

**SEB-312 Mobile Application Development**

**LAB # 01**

**LAB Title**

|  |
| --- |
| Install Visual Studio Code and also install flutter and dart extensions. Create a first application of Hello World. Exploring project structure and main files. Create a basic counter app on Flutter. Writing simple Dart programs. |

**Assessment of CLO: 03, PLO: 05**

|  |  |  |  |
| --- | --- | --- | --- |
| **Student Name:** | Abdullah Mohsin | | |
| **Roll No.** | 23fa-048-st | | |
| **Semester** | 3rd | **Session** | SPRING-2025 |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **S. No.** | **Perf. Level**  **Criteria** | **Excellent**  **(2.5)** | **Good**  **(2)** | **Satisfactory**  **(1.5)** | **Needs Improvement**  **(0 ~ 1)** | **Marks Obtained** |
| **1** | Project Execution & Implementation | Fully functional, optimized, and well-structured. | Minor errors, mostly functional. | Some errors, requires guidance. | Major errors, non-functional, or not Performed. |  |
| **2** | Results & Debugging  Or Troubleshooting | Accurate results with effective debugging  Or Troubleshooting. | Mostly correct, some debugging Or Troubleshooting needed. | Partial results, minimal debugging  Or Troubleshooting. | Incorrect results, no debugging Or Troubleshooting, or not attempted. |  |
| **3** | Problem-Solving & Adaptability  (VIVA) | Creative approach, efficiently solves challenges. | Adapts well, minor struggles. | Some adaptability, needs guidance. | Lacks innovation or no innovation, unable to solve problems. |  |
| **4** | Report Quality & Documentation | Clear, structured, with detailed visuals. | Mostly clear, minor gaps. | Some clarity issues, missing details. | Poorly structured, lacks clarity, or not submitted. |  |
| **Total Marks Obtained Out of 10** | | | | | |  |

**Experiment evaluated by**

|  |  |  |  |
| --- | --- | --- | --- |
| **Instructor’s Name** | **Sidra Khatoon** | | |
| **Date** |  | **Signature** |  |

# **Objective**

The purpose of this lab session is installing visual studio code. Also install Flutter extension is VS code. Create a first application of Hello World.

**Assessments:**

1. What is the role of the Flutter extension in Visual Studio Code, and how does it enhance the development experience for Flutter applications?

**Ans:**

The Flutter extension in Visual Studio Code (VS Code) helps make Flutter app development easier with these key features:

1. **Flutter SDK Integration**: Run Flutter commands (like flutter run or flutter doctor) directly in VS Code.
2. **Code Assistance**: It provides syntax highlighting, autocompletion, and IntelliSense for Flutter and Dart code.
3. **Hot Reload & Hot Restart**: Instantly update the UI without restarting the app, speeding up development.
4. **Widget Inspector**: Visualize and debug the widget tree to design UIs more efficiently.
5. **Project Creation & Management**: Easily create new Flutter projects and manage dependencies.
6. How does the debugging functionality of the Flutter extension in Visual Studio Code improve the efficiency of troubleshooting Flutter applications?

**Ans:**

The Flutter extension makes debugging easier with these tools:

1. **Breakpoints & Step Debugging**: Pause the app to check variables and logic.
2. **Error Messages & Stack Traces**: Get clear error logs to find issues quickly.
3. **Debug Console**: See logs, print statements, and Flutter messages in real-time.
4. **Hot Reload**: Make changes to UI or code without restarting the app.
5. **Performance Profiling**: Find slow widgets and other performance problems.
6. Create a simple Flutter app that displays "Your introduction and your hobbies" on the screen. Submit screen shot of output.

**Ans:**

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: Scaffold(

backgroundColor: Color(0xFFFFF0F5), // Light pink background

body: Center(

child: Column(

mainAxisAlignment: MainAxisAlignment.center,

children: const [

Text(

"My name is Abdullah Mohsin",

style: TextStyle(fontSize: 20, fontWeight: FontWeight.bold),

),

SizedBox(height: 5),

Text(

"I love to read tech-related books in my free time",

style: TextStyle(fontSize: 18),

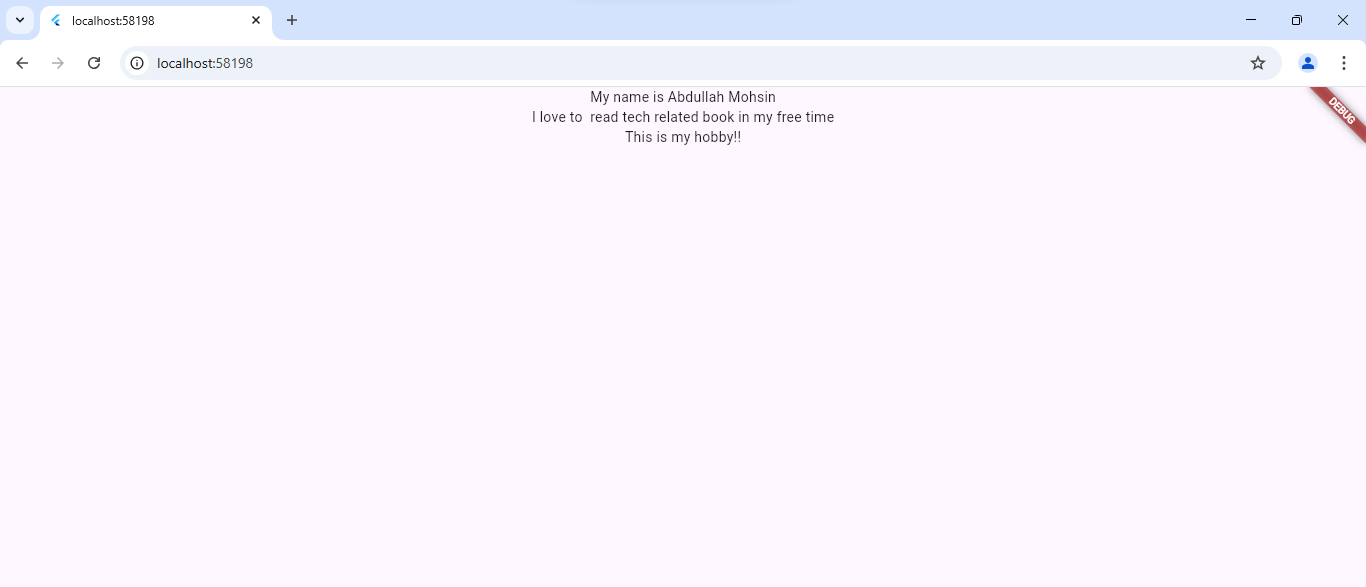
),

SizedBox(height: 5),

Text(

"This is my hobby!!",

style: TextStyle(fontSize: 18), ), ], ), ),), );}}



1. Create counter application that we created in lab. Your task is adding another button for decrement the counter values. Add floating button for decrement.

**Ans:**

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 Demo',

theme: ThemeData(

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

useMaterial3: true, ),

home: const MyHomePage(title: 'Flutter Demo Home Page'), ); }}

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++; });}

void \_decrementCounter() {

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(

'You have pushed the button this many times:',),

Text(

'$\_counter',

style: Theme.of(context).textTheme.headlineMedium, ), ], ), ),

floatingActionButton: Row(

mainAxisAlignment: MainAxisAlignment.end,

children: [

FloatingActionButton(

onPressed: \_incrementCounter,

tooltip: 'Increment',

child: const Icon(Icons.add),),

const SizedBox(width: 10), // Space between buttons

FloatingActionButton(

onPressed: \_decrementCounter,

tooltip: 'Decrement',

child: const Icon(Icons.remove), ),], ), ); }}

