Lab Report

Course Code: CSE 335

Course Name: Pervasive Computing and Mobile App Development Lab

Lab Report No: 02

Lab Report Name : Basics of Flutter

Date of submission: 17-03-2024

Submitted To :

Tasmim Sultana

Lecturer

Department of CSE

Daffodil International University

Submitted By:

Name: Rayhan Rafin

ID: 213-15-4278

Section: 60_B1

Flutter

Flutter is an open-source UI framework created by Google for building beautiful, fast, and cross-platform applications. It allows you to write code once and deploy it to various platforms, including mobile (Android, iOS), web, desktop (Windows, macOS, Linux), and even embedded devices.

Key Features

- **Cross-Platform Development:** Develop for multiple platforms with a single codebase, saving time and resources.
- **Fast Development:** Hot reload allows for near-instantaneous updates to your app as you code, streamlining the development process.
- **Beautiful UIs:** Flutter's rich set of widgets and customization options enable you to create stunning, native-looking user interfaces.
- **High Performance:** Flutter apps are known for their smooth performance and responsiveness, thanks to its use of widgets and a layered rendering architecture.
- **Declarative Style:** Define how your app looks and behaves instead of imperatively controlling every step, making code more maintainable and readable.
- **Rich Ecosystem:** Benefit from a vast library of community-developed packages that extend Flutter's functionality for various use cases.

Activities/Widgets in Flutter

Flutter applications consist primarily of widgets, which are reusable UI building blocks that represent a part of the screen. Widgets can be simple (like a Text widget displaying text) or complex (like a custom button). You can combine widgets to create more intricate UI elements and layouts.

Flutter Source Code Flow: The entry point of the Flutter app is the main() function where it initializes the app and calls runApp(). The runApp() function takes the given Widget and makes it the root of the widget tree.

Hot Reload

Hot reload is a powerful feature in Flutter that allows you to see changes you make to your code reflected in the running app almost instantly, without having to restart the app. This significantly speeds up the development process and helps you iterate on your UI design and functionality quickly.

Material App: Foundation for Material Design apps.

- **Functionality:**
    - Provides navigation, theming, and core app structure.
    - Wraps common Material Design widgets (AppBar, Scaffold, etc.).
- **Benefits:**
    - Easy to follow Material Design guidelines.
    - Streamlined app development process.
    - Consistent user experience across platforms.

Code Example:

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

class MyApp extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      title: 'My Flutter App',
      home: Scaffold(
        appBar: AppBar(
          title: Text('My App'),
        ),
        body: Center(
          child: Text('Hello, World!'),
        ),
      ),
    );
  }
}
```

Scaffold Class: The Scaffold class in Flutter provides the foundational layout structure for your app's screens

> **Functionality**: Provides a pre-defined, customizable layout structure for Material Design apps.
>
> **Benefits**: Simplifies app development, promotes consistency, and eases theming.

Code Example:

```
Scaffold(
  appBar: AppBar(
    title: Text('My App'),
  ),
  body: Center(
    child: Text('Hello, World!'),
  ),
);
```

Color Class: Represents a color value for UI elements.

> **Functionality**: Creates and manages colors
> **Benefits**: Simplifies color usage and consistency.

Code Example:

```
Scaffold(
  appBar: AppBar(
    title: Text('My App', style: TextStyle(color: Colors.white)),
    backgroundColor: Colors.blue
  ),
  body: Center(
    child: Text('Greetings!', style: TextStyle(color: Colors.teal)),
  ),
);
```

Text Class: Displays text on the screen with customizable styles.

Functionality: Renders text with styling options.
Benefits: Enables clear communication and visually appealing text elements.

Code Example:

```
Scaffold(
  appBar: AppBar(
    title: Text('My App'),
  body: Center(
    child: Text(
      'This is some styled text!',
      style: TextStyle(fontSize: 24.0, color: Colors.red),
    ),
  ),
);
```

Textstyle Class: Defines the visual appearance of text, including font size, color, weight, and more.

Functionality: Controls text formatting.

Benefits: Creates consistent and visually appealing text elements across your app.

Code Example:

```
body: Center(
    child: Text(
      'This text uses a custom style!',
      style: TextStyle(
        color: Colors.green,
        fontStyle: FontStyle.italic,
      ),
    ),
  ),
```

Container Class: A versatile widget that groups and styles other widgets, providing padding, margins, borders, and background decoration.

> **Functionality**: Organizes and decorates child widgets.
> **Benefits**: Simplifies layout management, enhances visual appeal, and promotes code reusability.

Code Example:

```
Scaffold(
  appBar: AppBar(
    title: Text('My App'),
  ),
  body: Center(
    child: Container(
      padding: EdgeInsets.all(20.0),
      decoration: BoxDecoration(
        color: Colors.lightBlueAccent,
        border: Border.all(color: Colors.blue),
      ),
      child: Text(
        'This text is inside a container!',
      ),
    ),
  ),
);
```

Center Class: Positions its child widget in the center of its available space.

> **Functionality**: Centers a child widget
> **Benefits**: Simplifies centering elements within a layout and promotes consistent alignment.

Code Example:

```
Scaffold(
  appBar: AppBar(
    title: Text('My App'),
  ),
  body: Center(
    child: Text(
      'This text is centered horizontally and vertically!',
    ),
  ),
);
```

Raised Button: Elevated button with a shadow effect for a prominent look.

Appearance: 3D-like with shadow
Functionality: Indicates primary action

Code Example:

```
RaisedButton(
  onPressed: () {
    print('Raised button pressed!');
  },
  child: Text('Raised Button'),
),
```

FlatButton: Flat button with minimal visual style, often used for secondary actions.

Appearance: Flat
Functionality: Secondary action

Code Example:

```
FlatButton(
  onPressed: () {
    print('Raised button pressed!');
  },
  child: Text('Raised Button'),
),
```

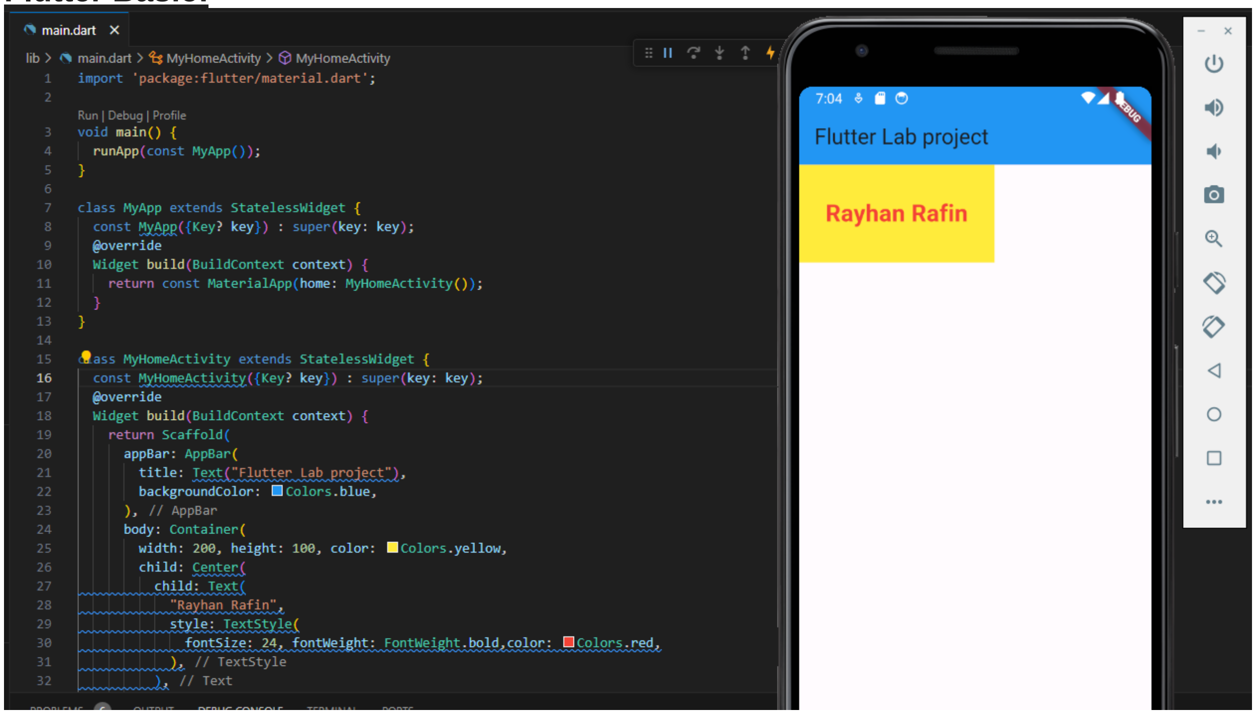
<u>Outlined Button:</u> Button with a colored border for a clear outline.

Appearance: Border around text
Functionality: Alternative or less important action

Code Example:

```
OutlinedButton(
  onPressed: () {
    print('outlined button pressed!');
  },
  child: Text('Outlined Button'),
  style: OutlinedButton.styleFrom(
    side: BorderSide(color: Colors.blue), // Customize border color
  ),
),
```

<u>Flutter Basic:</u>



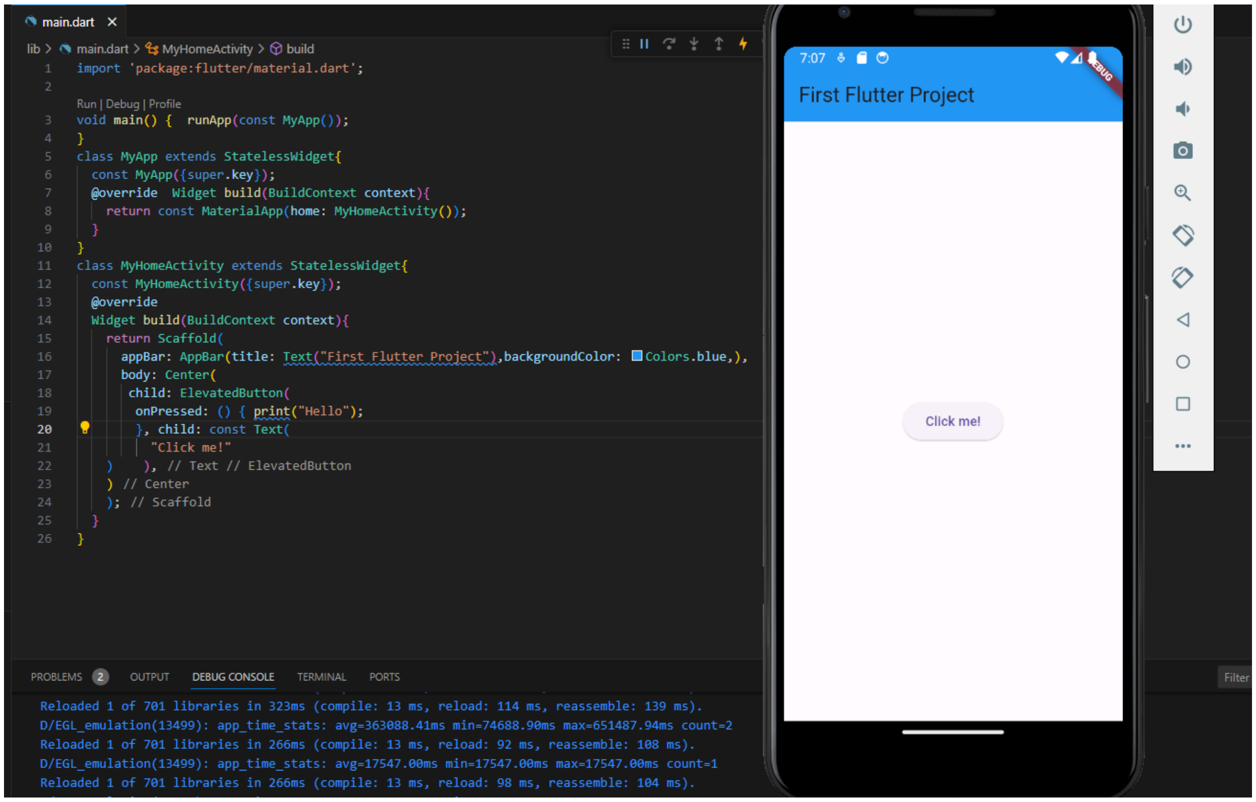
```
import 'package:flutter/material.dart';

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

class MyApp extends StatelessWidget {
  const MyApp({Key? key}) : super(key: key);
  @override
  Widget build(BuildContext context) {
    return const MaterialApp(home: MyHomeActivity());
  }
}

class MyHomeActivity extends StatelessWidget {
  const MyHomeActivity({Key? key}) : super(key: key);
  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
        title: Text("Flutter Lab project"),
        backgroundColor: Colors.blue,
      ), // AppBar
      body: Container(
        width: 200, height: 100, color: Colors.yellow,
        child: Center(
          child: Text(
            "Rayhan Rafin",
            style: TextStyle(
              fontSize: 24, fontWeight: FontWeight.bold,color: Colors.red,
            ), // TextStyle
          ), // Text
```

Button:



```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 const MaterialApp(home: MyHomeActivity());
  }
}
class MyHomeActivity extends StatelessWidget{
  const MyHomeActivity({super.key});
  @override
  Widget build(BuildContext context){
    return Scaffold(
      appBar: AppBar(title: Text("First Flutter Project"),backgroundColor: Colors.blue,),
      body: Center(
       child: ElevatedButton(
        onPressed: () { print("Hello");
        }, child: const Text(
          "Click me!"
    )     ), // Text // ElevatedButton
    ) // Center
    ); // Scaffold
  }
}
```

```
Reloaded 1 of 701 libraries in 323ms (compile: 13 ms, reload: 114 ms, reassemble: 139 ms).
D/EGL_emulation(13499): app_time_stats: avg=363088.41ms min=74688.90ms max=651487.94ms count=2
Reloaded 1 of 701 libraries in 266ms (compile: 13 ms, reload: 92 ms, reassemble: 108 ms).
D/EGL_emulation(13499): app_time_stats: avg=17547.00ms min=17547.00ms max=17547.00ms count=1
Reloaded 1 of 701 libraries in 266ms (compile: 13 ms, reload: 98 ms, reassemble: 104 ms).
```