J.N.T.U.H. UNIVERSITY COLLEGE OF ENGINEERING SCIENCE AND TECHNOLOGY HYDERABAD, KUKATPALLY, HYDERABAD – 500085

This is to certify that ***NALLABOTHULA MADHURI*** of CSE(Regular) III year I Semester bearing the Hall-Ticket number ***23015A0519*** has fulfilled her ***UI DESIGN- FLUTTER LAB*** record for the academic year 2024-2025.

___________________________________ ____________________________

Signature of the Head of the Department ***Signature of the Staff Member***

Date of Examination ______________________

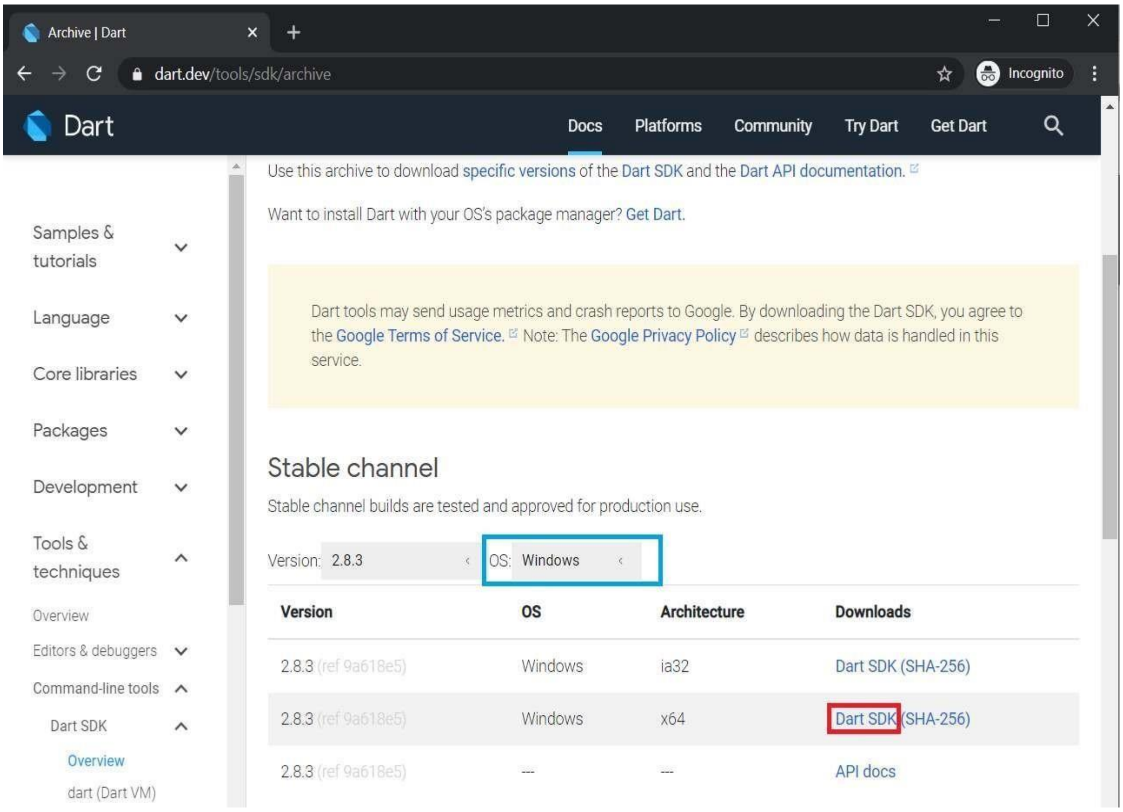
______________________ _____________________

Internal Examiner ***External Examiner***

INDEX

1. a) 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



Use this archive to download specific versions of the Dart SDK and the Dart API documentation.

Want to install Dart with your OS's package manager? Get Dart.

Dart tools may send usage metrics and crash reports to Google. By downloading the Dart SDK, you agree to the Google Terms of Service. Note: The Google Privacy Policy describes how data is handled in this service.

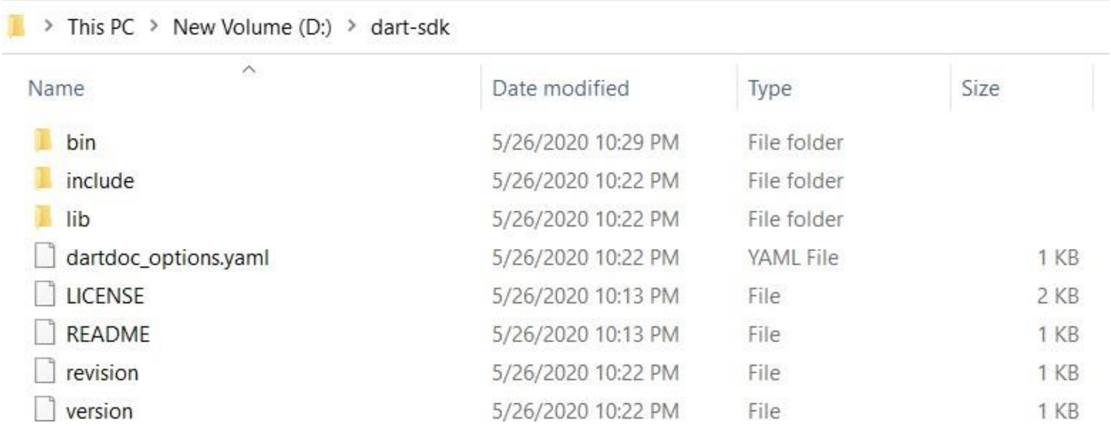
Stable channel

Stable channel builds are tested and approved for production use.

Version: 2.8.3 OS: Windows

Version	OS	Architecture	Downloads
2.8.3 (ref 9a618e5)	Windows	ia32	Dart SDK (SHA-256)
2.8.3 (ref 9a618e5)	Windows	x64	Dart SDK (SHA-256)
2.8.3 (ref 9a618e5)	---	---	API docs

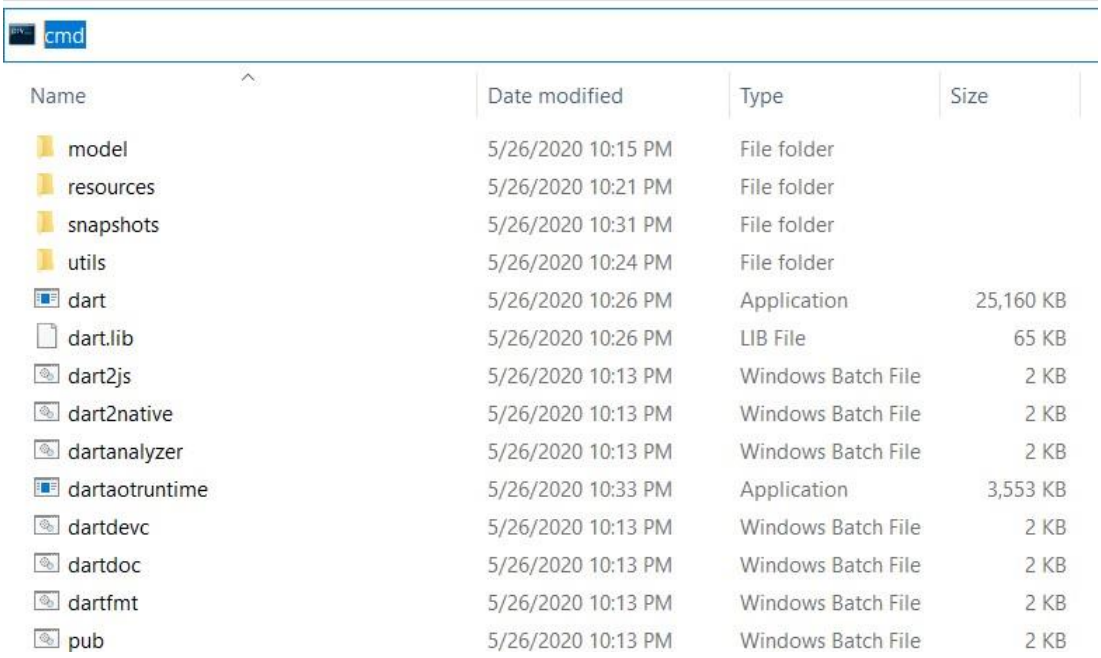
Click on DART SDK to download SDK for Windows 64-Bit Architecture. The download will start and a zip file will be downloaded. **Note:** To download SDK for any other OS select OS of your choice. **Step 2:** Extract the downloaded zip file. Extract the contents of downloaded zip file and after extracting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



> This PC > New Volume (D:) > dart-sdk

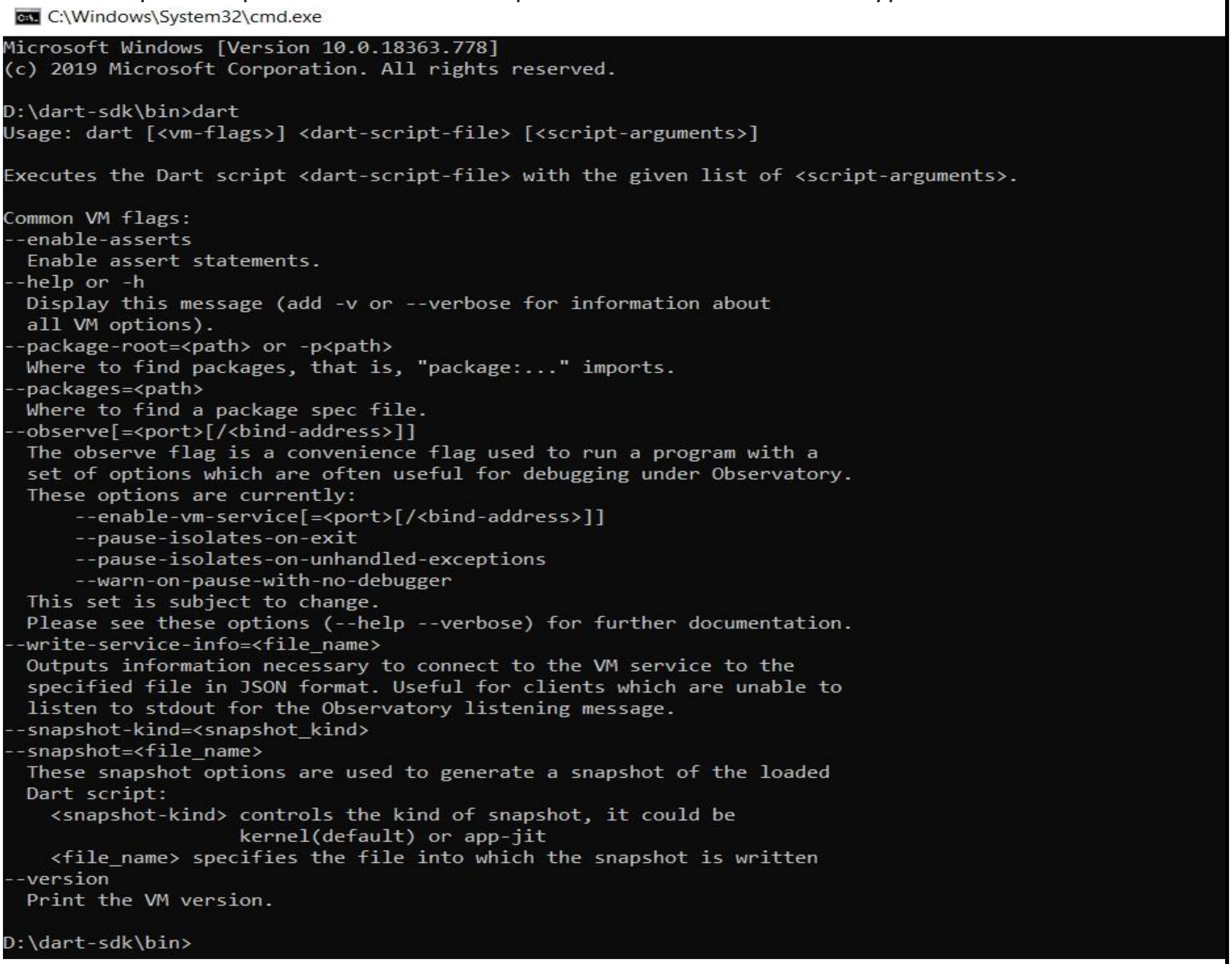
Name	Date modified	Type	Size
bin	5/26/2020 10:29 PM	File folder	
include	5/26/2020 10:22 PM	File folder	
lib	5/26/2020 10:22 PM	File folder	
dartdoc_options.yaml	5/26/2020 10:22 PM	YAML File	1 KB
LICENSE	5/26/2020 10:13 PM	File	2 KB
README	5/26/2020 10:13 PM	File	1 KB
revision	5/26/2020 10:22 PM	File	1 KB
version	5/26/2020 10:22 PM	File	1 KB

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



Name	Date modified	Type	Size
model	5/26/2020 10:15 PM	File folder	
resources	5/26/2020 10:21 PM	File folder	
snapshots	5/26/2020 10:31 PM	File folder	
utils	5/26/2020 10:24 PM	File folder	
dart	5/26/2020 10:26 PM	Application	25,160 KB
dart.lib	5/26/2020 10:26 PM	LIB File	65 KB
dart2js	5/26/2020 10:13 PM	Windows Batch File	2 KB
dart2native	5/26/2020 10:13 PM	Windows Batch File	2 KB
dartanalyzer	5/26/2020 10:13 PM	Windows Batch File	2 KB
dartaotruntime	5/26/2020 10:33 PM	Application	3,553 KB
dartdevc	5/26/2020 10:13 PM	Windows Batch File	2 KB
dartdoc	5/26/2020 10:13 PM	Windows Batch File	2 KB
dartfmt	5/26/2020 10:13 PM	Windows Batch File	2 KB
pub	5/26/2020 10:13 PM	Windows Batch File	2 KB

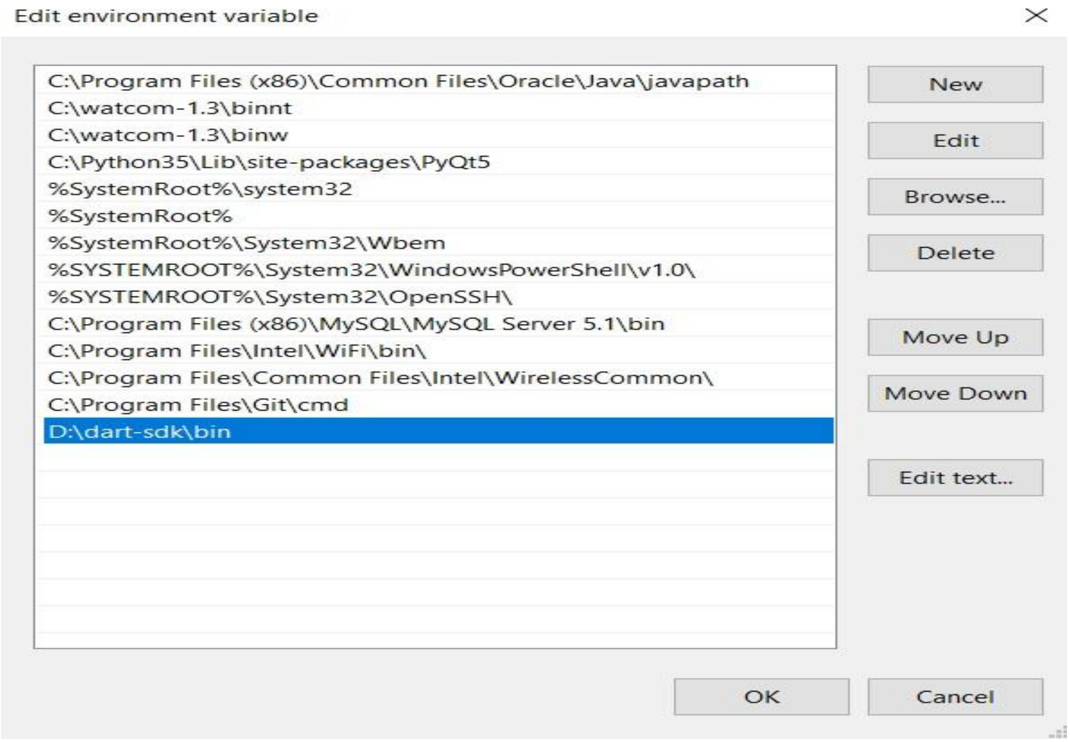
Command Prompt will open with our desired path of bin folder and now type dart”.



```
Microsoft Windows [Version 10.0.18363.778]
(c) 2019 Microsoft Corporation. All rights reserved.

D:\dart-sdk\bin>dart
Usage: dart [<vm-flags>] <dart-script-file> [<script-arguments>]

Executes the Dart script <dart-script-file> with the given list of <script-arguments>.

Common VM flags:
--enable-asserts
  Enable assert statements.
--help or -h
  Display this message (add -v or --verbose for information about
  all VM options).
--package-root=<path> or -p<path>
  Where to find packages, that is, "package:..." imports.
--packages=<path>
  Where to find a package spec file.
--observe[=<port>[/<bind-address>]]
  The observe flag is a convenience flag used to run a program with a
  set of options which are often useful for debugging under Observatory.
  These options are currently:
      --enable-vm-service[=<port>[/<bind-address>]]
      --pause-isolates-on-exit
      --pause-isolates-on-unhandled-exceptions
      --warn-on-pause-with-no-debugger
  This set is subject to change.
  Please see these options (--help --verbose) for further documentation.
--write-service-info=<file_name>
  Outputs information necessary to connect to the VM service to the
  specified file in JSON format. Useful for clients which are unable to
  listen to stdout for the Observatory listening message.
--snapshot-kind=<snapshot_kind>
--snapshot=<file_name>
  These snapshot options are used to generate a snapshot of the loaded
  Dart script:
    <snapshot-kind> controls the kind of snapshot, it could be
                    kernel(default) or app-jit
    <file_name> specifies the file into which the snapshot is written
--version
  Print the VM version.

D:\dart-sdk\bin>
```

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:



Edit environment variable

```
C:\Program Files (x86)\Common Files\Oracle\Java\javapath
C:\watcom-1.3\binnt
C:\watcom-1.3\binw
C:\Python35\Lib\site-packages\PyQt5
%SystemRoot%\system32
%SystemRoot%
%SystemRoot%\System32\Wbem
%SYSTEMROOT%\System32\WindowsPowerShell\v1.0\
%SYSTEMROOT%\System32\OpenSSH\
C:\Program Files (x86)\MySQL\MySQL Server 5.1\bin
C:\Program Files\Intel\WiFi\bin\
C:\Program Files\Common Files\Intel\WirelessCommon\
C:\Program Files\Git\cmd
D:\dart-sdk\bin
```

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

Step 5: Run Dart Using cmd

1b) 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}"); }
```

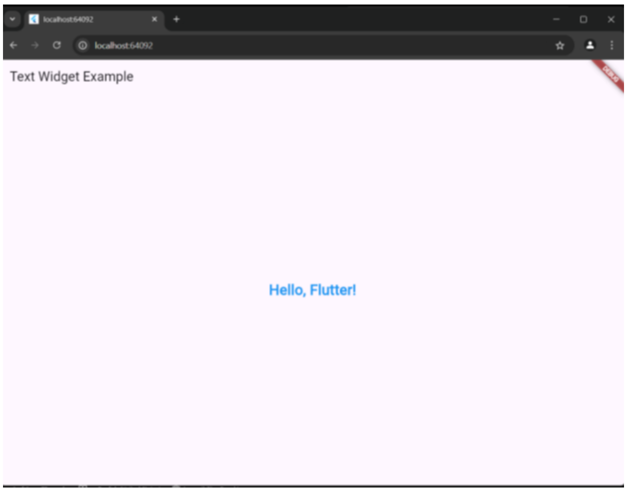
Output:

```
        Enter number:
        50
        The entered number is 50
```

2a). 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,
             fontWeight: FontWeight.bold,
             color: Colors.blue,
           ),
         ),
       ),
     ),
   );
 }
}
```



Text Widget Example

Hello, Flutter!

Image widget :

Pubsec.yaml:

```
flutter:
  assets:
    - assets/google.jpg
```

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('Image Asset Example'),
        ),
        body: Center(
          child: Image.asset('assets/google.jpg'),
        ),
      ),
    );
  }
}
```

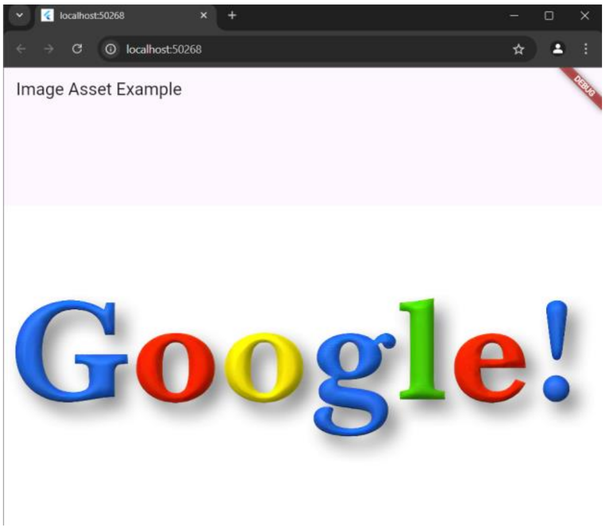
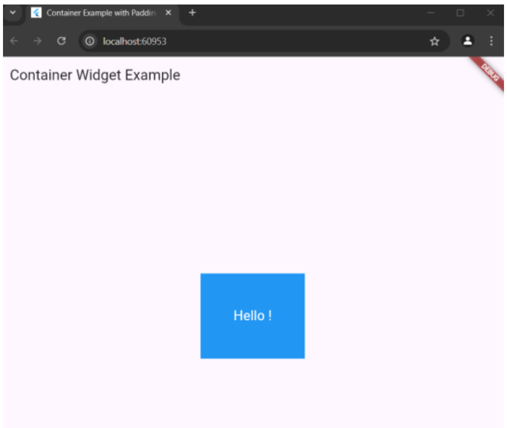


Image Asset Example

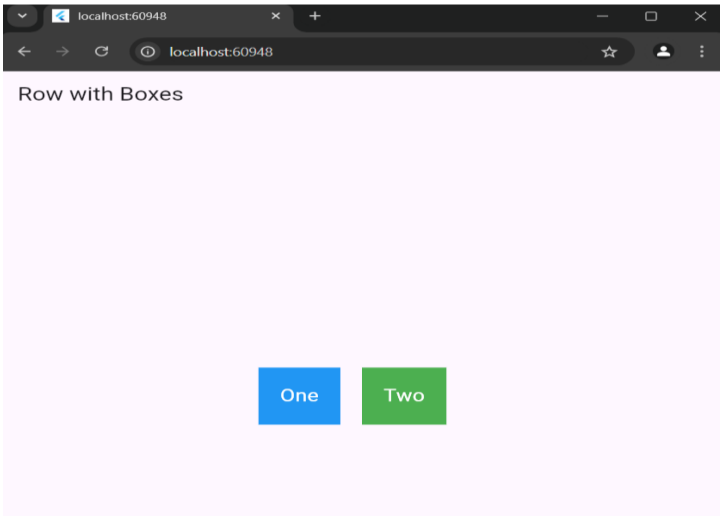
Container Widget :

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

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

class MyApp extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      title: 'Container Example with Padding and Margin',
      theme: ThemeData(primarySwatch: Colors.blue),
      home: Scaffold(
        appBar: AppBar(title: Text('Container Widget Example')),
        body: Center(
          child: Container(
            padding: EdgeInsets.all(50),
            margin: EdgeInsets.all(50),
            color: Colors.blue,
            child: Text(
              'Hello !',
              style: TextStyle(color: Colors.white, fontSize: 20),
            ),),),),);
  }
}
```



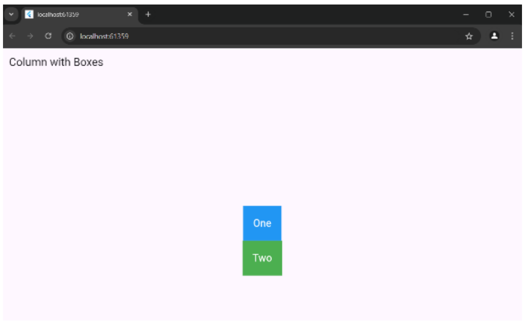
2b.Implement different layout structures using Row, Column, and Stack widgets.

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

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

class MyApp extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      debugShowCheckedModeBanner: false,

      home: Scaffold(
        appBar: AppBar(
          title: Text('Row with Boxes'),
        ),
        body: Center(
          child: Row(
            mainAxisAlignment: MainAxisAlignment.center,
            children: [
              Container(
                padding: EdgeInsets.all(20),
                color: Colors.blue,

                child: Text(
                  'One',
                  style: TextStyle(color: Colors.white, fontSize: 20),
                ),
              ),
              SizedBox(width: 20),

         Container(
                padding: EdgeInsets.all(20),
                color: Colors.green,
                child: Text(
                  'Two',
                  style: TextStyle(color: Colors.white, fontSize: 20),
                ),
              ),
              SizedBox(width: 20),

            ],
          ),
        ) ;}   }
```



Column Widget :

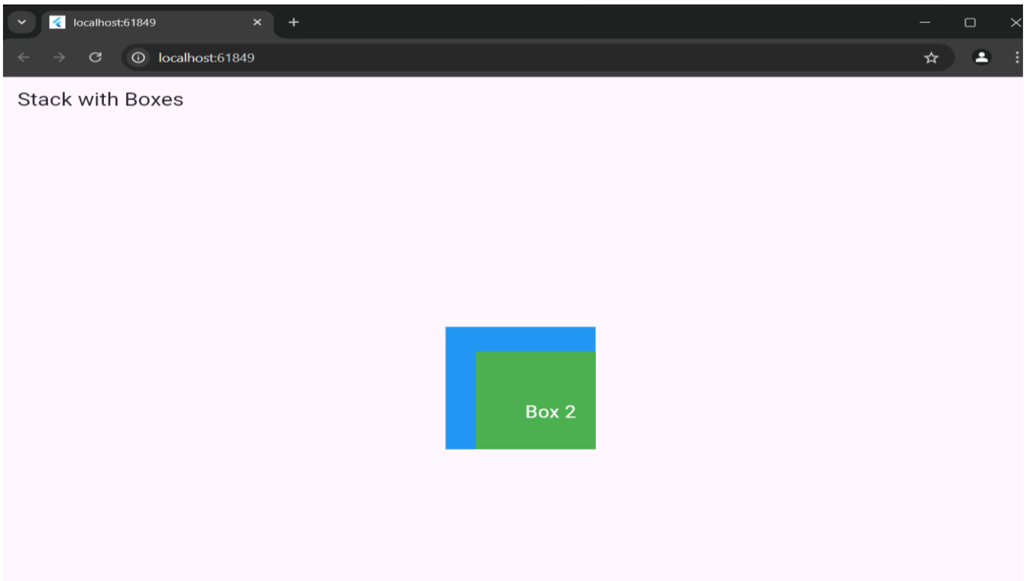
```
import 'package:flutter/material.dart';

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

class MyApp extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      debugShowCheckedModeBanner: false,
      home: Scaffold(
        appBar: AppBar(
          title: Text('Column with Boxes'),
        ),
        body: Center(
          child: Column(
           mainAxisAlignment: MainAxisAlignment.center,

            children: [
              Container(
                padding: EdgeInsets.all(20),
                color: Colors.blue,

                child: Text(
                  'One',
                  style: TextStyle(color: Colors.white, fontSize: 20),
        ),
       ),
       SizedBox(width: 20),

        Container(
        padding: EdgeInsets.all(20),
        color: Colors.green,
        child: Text(
         'Two',
         style: TextStyle(color: Colors.white, fontSize: 20),
        ),
       ),
       SizedBox(width: 20),


      ],
     ),
    ),
   ),
  );
 }
}
```



Stack widget:

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

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

class MyApp extends StatelessWidget {
 @override
 Widget build(BuildContext context) {
  return MaterialApp(
   debugShowCheckedModeBanner: false,
   home: Scaffold(
    appBar: AppBar(
     title: Text('Stack with Boxes'),
    ),
    body: Center(
      child: Stack(
        children: [

          Container(
            width: 150,
            height: 150,
            color: Colors.blue,
            child: Center(
              child: Text(
                'Box 1',
                style: TextStyle(color: Colors.white, fontSize: 20),
              ),
            ),
          ),

          Positioned(
            top: 30,
            left: 30,
            child: Container(
              width: 150,
              height: 150,
              color: Colors.green,
              child: Center(
                child: Text(
                  'Box 2',
                  style: TextStyle(color: Colors.white, fontSize: 20),
                ),
              ),
            ),
          ),
        ],
      ),
    ),
  ),
);
```

```
}}}
```

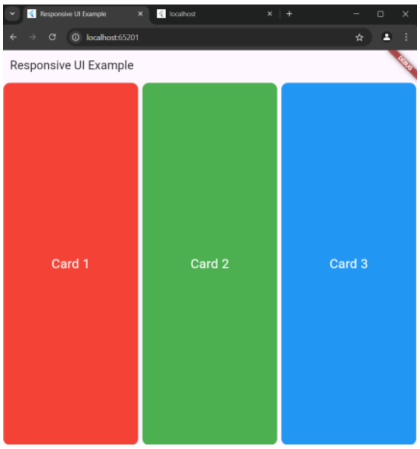


Stack with Boxes

Box 2

3a. Design a responsive UI that adapts to different screen sizes.

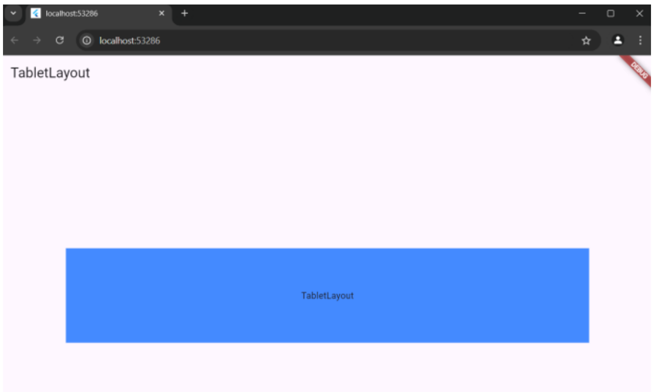
```
import 'package:flutter/material.dart';

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

class MyApp extends StatelessWidget {
 @override
 Widget build(BuildContext context) {
  return MaterialApp(
   title: 'Responsive UI Example',
   theme: ThemeData(
    primarySwatch: Colors.blue,
   ),
   home: ResponsiveHome(),
  );
 }
}

class ResponsiveHome extends StatelessWidget {
 @override
 Widget build(BuildContext context) {
  return Scaffold(
   appBar: AppBar(
      title: Text('Responsive UI Example'),
    ),
    body: LayoutBuilder(
      builder: (context, constraints) {
        // Determine the screen width
        if (constraints.maxWidth < 600) {
          // Mobile Layout
          return Column(
            children: [
              Expanded(child: _buildCard(Colors.red, 'Card 1')),
              Expanded(child: _buildCard(Colors.green, 'Card 2')),
              Expanded(child: _buildCard(Colors.blue, 'Card 3')),
            ],
          );
        } else {
          // Tablet/Desktop Layout
          return Row(
            children: [
              Expanded(child: _buildCard(Colors.red, 'Card 1')),
              Expanded(child: _buildCard(Colors.green, 'Card 2')),
              Expanded(child: _buildCard(Colors.blue, 'Card 3')),
            ],
          );
        }
      },
    ), );}

Widget _buildCard(Color color, String title) {
  return Card(
    color: color,
    child: Center(
      child: Text(
        title,
        style: TextStyle(color: Colors.white, fontSize: 24),
      ),), ); } }
```



3b) Implement media queries and breakpoints for responsiveness.

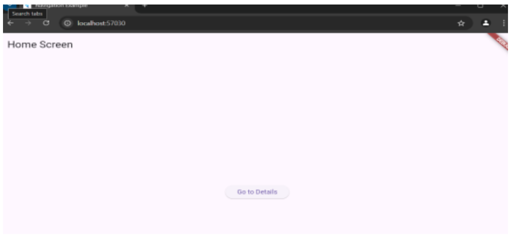
```
import 'package:flutter/material.dart';

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

class MyApp extends StatelessWidget{
  @override
  Widget build(BuildContext context){
    return MaterialApp(
      home: ResponsiveUI(),
    );
  }
}

class ResponsiveUI extends StatelessWidget{
  @override
  Widget build(BuildContext context){
    double screenWidth = MediaQuery.of(context).size.width;

    if (screenWidth < 600)
    {
      return Scaffold(
        appBar: AppBar(
          title: Text("MobileLayout"),
        ),
        body: Center(
      child: Container(
          width: screenWidth * 0.8,
          height: 150,
          color: Colors.yellowAccent,
          child: Center(
            child: Text("MobileLayout"),
          )  ),),
  );}
else if(screenWidth >=600 && screenWidth <1200){
  return Scaffold(
    appBar: AppBar(
      title: Text("TabletLayout"),
    ),
    body: Center(
      child: Container(
          width: screenWidth * 0.8,
          height: 150,
          color: Colors.blueAccent,
          child: Center(
            child: Text("TabletLayout"),
          )
      ),),);
}
else{
  return Scaffold(
    appBar: AppBar(
      title: Text("DesktopLayout"),
    ),
    body: Center(
      child: Container(
          width: screenWidth * 0.8,
          height: 150,
          color: Colors.blueAccent,
          child: Center(
            child: Text("DesktopLayout"),
          )         ),


      ),
    ),
  );
}
```

```
 }
}
```



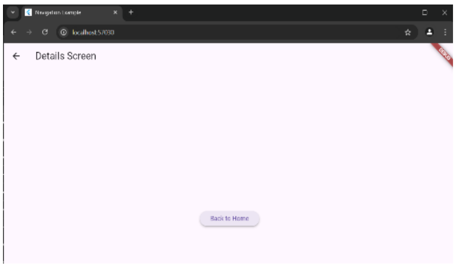
4. a) Set up navigation between different screens using Navigator.

```
import 'package:flutter/material.dart';
void main() => runApp(MaterialApp(
  title: 'Navigation Example',
  theme: ThemeData(primarySwatch: Colors.blue),
  home: HomeScreen(),
));
class HomeScreen extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(title: Text('Home Screen')),
      body: Center(
        child: ElevatedButton(
          onPressed: () => Navigator.push(
            context,
            MaterialPageRoute(builder: (context) => DetailsScreen()),
          ),
          child: Text('Go to Details'),
        ),
      ),
    );
  }
}
class DetailsScreen extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(title: Text('Details Screen')),
      body: Center(
        child: ElevatedButton(
          onPressed: () => Navigator.pop(context),
          child: Text('Back to Home'),
        ),
      ),
    );
  }
}
```



Home Screen

Go to Details



Details Screen

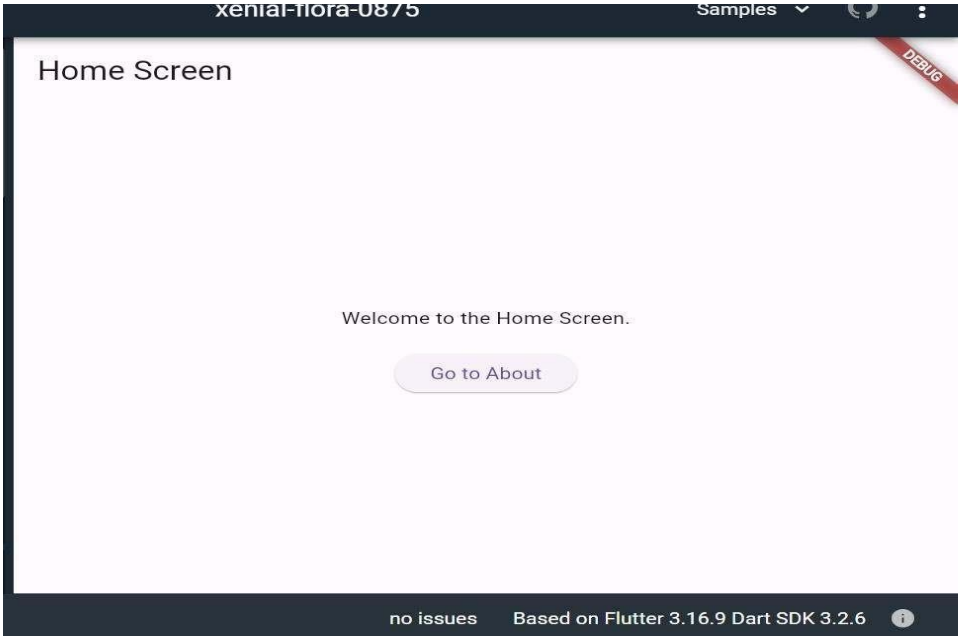
Back to Home

4b) 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'),

            ), ],),),

      } }
```

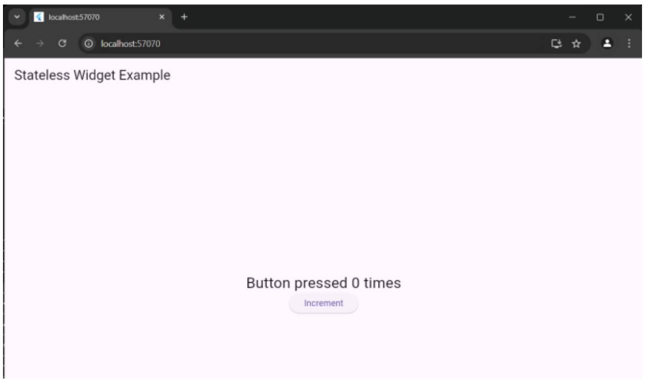
Output:



xenial-flora-0875
Samples
DEBUG
Home Screen
Welcome to the Home Screen.
Go to About
no issues
Based on Flutter 3.16.9 Dart SDK 3.2.6

5a.Learn about stateful and stateless widget Stateless 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('Stateless Widget Example')),
    body: CounterScreen(),
   ),
  );
 }
}

class CounterScreen extends StatelessWidget {

 @override
 Widget build(BuildContext context) {
  int _count = 0; // Counter state moved here.

  // Function to increment the counter (not possible inside a StatelessWidget)
  void _incrementCounter() {
   // This will be done in a stateful context typically.
   // Ideally, you'd use a callback or external state management to modify _count.
  }

  return Center(
   child: Column(
    mainAxisAlignment: MainAxisAlignment.center,
    children: [
     Text(
      'Button pressed $_count times',
      style: TextStyle(fontSize: 24),
     ),
     ElevatedButton(
      onPressed: _incrementCounter,
      child: Text('Increment'),
     ),
    ],
   ),
  );
 }
}
```

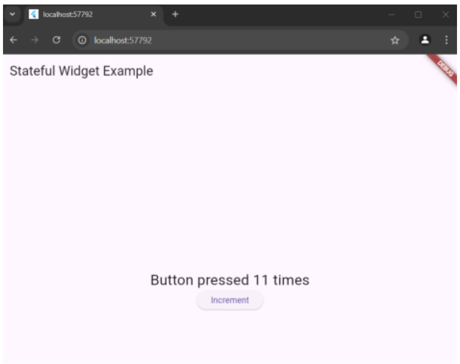


Stateless Widget Example

Button pressed 0 times

Stateful 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('Stateful Widget Example')),
        body: Counter(),
      ),
    );
  }
}

class Counter extends StatefulWidget {
  @override
  _CounterState createState() => _CounterState();
}

class _CounterState extends State<Counter> {
  int _count = 0; // Mutable state

  void _incrementCounter() {
    setState(() {
      _count++; // Updating the state
    });
  }
  @override
  Widget build(BuildContext context) {
    return Center(
      child: Column(
        mainAxisAlignment: MainAxisAlignment.center,
        children: [
          Text(
            'Button pressed $_count times',
            style: TextStyle(fontSize: 24),
          ),
          ElevatedButton(
            onPressed: _incrementCounter,
            child: Text('Increment'),
          ),
        ],
      ),
    );
  }
}
```



Stateful Widget Example

Button pressed 11 times

5b. Implement state management using set State and Provider.

<u>Pubspec.yaml</u>

```
dependencies:
  flutter:
    sdk: flutter
  provider: ^6.0.0
  cupertino_icons: ^1.0.8
```

<u>counter.dart</u>

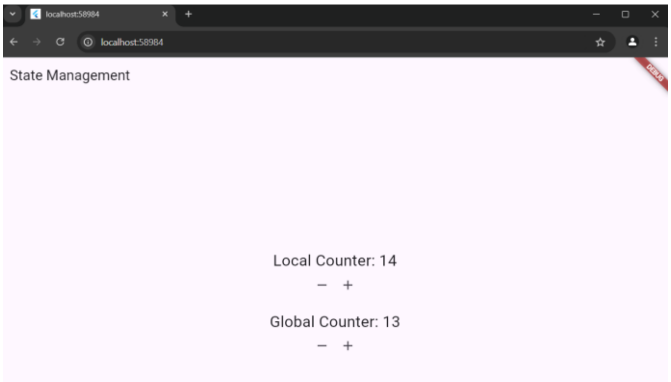
```
import 'package:flutter/material.dart';
import 'package:provider/provider.dart';

class GlobalCounter with ChangeNotifier {
  int _counter = 0;

  int get counter => _counter;

  void increment() {
    _counter++;
    notifyListeners();
  }

  void decrement() {
    _counter--;
    notifyListeners();
  }
}
```

main.dart:

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


class GlobalCounter with ChangeNotifier {
  int _counter = 0;
  int get counter => _counter;
  void increment() => _counter++;
  void decrement() => _counter--;
}

void main() => runApp(
  ChangeNotifierProvider(
    create: (_) => GlobalCounter(),
    child: MyApp(),
  ),
);

class MyApp extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return MaterialApp(home: HomePage());
  }
}
```

```
class HomePage extends StatefulWidget {
  @override
  _HomePageState createState() => _HomePageState();
}

class _HomePageState extends State<HomePage> {
  int _localCounter = 0;

  @override
  Widget build(BuildContext context) {
    final globalCounter = Provider.of<GlobalCounter>(context);

    return Scaffold(
      appBar: AppBar(title: Text("State Management")),
      body: Center(
        child: Column(
          mainAxisAlignment: MainAxisAlignment.center,
          children: [
            Text('Local Counter: $_localCounter', style: TextStyle(fontSize: 24)),
            Row(
              mainAxisAlignment: MainAxisAlignment.center,
              children: [
                IconButton(icon: Icon(Icons.remove), onPressed: () => setState(() =>
_localCounter--)),
                IconButton(icon: Icon(Icons.add), onPressed: () => setState(() =>
_localCounter++)),
              ],
            ),
            SizedBox(height: 20),
            Text('Global Counter: ${globalCounter.counter}', style: TextStyle(fontSize: 24)),
            Row(
              mainAxisAlignment: MainAxisAlignment.center,
              children: [
                IconButton(icon: Icon(Icons.remove), onPressed: globalCounter.decrement),
                IconButton(icon: Icon(Icons.add), onPressed: globalCounter.increment),
              ],
            ),
          ],
        ),
      ),
    );
  }
}
```

output:



State Management

Local Counter: 14

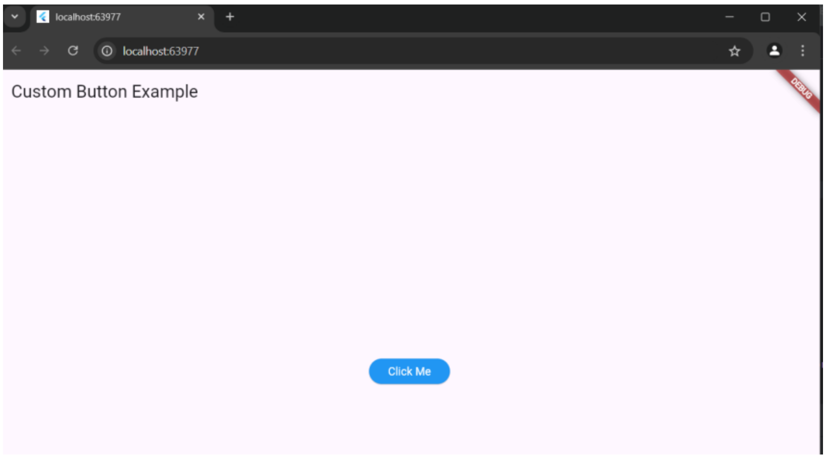
− +

Global Counter: 13

− +

6a) Create custom widgets for specific UI elements.

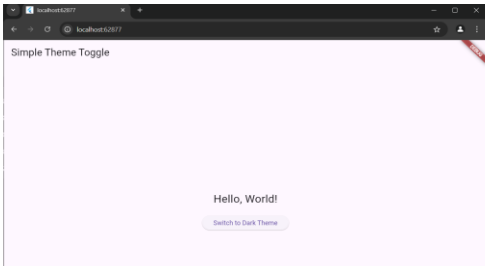
```
import 'package:flutter/material.dart';
class CustomButton extends StatelessWidget {
  final String text;
  final Function onPressed;
  final Color buttonColor;
  final Color textColor;
  CustomButton({
    required this.text,
    required this.onPressed,
    this.buttonColor = Colors.blue,
    this.textColor = Colors.white,
  });
  @override
  Widget build(BuildContext context) {
    return ElevatedButton(
      onPressed: () => onPressed(),
      style: ButtonStyle(
        backgroundColor: MaterialStateProperty.all(buttonColor),
        foregroundColor: MaterialStateProperty.all(textColor),
      ),
      child: Text(text),
    );
  }
}
class CustomAlertDialog extends StatelessWidget {
  final String title, message, positiveButtonText, negativeButtonText;
  final Function onPositivePressed, 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: [
        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: () {
              print('Button Pressed');
            },
        ),
       ),
      ),
    );
  }
}
```

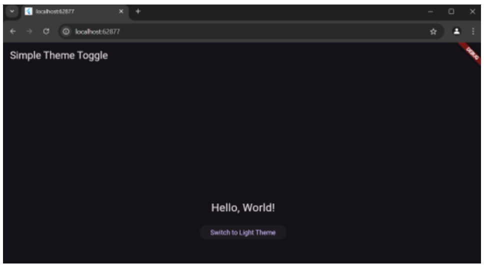


b) Apply styling using themes and custom styles.

6b. Apply styling using themes and custom styles

```
import 'package:flutter/material.dart';
void main() {
  runApp(MyApp());
}
class MyApp extends StatefulWidget {
  @override
  _MyAppState createState() => _MyAppState();
}
class _MyAppState extends State<MyApp> {
  bool _isDarkMode = false; // Track theme state
  @override
  Widget build(BuildContext context) {
    return MaterialApp(
// Toggle between light and dark themes based on _isDarkMode
      theme: _isDarkMode ? ThemeData.dark() : ThemeData.light(),
      home: Scaffold(
        appBar: AppBar(
          title: Text('Simple Theme Toggle'),
        ),
        body: Center(
          child: Column(
            mainAxisAlignment: MainAxisAlignment.center,
            children: <Widget>[
              Text(
        'Hello, World!',
        style: TextStyle(fontSize: 24),
       ),
       SizedBox(height: 20),
       ElevatedButton(
        onPressed: () {
         setState(() {
          _isDarkMode = !_isDarkMode; // Toggle theme
         });
        },
        child: Text(
         _isDarkMode ? 'Switch to Light Theme' : 'Switch to Dark Theme',
        ),
       ),
      ],
     ),
    ),
   ),
  );
 }
}
```


localhost:62877
Simple Theme Toggle
Hello, World!
Switch to Dark Theme


localhost:62877
Simple Theme Toggle
Hello, World!
Switch to Light Theme