**R22CSE3121: UI DESIGN-FLUTTER**

# B.Tech. III Year I Sem. L T P C

**0 0 2 1**

# Course Objectives:

* Learns to Implement Flutter Widgets and Layout
* Understands Responsive UI Design and with Navigation in Flutter
* Knowledge on Widges and customize widgets for specific UI elements, Themes
* Understand to include animation apart from fetching data

# Course Outcomes:

* Implements Flutter Widgets and Layouts
* Responsive UI Design and with Navigation in Flutter
* Create custom widgets for specific UI elements and also Apply styling using themes and custom styles.
* Design a form with various input fields, along with validation and error handling
* Fetches data and write code for unit Test for UI components and also animation

# List of Experiments: Students need to implement the following experiments

1. Install Flutter and Dart SDK.

2. Write a simple Dart program to understand the language basics.

3. Explore various Flutter widgets (Text, Image, Container, etc.).

4. Implement different layout structures using Row, Column, and Stack widgets.

5. Set up navigation between different screens using Navigator.

6. Implement navigation with named routes

7. Use Flutter's debugging tools to identify and fix issues.

8. Design a responsive UI that adapts to different screen sizes.

9. Implement media queries and breakpoints for responsiveness.

10. Create custom widgets for specific UI elements.

11. Apply styling using themes and custom styles.

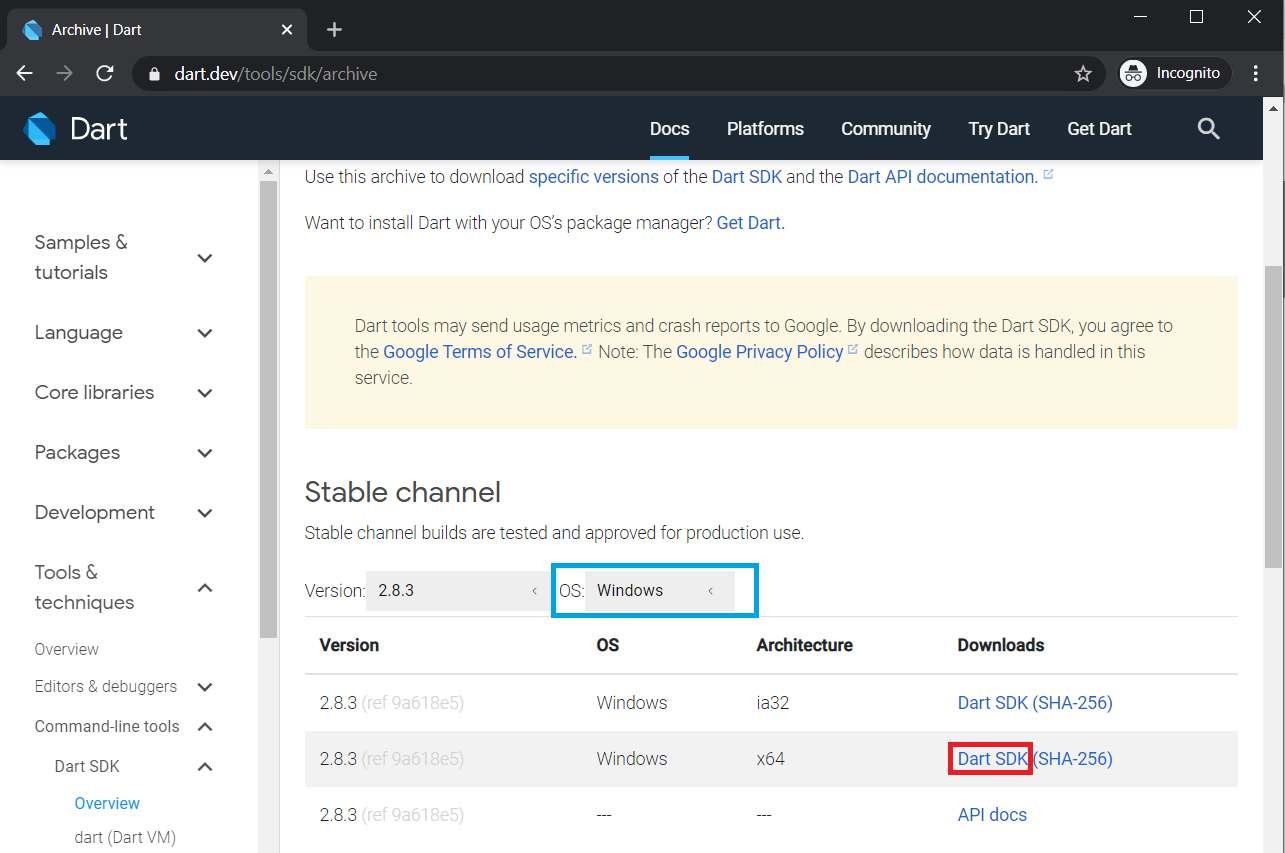
12. Add animations to UI elements using Flutter's animation framework.

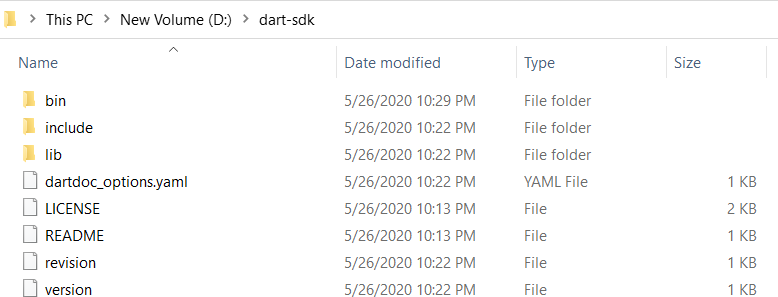
TEXT BOOK:

1. Marco L. Napoli, Beginning Flutter: A Hands-on Guide to App Development.

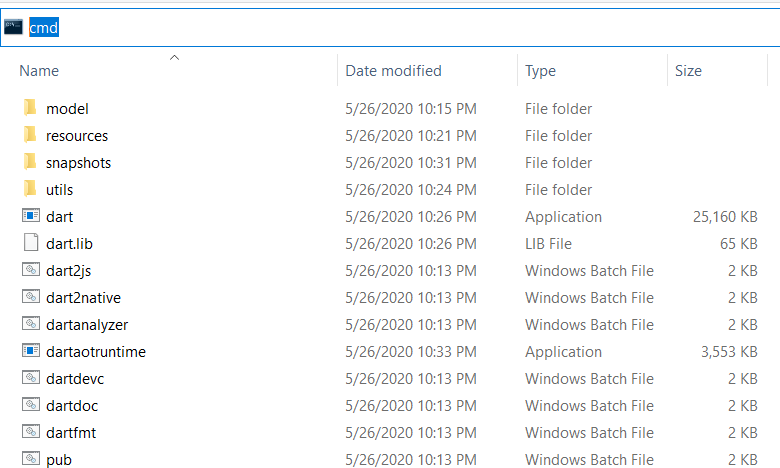
# 1. Install Flutter and Dart SDK.

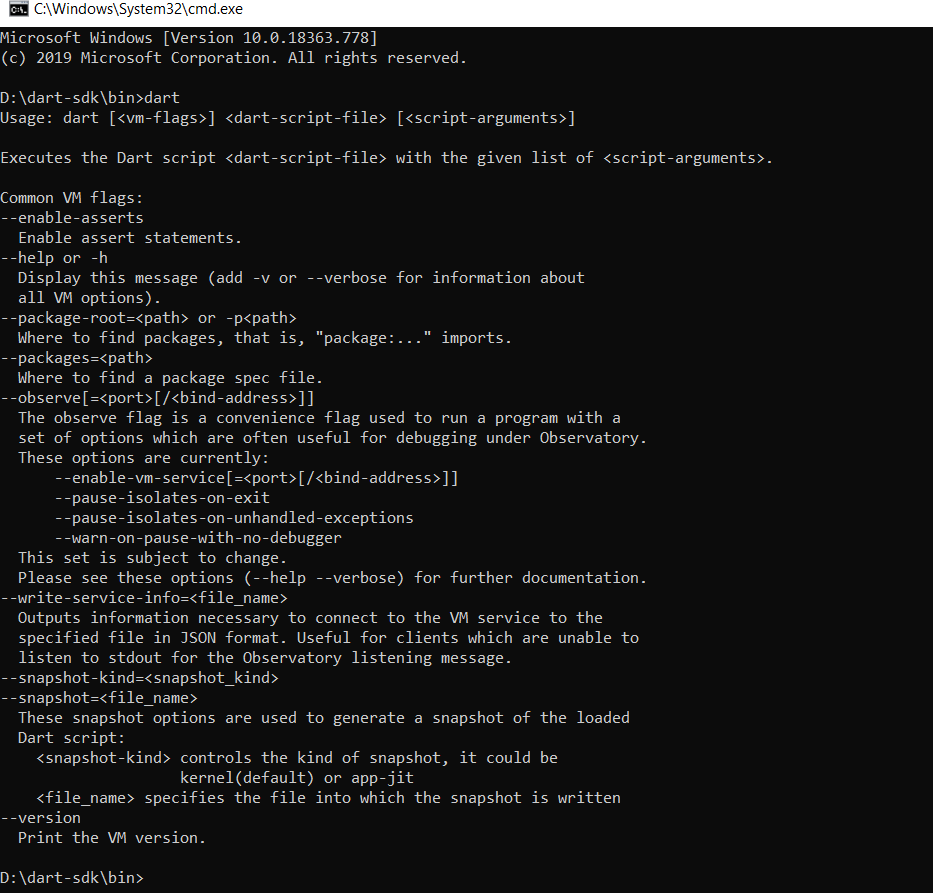
**Ans)** Dart SDK is a pre-compiled version so we have to download and extract it only. For this follow the below-given instructions: **Step 1:** Download Dart SDK. Download Dart SDK from the Dart SDK archive page. The URL is: <https://dart.dev/tools/sdk/archive>



Click on DART SDK to download SDK for Windows 64-Bit Architecture. The download will start and a zip file will be downloaded. **Note:** To download SDK for any other OS select OS of your choice. **Step 2:** Extract the downloaded zip file. Extract the contents of downloaded zip file and after extracting contents of zip file will be as shown:

**Step 3:** Running Dart. Now open bin folder and type “cmd” as given below:

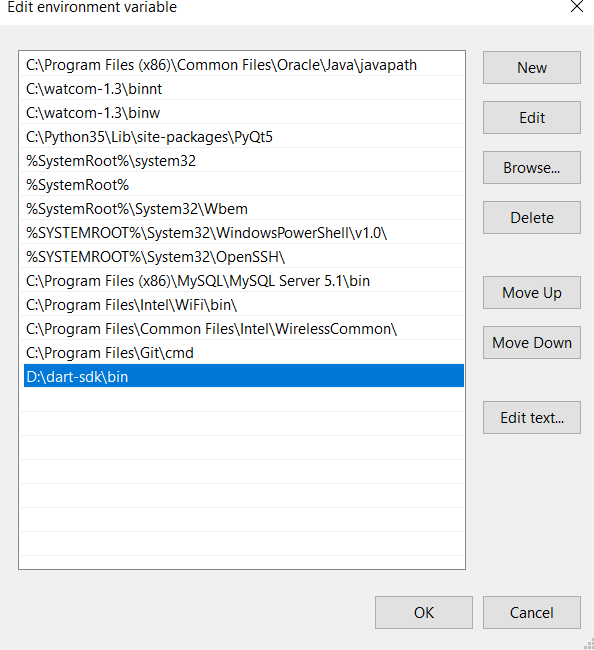


Command Prompt will open with our desired path of bin folder and now type dart”.

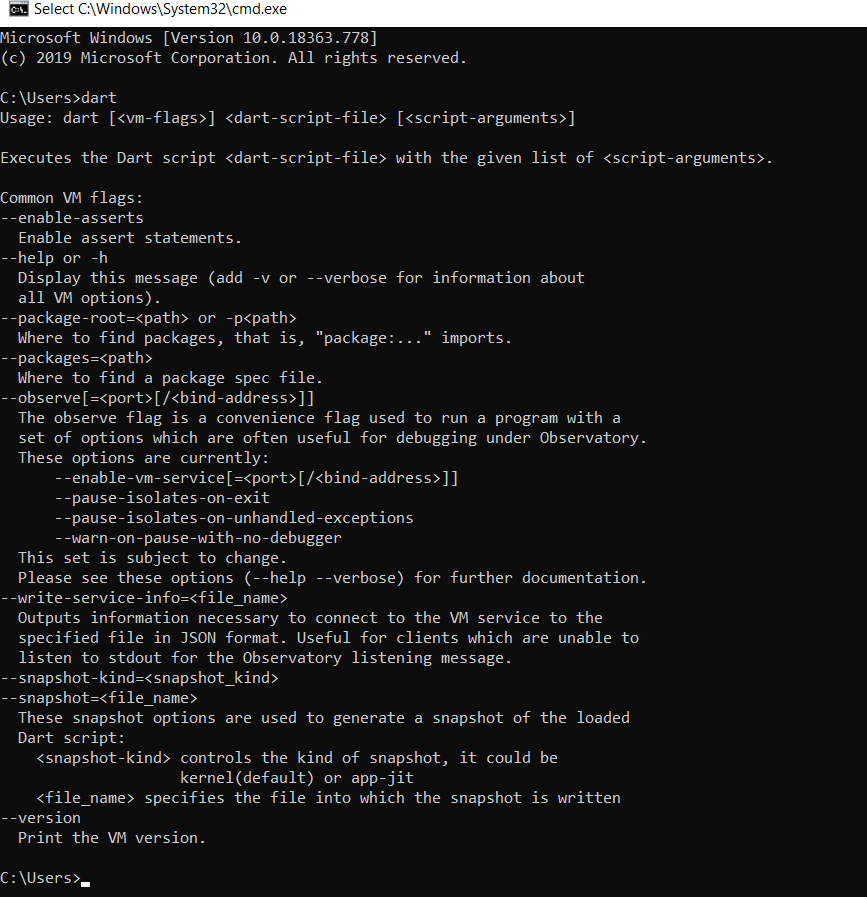
And now we are ready to use dart through bin folder but setting up the path in environment variables will ease our task of Step3 and we can run dart from anywhere in the file system using command prompt.

**Step 4:** Setting up path in environment variables. Open Environment Variables from advanced system settings and add Path in System Variables as depicted in image:

5



Now we are done to use Dart from anywhere in the file system.

**Step 5:** Run Dart Using cmd

# 2) Write a simple Dart program to understand the language basics. Ans)

void main(){

var firstName = "John"; var lastName = "Doe";

print("Full name is $firstName $lastName");

}

# Output: Full name is John Doe

void main() {

int num1 = 10; //declaring number1 int num2 = 3; //declaring number2

// Calculation

int sum = num1 + num2;

int diff = num1 - num2; int mul = num1 \* num2;

double div = num1 / num2; // It is double because it outputs number with decimal.

// displaying the output print("The sum is $sum"); print("The diff is $diff"); print("The mul is $mul"); print("The div is $div");

}

# Output:

**The sum is 13 The diff is 7 The mul is 30**

# The div is 3.3333333333333335

import 'dart:io'; void main() {

print("Enter number:");

int? number = int.parse(stdin.readLineSync()!); print("The entered number is ${number}");

}

# Output:

**Enter number:**

# 50

**The entered number is 50**

# 3. Explore various Flutter widgets (Text, Image, Container, etc.).

**Text Widget:**

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('Text Widget Example'),

        ),

        body: Center(

          child: Text(

            'Hello, Flutter!',

            style: TextStyle(

              fontSize: 24.0,

              fontWeight: FontWeight.bold,

              color: Colors.blue,

            ),

          ),

        ),

      ),

    );

  }

}

# Image Widget:

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('Image from Assets Example'),

        ),

        body: Center(

          child: Image.asset('images/ab.png'),

        ),

      ),

    );

  }

}

# Containter Widget:

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 MaterialApp(

home: Scaffold(

appBar: AppBar(

title: const Text("Container example"),

),

body: Container( height: 200,

width: double.infinity,

//color: Colors.purple, alignment: Alignment.center, margin: const EdgeInsets.all(20),

padding: const EdgeInsets.all(30), decoration: BoxDecoration(

border: Border.all(color: Colors.black, width: 3),

),

child: const Text("Hello! i am inside a container!", style: TextStyle(fontSize: 20)),

),

),

);

}

}

Output:



# 4) Implement different layout structures using Row, Column, and Stack widgets Row Widget

import 'package:flutter/material.dart';

void main() {

  runApp(MyApp());

}

class MyApp extends StatelessWidget {

  @override

  Widget build(BuildContext context) {

    return MaterialApp(home: MyHomePage());

  }

}

class MyHomePage extends StatefulWidget {

  @override

  \_MyHomePageState createState() => \_MyHomePageState();

}

class \_MyHomePageState extends State<MyHomePage> {

  @override

  Widget build(BuildContext context) {

    return Scaffold(

      appBar: AppBar(

        title: Text("Flutter Row Example"),

      ),

      body: 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.green),

              child: Text(

                "React.js",

                style: TextStyle(color: Colors.yellowAccent, fontSize: 25),

              ),

            ),

            Container(

              margin: EdgeInsets.all(15.0),

              padding: EdgeInsets.all(8.0),

              decoration: BoxDecoration(

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

              child: Text(

                "Flutter",

                style: TextStyle(color: Colors.yellowAccent, fontSize: 25),

              ),

            ),

            Container(

              margin: EdgeInsets.all(12.0),

              padding: EdgeInsets.all(8.0),

              decoration: BoxDecoration(

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

              child: Text(

                "MySQL",

                style: TextStyle(color: Colors.yellowAccent, fontSize: 25),

              ),

            )

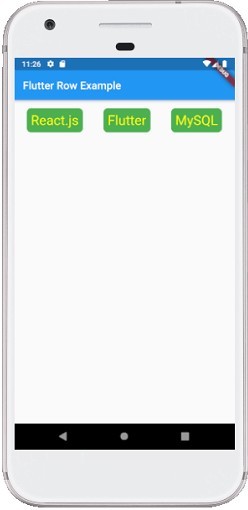
          ]),

    );

  }

}

Output:



**Column Widget:**

import 'package:flutter/material.dart';

void main() {

  runApp(MyApp());

}

class MyApp extends StatelessWidget {

  @override

  Widget build(BuildContext context) {

    return MaterialApp(home: MyHomePage());

  }

}

class MyHomePage extends StatefulWidget {

  @override

  \_MyHomePageState createState() => \_MyHomePageState();

}

class \_MyHomePageState extends State<MyHomePage> {

  @override

  Widget build(BuildContext context) {

    return Scaffold(

      appBar: AppBar(

        title: Text("Flutter Column Example"),

      ),

      body:Column (

          mainAxisAlignment: MainAxisAlignment.spaceEvenly,

          children: <Widget>[

            Container(

              margin: EdgeInsets.all(12.0),

              padding: EdgeInsets.all(8.0),

              decoration: BoxDecoration(

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

              child: Text(

                "React.js",

                style: TextStyle(color: Colors.yellowAccent, fontSize: 25),

              ),

            ),

            Container(

              margin: EdgeInsets.all(15.0),

              padding: EdgeInsets.all(8.0),

              decoration: BoxDecoration(

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

              child: Text(

                "Flutter",

                style: TextStyle(color: Colors.yellowAccent, fontSize: 25),

              ),

            ),

            Container(

              margin: EdgeInsets.all(12.0),

              padding: EdgeInsets.all(8.0),

              decoration: BoxDecoration(

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

              child: Text(

                "MySQL",

                style: TextStyle(color: Colors.yellowAccent, fontSize: 25),

              ),

            )

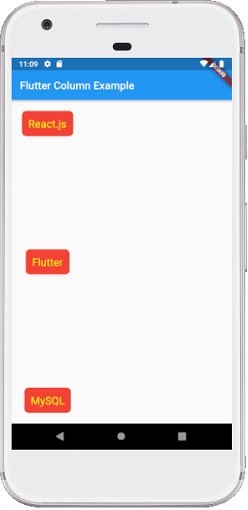
          ]),

    );

  }

}

**Output:**



**Stack Widget:**

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

runApp(MaterialApp(

home: Scaffold(

appBar: AppBar(

title: Text('GeeksforGeeks'), backgroundColor: Colors.greenAccent[400],

),

//Appar body:center(

child: SizedBox( width:300,

height:300, child:Center(

child: Stack( children: <Widget>[

Conainer( width: 300,

height: 300, color: Colors.red,

), //Container Container( width: 250,

height: 250,

color: Colors.black,

), //Container

Container( height: 200,

width: 200,

color: Colors.purple,

), //Container

], //<Widget>[]

), //Stack

), //Center

), //SizedBox

) //Center

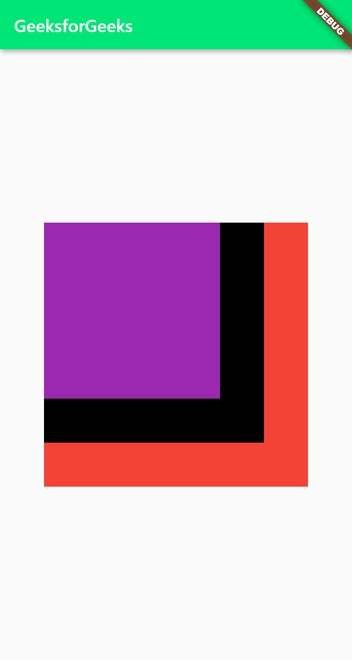
) //Scaffold

) //MaterialApp

);

}

**Output:**



# 5) Set up navigation between different screens using Navigator.

Ans)

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>Screen Navigation Example</title>

<style>

body {

font-family: Arial, sans-serif; margin: 0;

padding: 0;

background-color: #f4f4f4;

}

header {

background-color: #333; color: #fff;

text-align: center; padding: 1em;

}

main {

max-width: 1200px; margin: 0 auto; padding: 20px;

}

section { display: none;

}

footer {

background-color: #333; color: #fff;

text-align: center; padding: 1em; position: fixed; bottom: 0;

width: 100%;

}

.active {

display: block;

}

</style>

</head>

<body>

<header class="jumbotron text-center">

<h1>Screen Navigation Example</h1>

</header>

<main>

<section id="home" class="active">

<h2>Home Screen</h2>

<p>Welcome to the Home Screen.</p>

<button onclick="navigateTo('about')">Go to About</button>

</section>

<section id="about">

<h2>About Screen</h2>

<p>This is the About Screen.</p>

<button onclick="navigateTo('home')">Go to Home</button>

</section>

</main>

<footer class="bg-dark text-light text-center py-3 mt-5">

&copy; 2024 Your Company Name

</footer>

<script>

function navigateTo(screenId) {

// Hide all sections document.querySelectorAll('section').forEach(section => {

section.classList.remove('active');

});

// Show the selected section document.getElementById(screenId).classList.add('active');

}

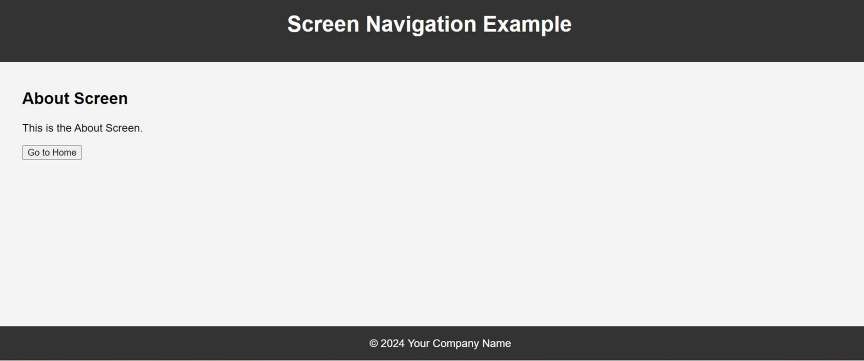
</script>

</body>

</html>

# Output:





# 6) Implement navigation with named routes.

**Ans)**

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

runApp(MyApp());

}

class MyApp extends StatelessWidget { @override

Widget build(BuildContext context) { return MaterialApp(

title: 'Named Routes Navigation Example', initialRoute: '/',

routes: {

'/': (context) => HomeScreen(), '/about': (context) => AboutScreen(),

},

);

}

}

class HomeScreen extends StatelessWidget { @override

Widget build(BuildContext context) { return Scaffold(

appBar: AppBar(

title: Text('Home Screen'),

),

body: Center( child: Column(

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

Text(

'Welcome to the Home Screen.',

),

SizedBox(height: 20), ElevatedButton( onPressed: () {

Navigator.pushNamed(context, '/about');

},

child: Text('Go to About'),

), ],

),

),

);

}

}

class AboutScreen extends StatelessWidget { @override

Widget build(BuildContext context) { return Scaffold(

appBar: AppBar(

title: Text('About Screen'),

),

body: Center( child: Column(

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

Text(

'This is the About Screen.',

),

SizedBox(height: 20), ElevatedButton( onPressed: () {

Navigator.pop(context);

},

child: Text('Go back to Home'),

),

],

),

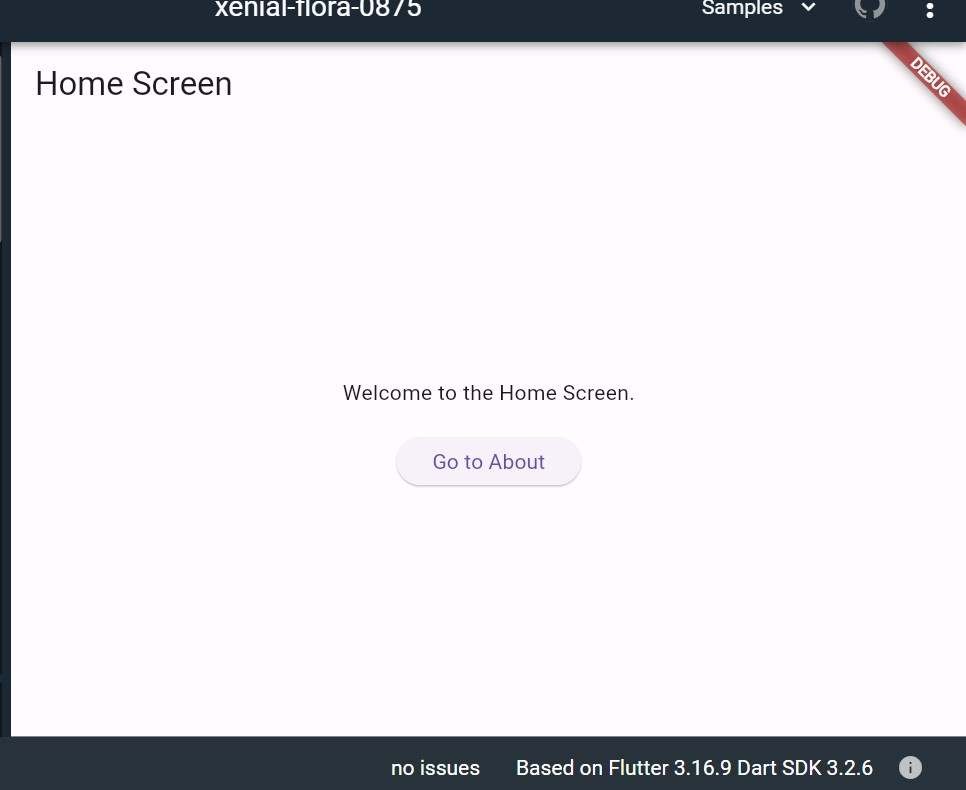
),

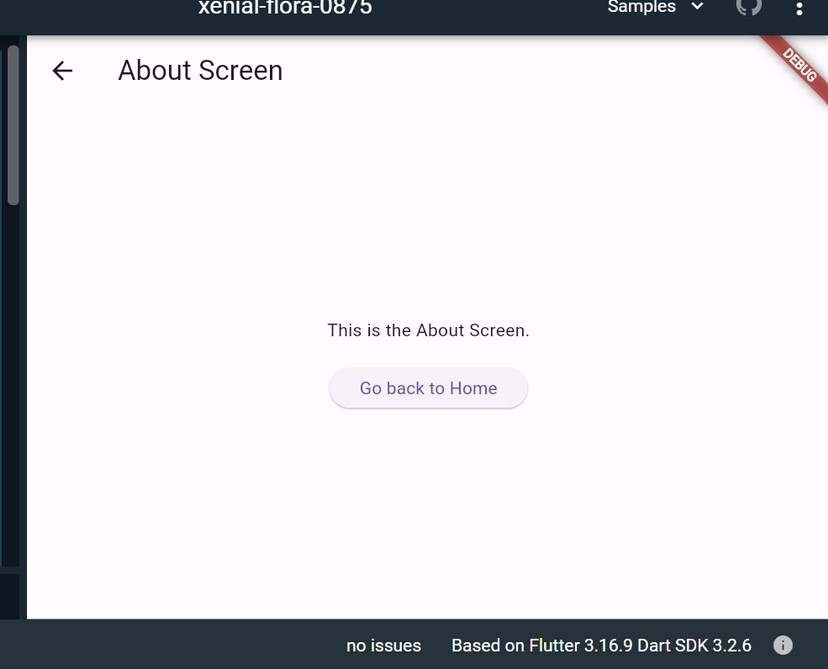
);

}

}

# Output:





# 7) Use Flutter's debugging tools to identify and fix issues.

**Ans)** Flutter provides a set of debugging tools that can help you identify and fix issues in your app. Here's a step-by-step guide on how to use these tools:

1. Flutter DevTools:

Run your app with the flutter run command.

Open DevTools by running the following command in your terminal: bash

flutter pub global activate devtools flutter pub global run devtools

Open your app in a Chrome browser and connect it to DevTools by clicking on the "Open DevTools" button in the terminal or by navigating to http://127.0.0.1:9100/.

DevTools provides tabs like Inspector, Timeline, Memory, and more.

1. Flutter Inspector:

Use the Flutter Inspector in your integrated development environment (IDE) like Android Studio or Visual Studio Code.

Toggle the Inspector in Android Studio with the shortcut Alt + Shift + D (Windows/Linux) or Option + Shift + D (Mac).

Inspect the widget tree, modify widget properties, and observe widget relationships.

1. Hot Reload:

Leverage Hot Reload to see the immediate effect of code changes without restarting the entire app.

Press R in the terminal or use the "Hot Reload" button in your IDE.

1. Debugging with Breakpoints:

Set breakpoints in your code to pause execution and inspect variables. Use the debugger in your IDE to step through code and identify issues.

1. Logging:

Utilize the print function to log messages to the console.

print('Debugging message');

View logs in the terminal or the "Logs" tab in DevTools.

1. Debug Paint:

Enable debug paint to visualize the layout and rendering of widgets. Use the debugPaintSizeEnabled and debugPaintBaselinesEnabled flags.

void main() {

debugPaintSizeEnabled = true; // Shows bounding boxes of widgets runApp(MyApp());

}

1. Memory Profiling:

Use the "Memory" tab in DevTools to analyze memory usage and identify potential memory leaks.

Monitor object allocations and deallocations.

1. Performance Profiling (Timeline):

Analyze app performance using the "Timeline" tab in DevTools. Identify UI jank, slow frames, and performance bottlenecks.

1. Flutter Driver Tests:

Write automated UI tests using Flutter Driver.

Simulate user interactions and validate the correctness of your UI.

**8) Design a responsive UI that adapts to different screen sizes.**

**Ans)**

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<link rel="stylesheet"

href="https://stackpath.bootstrapcdn.com/bootstrap/4.5.2/css/bootstrap.min.css">

<title>Responsive UI Example</title>

</head>

<body>

<div class="container">

<header class="jumbotron text-center">

<h1>Responsive UI Example</h1>

</header>

<main>

<section class="mb-4">

<h2>Section 1</h2>

<p>This is some content for section 1.</p>

</section>

<section class="mb-4">

<h2>Section 2</h2>

<p>This is some content for section 2.</p>

</section>

</main>

<footer class="bg-dark text-light text-center py-3 mt-5"> &copy; 2024 Your Company Name

</footer>

</div>

<!-- Bootstrap JS and dependencies (jQuery) -->

<script src="https://code.jquery.com/jquery-3.5.1.slim.min.js"></script>

<script src=["ht](https://cdn.jsdelivr.net/npm/%40popperjs/core%402.10.2/dist/umd/popper.min.js)t[ps://cdn.jsdelivr.net/npm/@popperjs/core@2.10.2/dist/umd/popper.min.js](https://cdn.jsdelivr.net/npm/%40popperjs/core%402.10.2/dist/umd/popper.min.js)"></ script>

<script src="https://stackpath.bootstrapcdn.com/bootstrap/4.5.2/js/bootstrap.min.js"></ script>

</body>

</html>

**Output:**



# 9) Implement media queries and breakpoints for responsiveness.

Ans) <!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<link rel="stylesheet"

href="https://stackpath.bootstrapcdn.com/bootstrap/4.5.2/css/bootstrap.min.css">

<style>

body {

font-family: Arial, sans-serif; margin: 0;

padding: 0;

background-color: #f4f4f4;

}

header {

background-color: #333; color: #fff;

text-align: center; padding: 1em;

}

main {

max-width: 1200px; margin: 0 auto; padding: 20px;

}

section {

margin-bottom: 20px;

}

footer {

background-color: #333; color: #fff;

text-align: center; padding: 1em; position: fixed; bottom: 0;

width: 100%;

}

@media only screen and (max-width: 768px) { main {

padding: 10px;

}

footer {

position: static;

}

}

</style>

<title>Responsive UI Example</title>

</head>

<body>

<div class="container">

<header class="jumbotron text-center">

<h1>Responsive UI Example</h1>

</header>

<main>

<section class="mb-4">

<h2>Section 1</h2>

<p>This is some content for section 1.</p>

</section>

<section class="mb-4">

<h2>Section 2</h2>

<p>This is some content for section 2.</p>

</section>

</main>

<footer class="bg-dark text-light text-center py-3 mt-5"> &copy; 2024 Your Company Name

</footer>

</div>

<!-- Bootstrap JS and dependencies (jQuery) -->

<script src="https://code.jquery.com/jquery-3.5.1.slim.min.js"></script>

<script src=["ht](https://cdn.jsdelivr.net/npm/%40popperjs/core%402.10.2/dist/umd/popper.min.js)t[ps://cdn.jsdelivr.net/npm/@popperjs/core@2.10.2/dist/umd/popper.min.js](https://cdn.jsdelivr.net/npm/%40popperjs/core%402.10.2/dist/umd/popper.min.js)"></ script>

<script src="https://stackpath.bootstrapcdn.com/bootstrap/4.5.2/js/bootstrap.min.js"></ script>

</body>

</html>

Output:



**10) Create custom widgets for specific UI elements.**

# Ans)

import 'package:flutter/material.dart';

class CustomButton extends StatelessWidget { final String text;

final Function onPressed; final Color buttonColor; final Color textColor; CustomButton({

required this.text, required this.onPressed,

this.buttonColor = Colors.blue, this.textColor = Colors.white,

});

@override

Widget build(BuildContext context) { return ElevatedButton(

onPressed: () => onPressed(), style: ButtonStyle(

backgroundColor: MaterialStateProperty.all<Color>(buttonColor), foregroundColor: MaterialStateProperty.all<Color>(textColor),

),

child: Text(text),

);

}

}

class CustomAlertDialog extends StatelessWidget { final String title;

final String message;

final String positiveButtonText; final String negativeButtonText; final Function onPositivePressed; final Function onNegativePressed; CustomAlertDialog({

required this.title, required this.message,

required this.positiveButtonText, required this.negativeButtonText, required this.onPositivePressed, required this.onNegativePressed,

});

@override

Widget build(BuildContext context) { return AlertDialog(

title: Text(title), content: Text(message), actions: <Widget>[ CustomButton(

text: negativeButtonText,

onPressed: () => onNegativePressed(),

),

CustomButton(

text: positiveButtonText,

onPressed: () => onPositivePressed(),

),

],

);

}

}

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

}

class MyApp extends StatelessWidget { @override

Widget build(BuildContext context) { return MaterialApp(

home: Scaffold( appBar: AppBar(

title: Text('Custom Button Example'),

),

body: Center(

child: CustomButton( text: 'Click Me', onPressed: () {

// Handle button press print('Button Pressed');

},

),

),

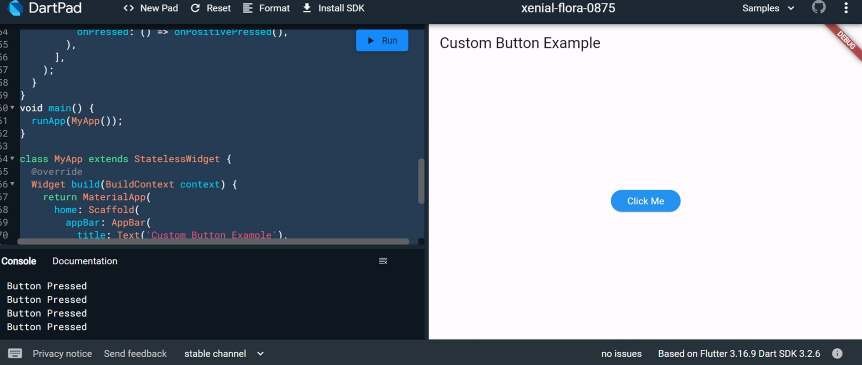
),

);

}

}

# Output:



**11) Apply styling using themes and custom styles.**

**Ans)**

import 'package:flutter/material.dart';

import 'package:google\_fonts/google\_fonts.dart';

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

}

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

@override

Widget build(BuildContext context) { const appName = 'Custom Themes';

return MaterialApp( title: appName, theme: ThemeData( useMaterial3: true,

// Define the default brightness and colors. colorScheme: ColorScheme.fromSeed( seedColor: Colors.purple,

// TRY THIS: Change to "Brightness.light"

// and see that all colors change

// to better contrast a light background. brightness: Brightness.dark,

),

// Define the default `TextTheme`. Use this to specify the default

// text styling for headlines, titles, bodies of text, and more. textTheme: TextTheme(

displayLarge: const TextStyle( fontSize: 72,

fontWeight: FontWeight.bold,

),

// TRY THIS: Change one of the GoogleFonts

// to "lato", "poppins", or "lora".

// The title uses "titleLarge"

// and the middle text uses "bodyMedium". titleLarge: GoogleFonts.oswald(

fontSize: 30,

fontStyle: FontStyle.italic,

),

bodyMedium: GoogleFonts.merriweather(), displaySmall: GoogleFonts.pacifico(),

),

),

home: const MyHomePage( title: appName,

),

);

}

}

class MyHomePage extends StatelessWidget { final String title;

const MyHomePage({super.key, required this.title}); @override

Widget build(BuildContext context) { return Scaffold(

appBar: AppBar( title: Text(title,

style: Theme.of(context).textTheme.titleLarge!.copyWith( color: Theme.of(context).colorScheme.onSecondary,

)),

backgroundColor: Theme.of(context).colorScheme.secondary,

),

body: Center( child: Container(

padding: const EdgeInsets.symmetric( horizontal: 12,

vertical: 12,

),

color: Theme.of(context).colorScheme.primary, child: Text(

'Text with a background color',

// TRY THIS: Change the Text value

// or change the Theme.of(context).textTheme

// to "displayLarge" or "displaySmall".

style: Theme.of(context).textTheme.bodyMedium!.copyWith( color: Theme.of(context).colorScheme.onPrimary,

),

),

),

),

floatingActionButton: Theme(

data: Theme.of(context).copyWith(

// TRY THIS: Change the seedColor to "Colors.red" or

// "Colors.blue".

colorScheme: ColorScheme.fromSeed( seedColor: Colors.pink,

brightness: Brightness.dark,

),

),

child: FloatingActionButton( onPressed: () {},

child: const Icon(Icons.add),

),

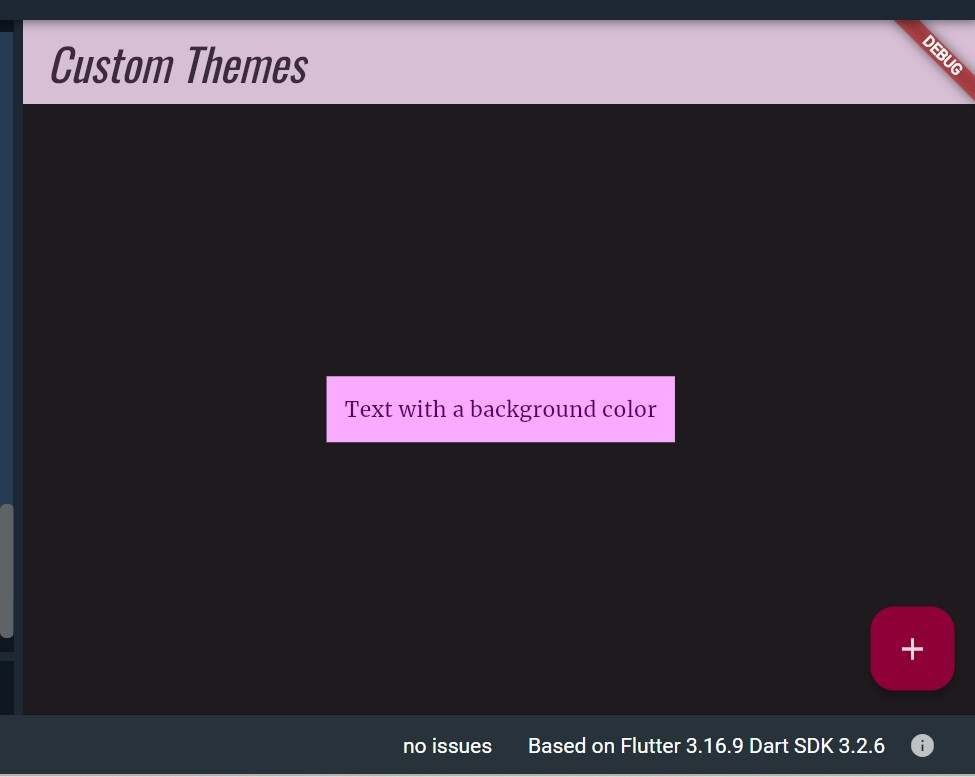
),

);

}

}

Output:



# 12) Add animations to UI elements using Flutter's animation framework.

**Ans)**

import 'package:flutter/material.dart';

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

}

class MyApp extends StatelessWidget { @override

Widget build(BuildContext context) { return MaterialApp(

title: 'Animation Example', theme: ThemeData( primarySwatch: Colors.blue,

),

home: MyAnimatedWidget(),

);

}

}

class MyAnimatedWidget extends StatefulWidget { @override

\_MyAnimatedWidgetState createState() => \_MyAnimatedWidgetState();

}

class \_MyAnimatedWidgetState extends State<MyAnimatedWidget>

with SingleTickerProviderStateMixin {

late AnimationController \_animationController; late Animation<double> \_opacityAnimation;

@override

void initState() { super.initState();

// Create an AnimationController with a duration of 1 second

\_animationController = AnimationController( vsync: this,

duration: Duration(seconds: 1),

);

// Create a Tween to animate opacity from 0.0 to 1.0

\_opacityAnimation = Tween<double>(begin: 0.0, end: 1.0).animate( CurvedAnimation(

parent: \_animationController, curve: Curves.easeInOut,

),

);

// Start the animation

\_animationController.forward();

}

@override

Widget build(BuildContext context) { return Scaffold(

appBar: AppBar(

title: Text('Animation Example'),

),

body: Center(

child: FadeTransition( opacity: \_opacityAnimation, child: Container(

width: 200,

height: 200,

color: Colors.blue, child: Center( child: Text(

'Animated Widget', style: TextStyle( color: Colors.white, fontSize: 20,

),

),

),

),

),

),

);

}

@override

void dispose() {

\_animationController.dispose(); super.dispose();

}

}

Output:

