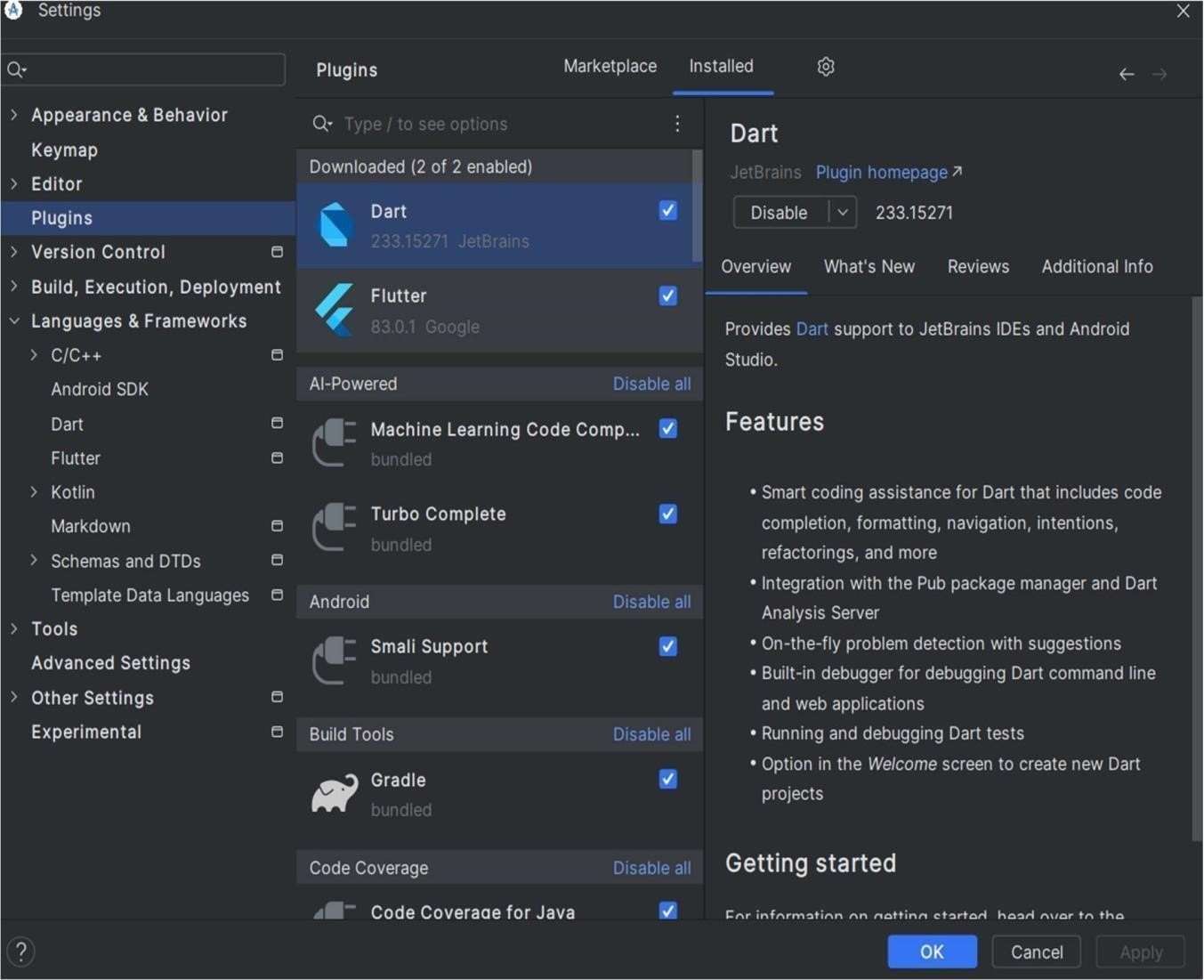


Install fluteer sdk from the flutter website from the below website [Make Android apps | Flutter]

Now set the path file in environment variables file and then go into Andriod app and add sdk path into the flutter



In the plugin download 2 plugin flutter and dart



Now apply the changes and create flutter app.

Experiment 2

**Title: Create a Hello Flutter ´application.**

**main.dart**

import 'package:flutter/material.dart';

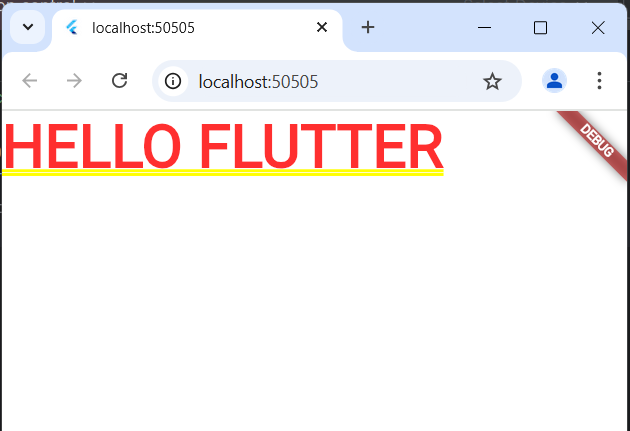
void main() { runApp(MaterialApp(

home: Text("HELLO FLUTTER")

));

}

**Output:**



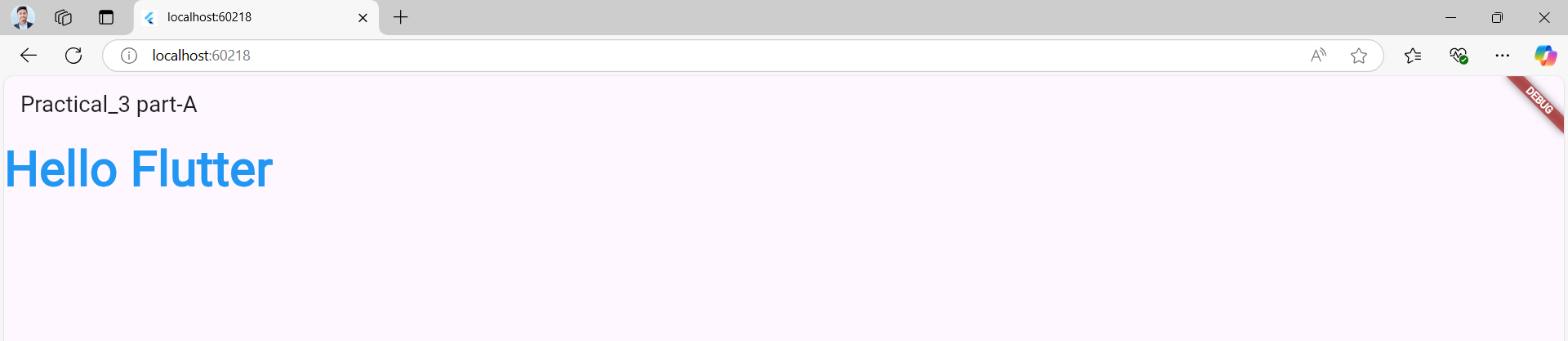
Experiment 3

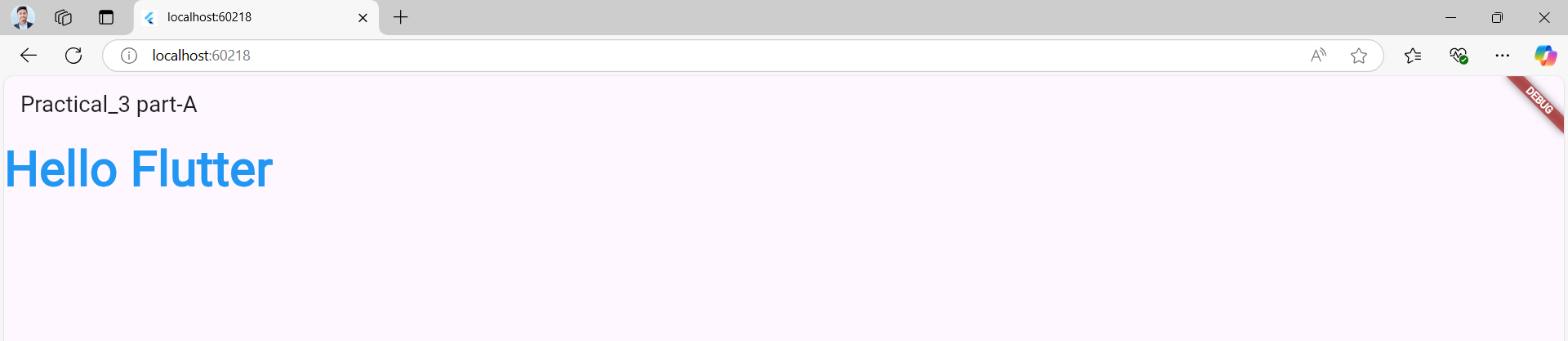
Aim: Create and application using Flutter Key Widgets.

**main.dart**

import 'package:flutter/material.dart';  
  
void main() {  
 runApp(MaterialApp(  
 home: Scaffold(  
 appBar: AppBar(  
 title: Text('Practical\_3 part-A'),  
 ),  
 body: Text(  
 "Hello Flutter",  
 style: TextStyle(  
 fontSize: 48,  
 fontWeight: FontWeight.*bold*,  
 color: Colors.*blue*,  
 overflow: TextOverflow.ellipsis,  
 ),  
 ),  
 ),  
 ));  
}

**Output:**

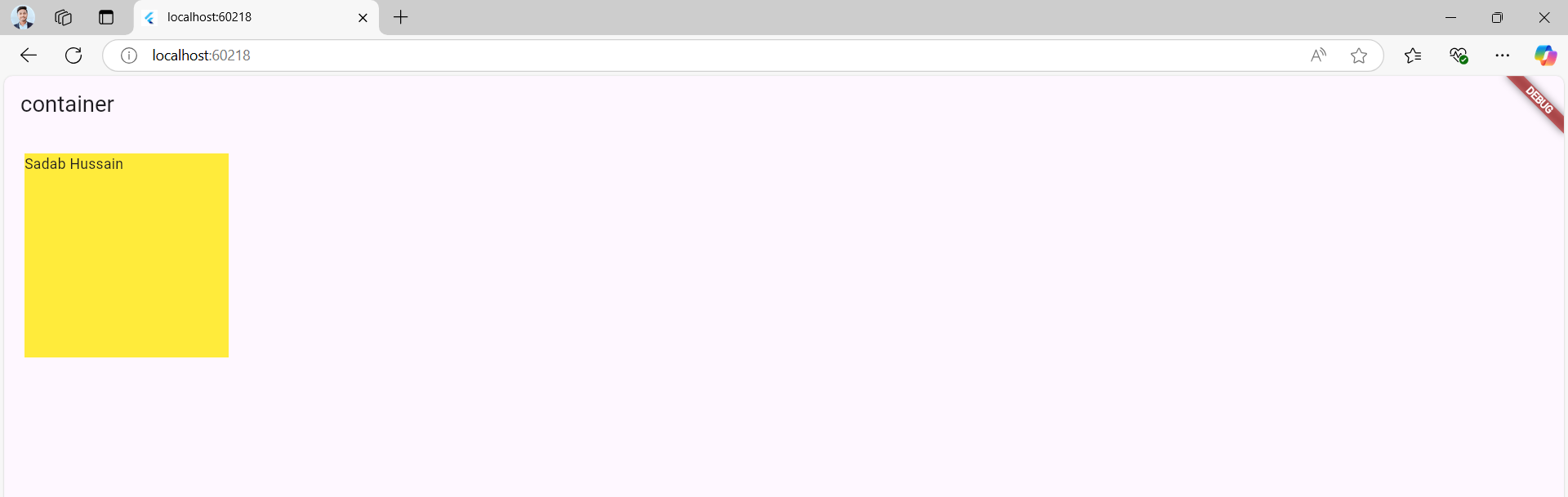
****



**main.dart**

import 'package:flutter/material.dart';  
  
void main() {  
 runApp(MaterialApp(  
 home: Scaffold(  
 appBar: AppBar(  
 title: Text("container"),  
 ),  
 body: Container(  
 height: 200.0,  
 width: 200.0,  
 margin: EdgeInsets.all(20.0),  
 color: Colors.yellow,  
 child: Text("Sadab Hussain"),  
 ),  
 ),  
 ));  
}

**Output:**



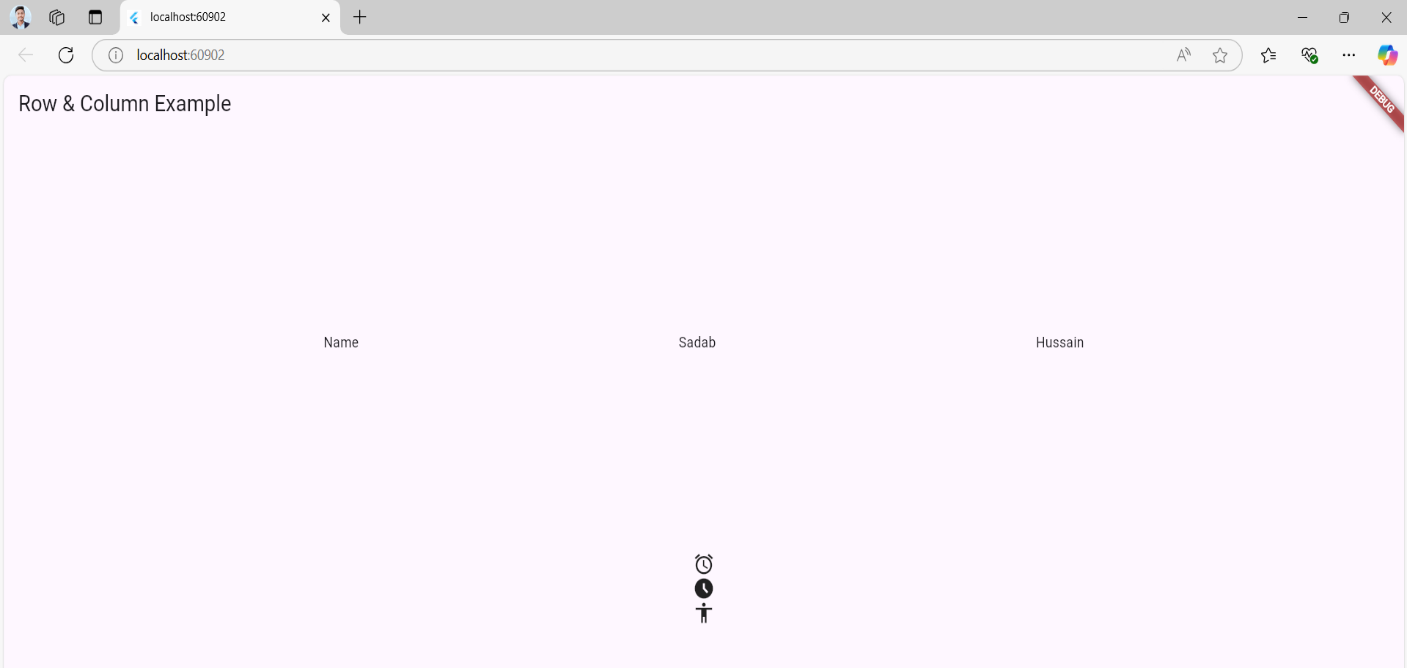
Experiment 4

Aim: Create and application using Flutter Key Widgets.

**main.dart**

import 'package:flutter/material.dart';  
  
void main() {  
 runApp(MaterialApp(  
 home: Scaffold(  
 appBar: AppBar(  
 title: Text("Row & Column Example"),  
 ),  
 body: Column(  
 mainAxisAlignment: MainAxisAlignment.spaceEvenly,  
 crossAxisAlignment: CrossAxisAlignment.center,  
 children: [  
 Row(  
 mainAxisAlignment: MainAxisAlignment.spaceEvenly,  
 crossAxisAlignment: CrossAxisAlignment.center,  
 mainAxisSize: MainAxisSize.max,  
 textDirection: TextDirection.ltr,  
 children: [  
 Text("Name"),  
 Text("Sadab"),  
 Text("Hussain"),  
 ],  
 ),  
 Column(  
 mainAxisAlignment: MainAxisAlignment.spaceEvenly,  
 crossAxisAlignment: CrossAxisAlignment.start,  
 children: [  
 Icon(Icons.access\_alarm\_outlined),  
 Icon(Icons.access\_time\_filled),  
 Icon(Icons.accessibility\_outlined),  
 ],  
 ),  
 ],  
 ),  
 ),  
 ));  
}

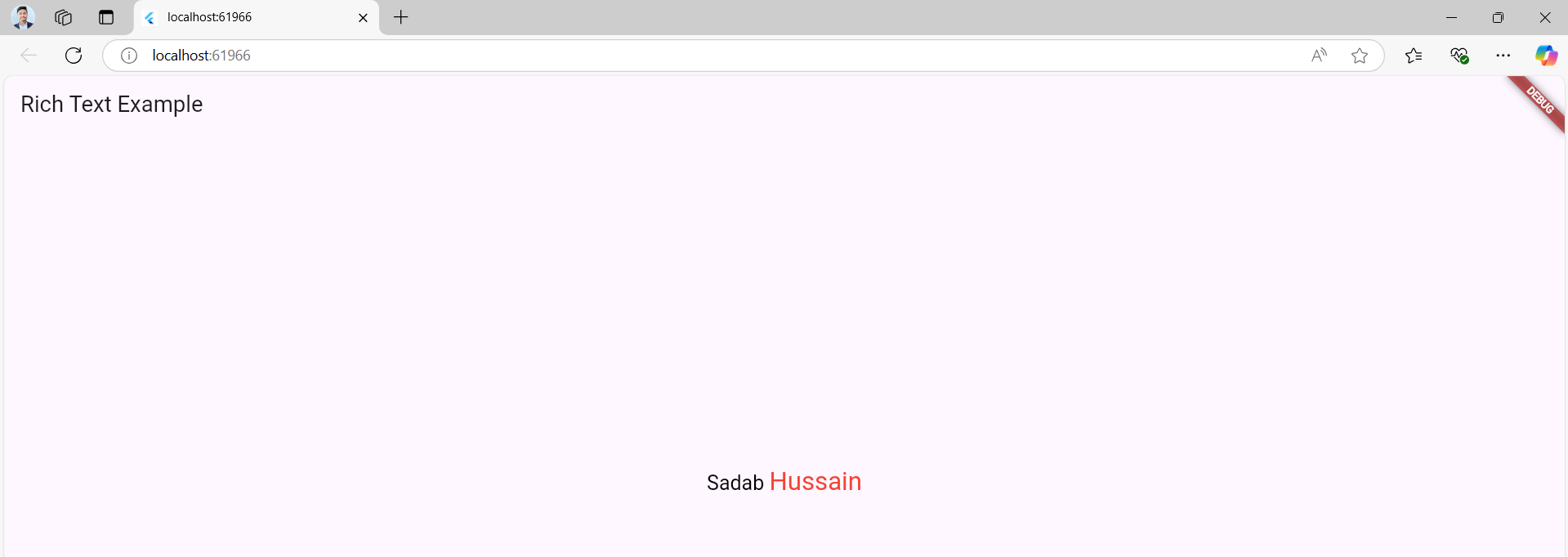
**Output:**



**main.dart**

import 'package:flutter/material.dart';  
  
void main() {  
 runApp(MaterialApp(  
 home: Scaffold(  
 appBar: AppBar(  
 title: Text("Rich Text Example"),  
 ),  
 body: Center(  
 child: RichText(  
 text: TextSpan(  
 text: "Sadab ",  
 style: TextStyle(color: Colors.*black*, fontSize: 20),  
 children: [  
 TextSpan(  
 text: "Hussain",  
 style: TextStyle(color: Colors.*red*, fontSize: 25),  
 ),  
 ],  
 ),  
 ),  
 ),  
 ),  
 ));  
}

**Output:**



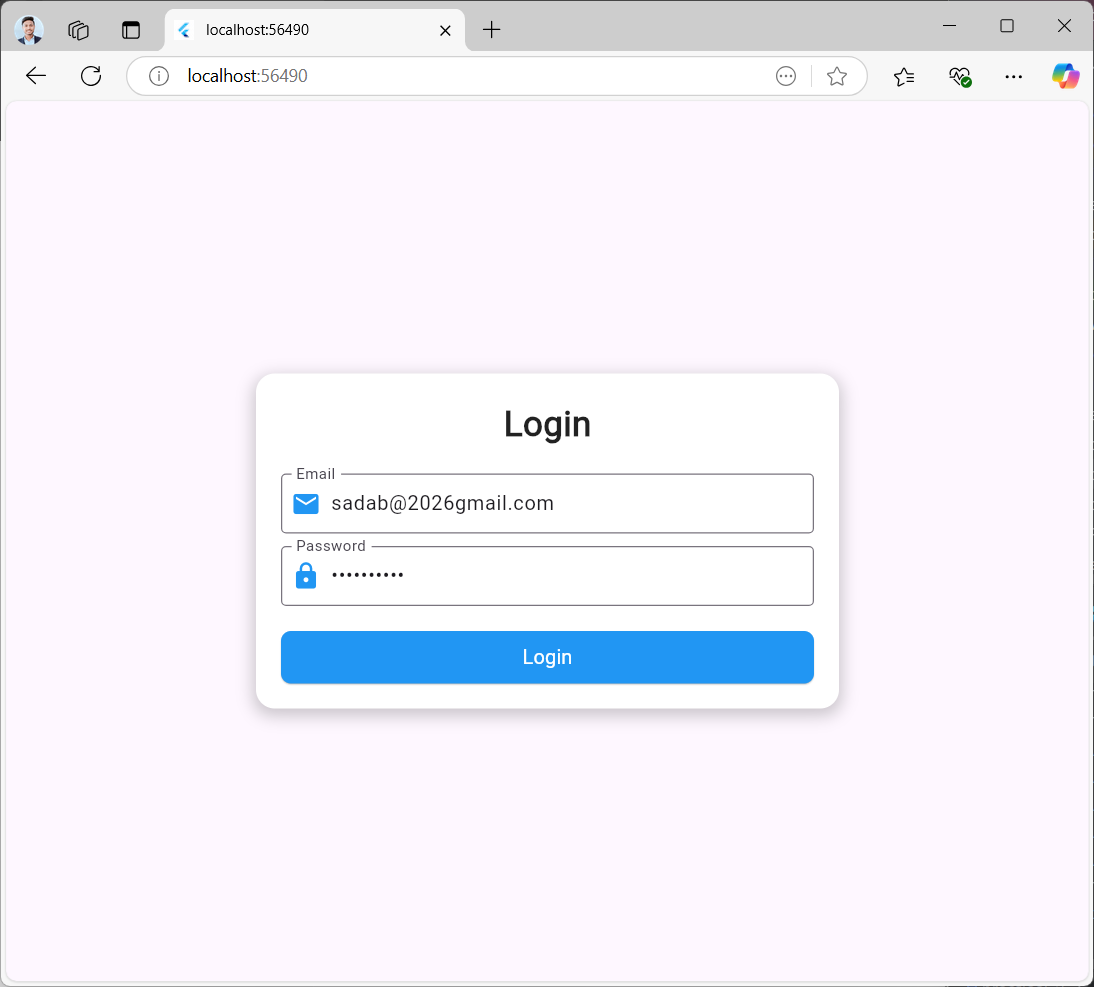
Experiment 5

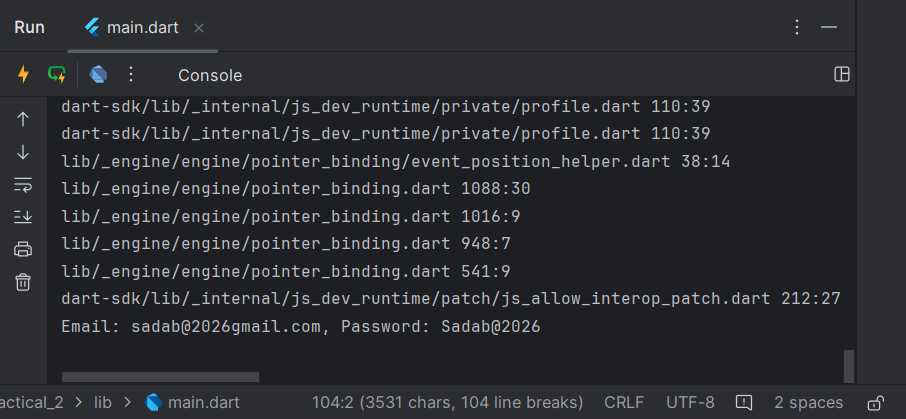
Aim: Create and application with Flutter UI Components.

**main.dart**

import 'package:flutter/material.dart';  
  
void main() {  
 runApp(const MyApp());  
}  
  
class MyApp extends StatelessWidget {  
 const MyApp({super.key});  
  
 @override  
 Widget build(BuildContext context) {  
 return MaterialApp(  
 debugShowCheckedModeBanner: false,  
 home: LoginScreen(),  
 );  
 }  
}  
  
class LoginScreen extends StatelessWidget {  
 final TextEditingController emailController = TextEditingController();  
 final TextEditingController passwordController = TextEditingController();  
  
 @override  
 Widget build(BuildContext context) {  
 return Scaffold(  
 body: Container(  
 width: double.*infinity*,  
 child: Center(  
 child: Padding(  
 padding: const EdgeInsets.all(200.0),  
 child: Container(  
 padding: const EdgeInsets.all(20.0),  
 decoration: BoxDecoration(  
 color: Colors.*white*, // Login box background color  
 borderRadius: BorderRadius.circular(15),  
 boxShadow: [  
 BoxShadow(  
 color: Colors.*black*.withOpacity(0.2),  
 blurRadius: 10,  
 spreadRadius: 2,  
 offset: const Offset(0, 4),  
 ),  
 ],  
 ),  
 child: Column(  
 mainAxisSize: MainAxisSize.min,  
 children: [  
 const Text(  
 'Login',  
 style: TextStyle(  
 fontSize: 28,  
 fontWeight: FontWeight.*bold*,  
 color: Colors.*black87*,  
 ),  
 ),  
 const SizedBox(height: 20),  
 TextField(  
 controller: emailController,  
 decoration: const InputDecoration(  
 labelText: 'Email',  
 border: OutlineInputBorder(),  
 prefixIcon: Icon(Icons.*email*, color: Colors.*blue*),  
 ),  
 keyboardType: TextInputType.*emailAddress*,  
 ),  
 const SizedBox(height: 10),  
 TextField(  
 controller: passwordController,  
 decoration: const InputDecoration(  
 labelText: 'Password',  
 border: OutlineInputBorder(),  
 prefixIcon: Icon(Icons.*lock*, color: Colors.*blue*),  
 ),  
 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');  
 },  
 style: ElevatedButton.*styleFrom*(  
 minimumSize: const Size(double.*infinity*, 50),  
 backgroundColor: Colors.*blue*,  
 shape: RoundedRectangleBorder(  
 borderRadius: BorderRadius.circular(8),  
 ),  
 ),  
 child: const Text(  
 'Login',  
 style: TextStyle(fontSize: 16, color: Colors.*white*),  
 ),  
 ),  
 ],  
 ),  
 ),  
 ),  
 ),  
 ),  
 );  
 }  
}

**Output:**





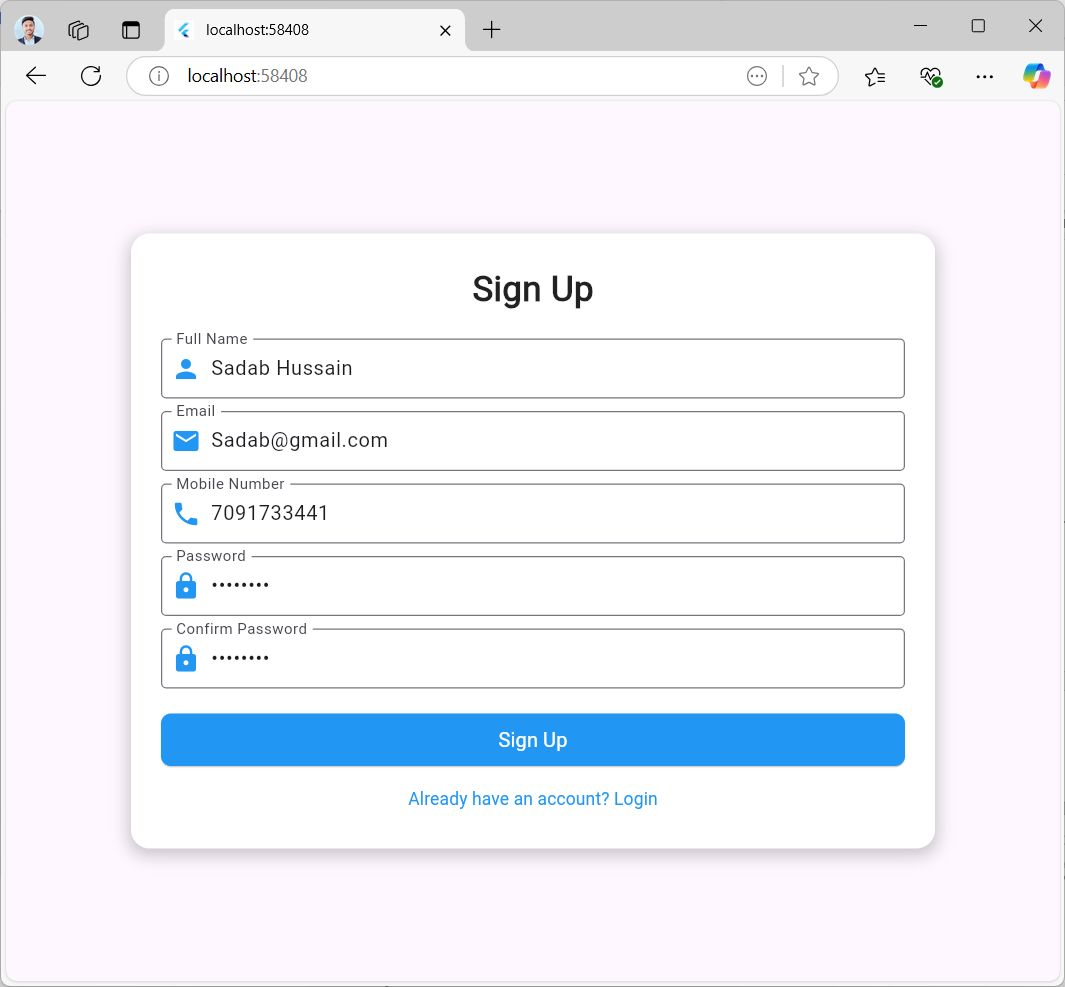
Experiment 6

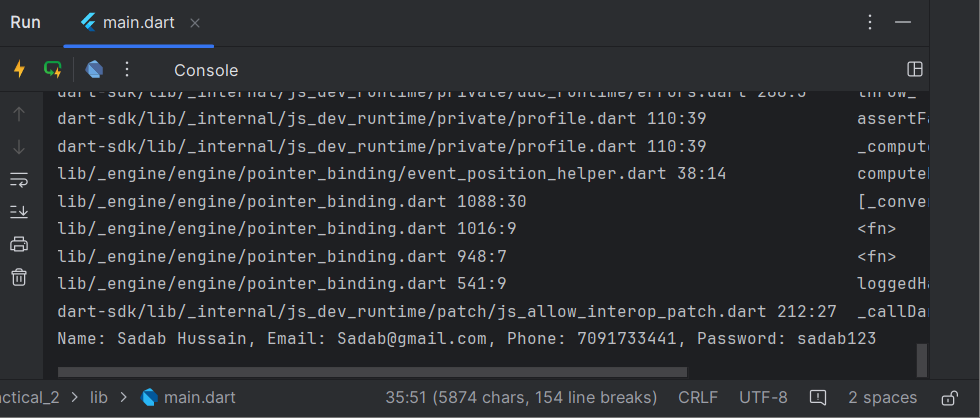
Aim: Create and application with Flutter UI Components.

**main.dart**

import 'package:flutter/material.dart';  
  
void main() {  
 runApp(const MyApp());  
}  
  
class MyApp extends StatelessWidget {  
 const MyApp({super.key});  
  
 @override  
 Widget build(BuildContext context) {  
 return MaterialApp(  
 debugShowCheckedModeBanner: false,  
 home: SignUpScreen(),  
 );  
 }  
}  
  
class SignUpScreen extends StatelessWidget {  
 final TextEditingController nameController = TextEditingController();  
 final TextEditingController emailController = TextEditingController();  
 final TextEditingController phoneController = TextEditingController();  
 final TextEditingController passwordController = TextEditingController();  
 final TextEditingController confirmPasswordController = TextEditingController();  
  
 @override  
 Widget build(BuildContext context) {  
 return Scaffold(  
 body: Container(  
 width: double.*infinity*,  
 child: Center(  
 child: Padding(  
 padding: const EdgeInsets.all(100.0),  
 child: Container(  
 padding: const EdgeInsets.all(24.0),  
 decoration: BoxDecoration(  
 color: Colors.*white*, // Form background  
 borderRadius: BorderRadius.circular(15),  
 boxShadow: [  
 BoxShadow(  
 color: Colors.*black*.withOpacity(0.2),  
 blurRadius: 10,  
 spreadRadius: 2,  
 offset: const Offset(0, 4),  
 ),  
 ],  
 ),  
 child: Column(  
 mainAxisSize: MainAxisSize.min,  
 children: [  
 const Text(  
 'Sign Up',  
 style: TextStyle(  
 fontSize: 28,  
 fontWeight: FontWeight.*bold*,  
 color: Colors.*black87*,  
 ),  
 ),  
 const SizedBox(height: 20),  
 TextField(  
 controller: nameController,  
 decoration: const InputDecoration(  
 labelText: 'Full Name',  
 border: OutlineInputBorder(),  
 prefixIcon: Icon(Icons.*person*, color: Colors.*blue*),  
 ),  
 ),  
 const SizedBox(height: 10),  
 TextField(  
 controller: emailController,  
 decoration: const InputDecoration(  
 labelText: 'Email',  
 border: OutlineInputBorder(),  
 prefixIcon: Icon(Icons.*email*, color: Colors.*blue*),  
 ),  
 keyboardType: TextInputType.*emailAddress*,  
 ),  
 const SizedBox(height: 10),  
 TextField(  
 controller: phoneController,  
 decoration: const InputDecoration(  
 labelText: 'Mobile Number',  
 border: OutlineInputBorder(),  
 prefixIcon: Icon(Icons.*phone*, color: Colors.*blue*),  
 ),  
 keyboardType: TextInputType.*phone*,  
 ),  
 const SizedBox(height: 10),  
 TextField(  
 controller: passwordController,  
 decoration: const InputDecoration(  
 labelText: 'Password',  
 border: OutlineInputBorder(),  
 prefixIcon: Icon(Icons.*lock*, color: Colors.*blue*),  
 ),  
 obscureText: true,  
 ),  
 const SizedBox(height: 10),  
 TextField(  
 controller: confirmPasswordController,  
 decoration: const InputDecoration(  
 labelText: 'Confirm Password',  
 border: OutlineInputBorder(),  
 prefixIcon: Icon(Icons.*lock*, color: Colors.*blue*),  
 ),  
 obscureText: true,  
 ),  
 const SizedBox(height: 20),  
 ElevatedButton(  
 onPressed: () {  
 String name = nameController.text;  
 String email = emailController.text;  
 String phone = phoneController.text;  
 String password = passwordController.text;  
 String confirmPassword = confirmPasswordController.text;  
  
 // Sign-up validation  
 if (password == confirmPassword) {  
 print('Name: $name, Email: $email, Phone: $phone, Password: $password');  
 } else {  
 print('Passwords do not match');  
 }  
 },  
 style: ElevatedButton.*styleFrom*(  
 minimumSize: const Size(double.*infinity*, 50),  
 backgroundColor: Colors.*blue*,  
 shape: RoundedRectangleBorder(  
 borderRadius: BorderRadius.circular(8),  
 ),  
 ),  
 child: const Text(  
 'Sign Up',  
 style: TextStyle(fontSize: 16, color: Colors.*white*),  
 ),  
 ),  
 const SizedBox(height: 10),  
 TextButton(  
 onPressed: () {  
 Navigator.*pop*(context); // Navigate back to the Login screen  
 },  
 child: const Text(  
 'Already have an account? Login',  
 style: TextStyle(color: Colors.*blue*, fontSize: 14),  
 ),  
 ),  
 ],  
 ),  
 ),  
 ),  
 ),  
 ),  
 );  
 }  
}

**Output:**





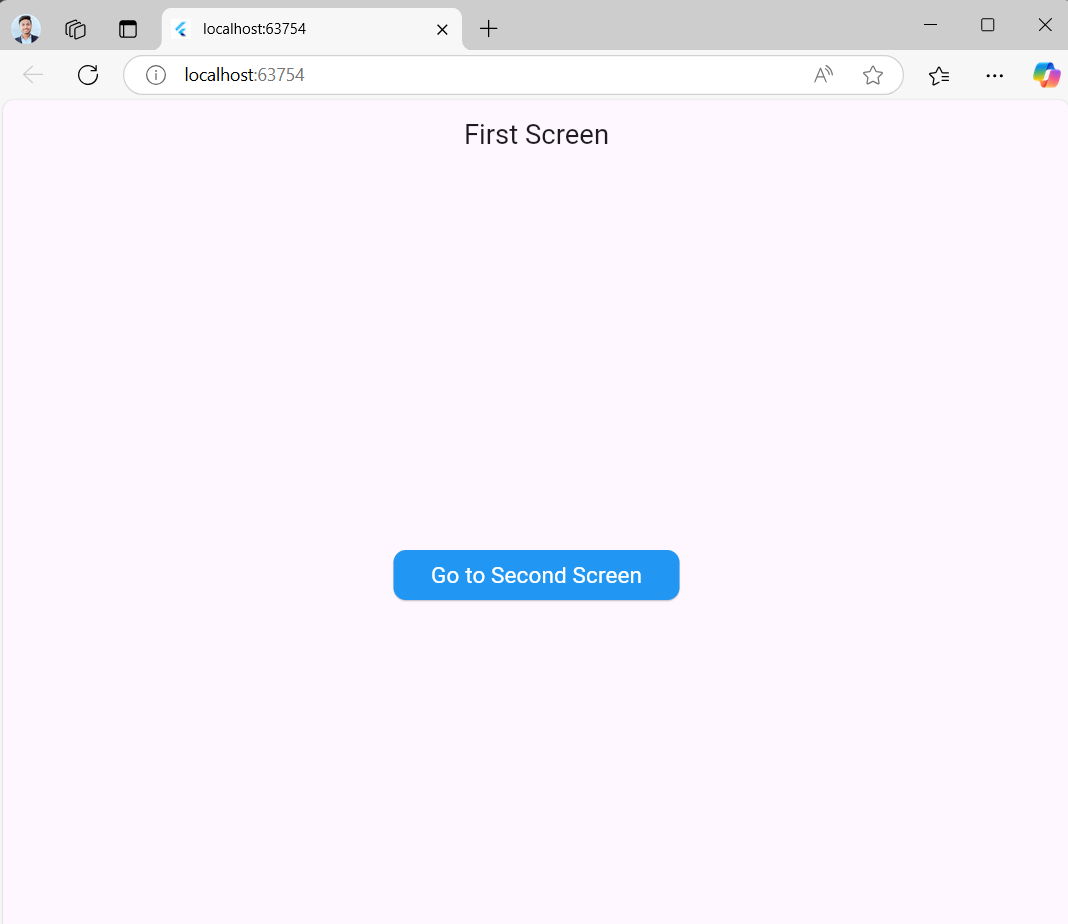
Experiment 7

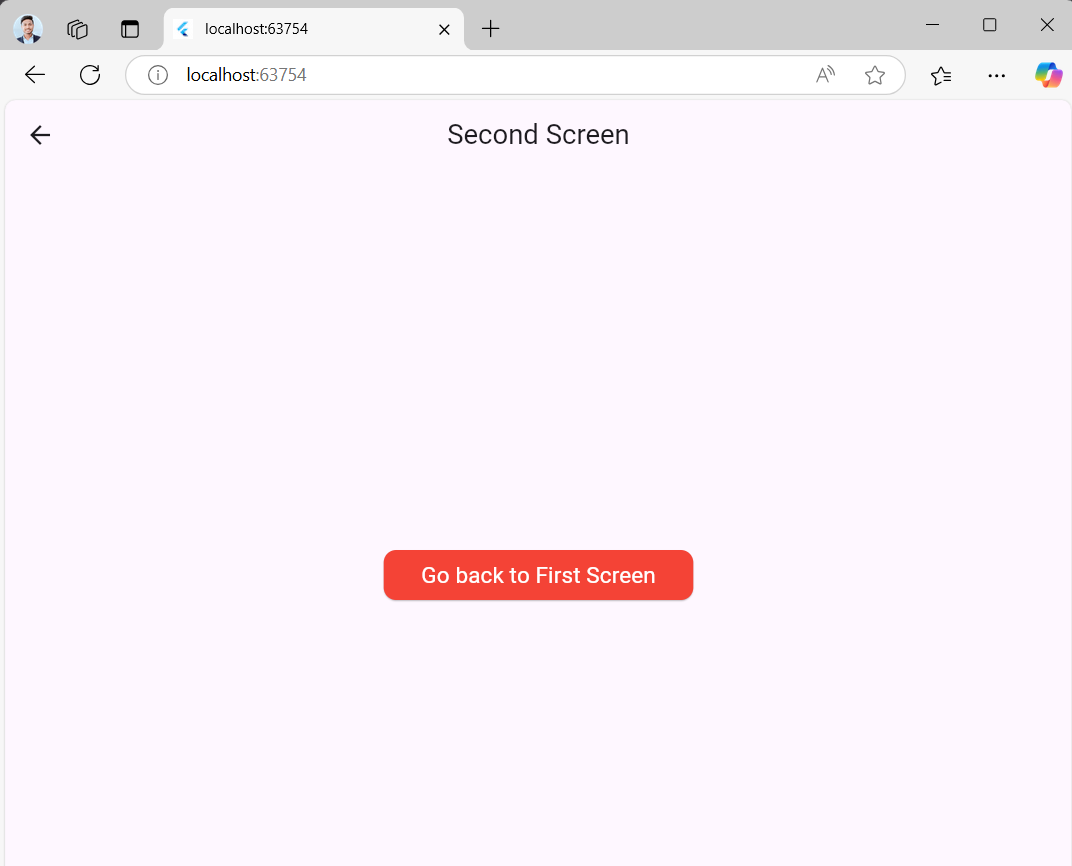
Aim: Create and application with Navigation in Flutter.

**main.dart**

import 'package:flutter/material.dart';  
  
void main() {  
 runApp(MyApp());  
}  
  
class MyApp extends StatelessWidget {  
 @override  
 Widget build(BuildContext context) {  
 return MaterialApp(  
 debugShowCheckedModeBanner: false,  
 theme: ThemeData(  
 primarySwatch: Colors.*blue*,  
 ),  
 home: FirstScreen(),  
 );  
 }  
}  
  
class FirstScreen extends StatelessWidget {  
 @override  
 Widget build(BuildContext context) {  
 return Scaffold(  
 appBar: AppBar(  
 title: Text('First Screen'),  
 centerTitle: true,  
 ),  
 body: Center(  
 child: ElevatedButton(  
 style: ElevatedButton.*styleFrom*(  
 padding: EdgeInsets.symmetric(horizontal: 30, vertical: 15),  
 backgroundColor: Colors.*blue*,  
 shape: RoundedRectangleBorder(  
 borderRadius: BorderRadius.circular(10),  
 ),  
 ),  
 onPressed: () {  
 Navigator.*push*(  
 context,  
 MaterialPageRoute(builder: (context) => SecondScreen()),  
 );  
 },  
 child: Text(  
 'Go to Second Screen',  
 style: TextStyle(fontSize: 18, color: Colors.*white*),  
 ),  
 ),  
 ),  
 );  
 }  
}  
  
class SecondScreen extends StatelessWidget {  
 @override  
 Widget build(BuildContext context) {  
 return Scaffold(  
 appBar: AppBar(  
 title: Text('Second Screen'),  
 centerTitle: true,  
 ),  
 body: Center(  
 child: ElevatedButton(  
 style: ElevatedButton.*styleFrom*(  
 padding: EdgeInsets.symmetric(horizontal: 30, vertical: 15),  
 backgroundColor: Colors.*red*,  
 shape: RoundedRectangleBorder(  
 borderRadius: BorderRadius.circular(10),  
 ),  
 ),  
 onPressed: () {  
 Navigator.*pop*(context);  
 },  
 child: Text(  
 'Go back to First Screen',  
 style: TextStyle(fontSize: 18, color: Colors.*white*),  
 ),  
 ),  
 ),  
 );  
 }  
}

**Output:**





Experiment 8

Aim: Create and application with list view in Flutter.

**main.dart**

import 'package:flutter/material.dart';  
  
void main() {  
 runApp(MyApp());  
}  
  
class MyApp extends StatelessWidget {  
 @override  
 Widget build(BuildContext context) {  
 return MaterialApp(  
 home: Scaffold(  
 appBar: AppBar(  
 title: Text('ListView Example'),  
 ),  
 body: ListView(  
 children: <Widget>[  
 ListTile(  
 leading: Icon(Icons.*star*),  
 title: Text('Item 1'),  
 ),  
 ListTile(  
 leading: Icon(Icons.*account\_circle\_outlined*),  
 title: Text('Item 2'),  
 ),  
 ListTile(  
 leading: Icon(Icons.*adb\_rounded*),  
 title: Text('Item 3'),  
 ),  
 ListTile(  
 leading: CircleAvatar(  
 backgroundImage:  
 AssetImage('images/butterfly.png'),  
 ),  
 title: Text('Item 1'),  
 subtitle: Text('Subtitle for Item 1'),  
 onTap: () {  
 },  
 ),  
 ListTile(  
 leading: CircleAvatar(  
 child: Icon(Icons.*account\_circle\_rounded*,color: Colors.*blue*,),  
 ),  
 title: Text('Item 2'),  
 subtitle: Text('Subtitle for Item 2'),  
 onTap: () {  
 // Handle tap  
 },  
 )  
 ]  
 )  
 )  
 );  
 }  
}

**Output:**



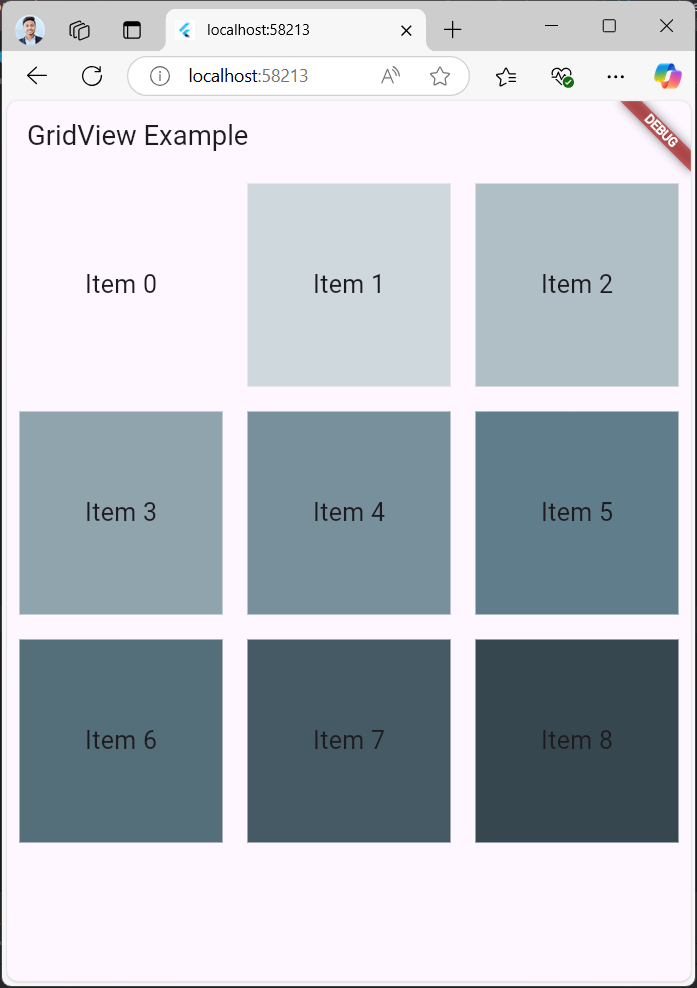
Experiment 9

Aim: Create and application with grid view in Flutter.

**main.dart**

import 'package:flutter/material.dart';  
  
void main() {  
 runApp(MyApp());  
}  
  
class MyApp extends StatelessWidget {  
 @override  
 Widget build(BuildContext context) {  
 return MaterialApp(  
 home: Scaffold(  
 appBar: AppBar(  
 title: Text('GridView Example'),  
 ),  
 body: GridView.count(  
 crossAxisCount: 3,  
 children: List.generate(9, (index) {  
 return Container(  
 margin: EdgeInsets.all(10.0),  
 color: Colors.*blueGrey*[100 \* (index % 9)],  
 child: Center(  
 child: Text(  
 'Item $index',  
 style: TextStyle(fontSize: 20),  
 ),  
 ),  
 );  
 }),  
 ),  
 ),  
 );  
 }  
}

**Output:**

****

Experiment 10

Aim: 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("""CREATE

TABLE items(

id INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,

title TEXT, description TEXT,

createdAt TIMESTAMP NOT NULL DEFAULT CURRENT\_TIMESTAMP

)

""");

}

// id: the id of a item

// title, description: name and description of your activity

// created\_at: the time that the item was created. It will be automatically handled by SQLite

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

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

await createTables(database);

},

);

}

// Create new item (journal)

static Future<int> createItem(String title, String? descrption) async {

final db = await SQLHelper.db();

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

}

// Read all items (journals)

static Future<List<Map<String, dynamic>>> getItems() async {

final db = await SQLHelper.db(); return db.query('items', orderBy: "id");

}

// Read a single item by id

// The app doesn't use this method but I put here in case you want to see it static Future<List<Map<String, dynamic>>> getItem(int id) async {

final db = await SQLHelper.db();

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

}

// Update an item by id

static Future<int> updateItem(

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

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

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

};

final result =

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

}

// Delete

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

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

} catch (err) {

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

}

}

}

dependencies: flutter:

sdk: flutter sqflite: ^2.0.0

path: ^1.9.0

path\_provider: any

**Output:**



Experiment 11

Aim: 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();

},

),

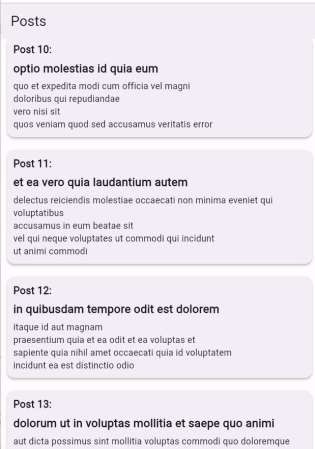
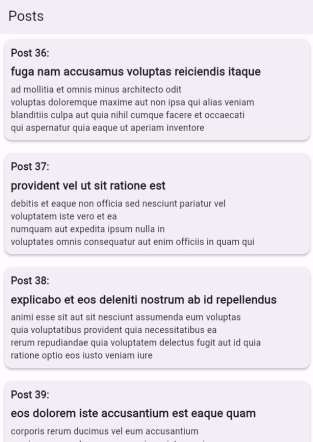
),

);

}

}

**Output:**



Experiment 12

Aim: 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'],

);

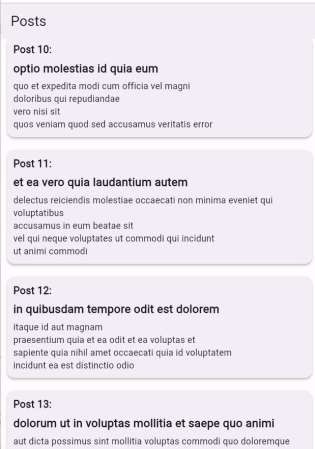
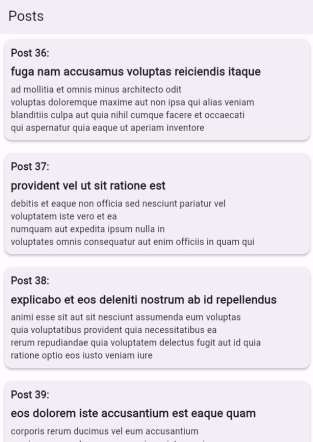
}

}

dev\_dependencies: flutter\_test:

sdk: flutter http: ^0.13.3

**Output:**



Experiment 13

Aim: 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:**

