EXPERIMENT – 3

AIM:
a) Design a responsive UI that adapts to different screen sizes.
b) Implement media queries and breakpoints for responsiveness.

DESCRIPTION:

LayoutBuilder:
In Flutter, LayoutBuilder is a powerful widget used to build a widget tree based on the constraints of its parent widget. It is particularly useful for creating responsive layouts that adapt to different screen sizes, orientations, or available space.

LayoutBuilder takes a builder function as a parameter. This function provides two arguments:

- BuildContext context: The build context of the widget.
- BoxConstraints constraints: An object containing the maximum and minimum width and height constraints passed down from the parent widget.

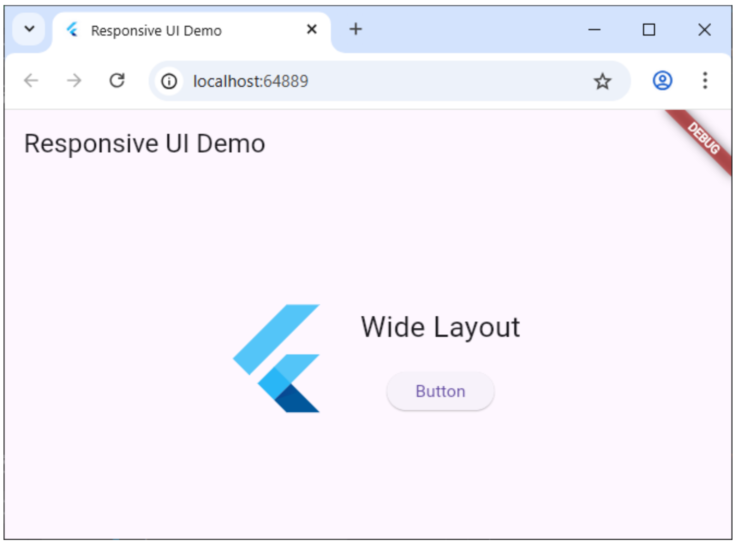
Inside the builder function, you can access the maxWidth, minWidth, maxHeight, and minHeight properties of the BoxConstraints object. This information allows you to conditionally render different layouts or adjust widget properties based on the available space.

<u>Experiment - 3(a)</u>

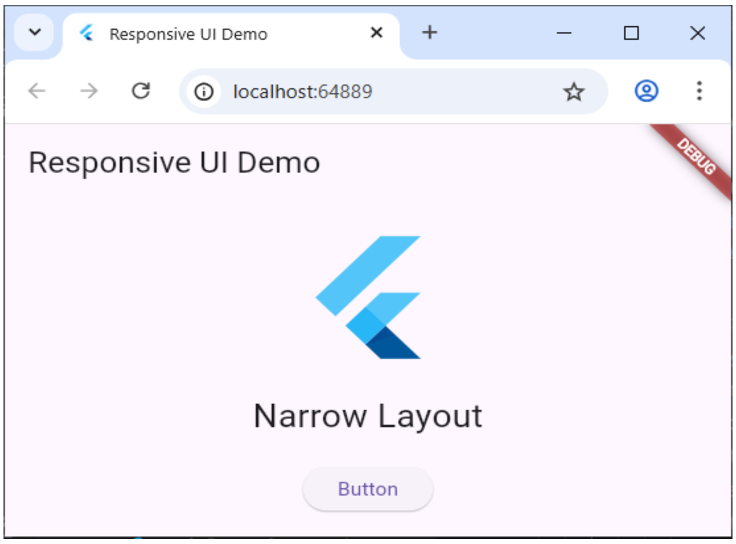
AIM: Design a responsive UI that adapts to different screen sizes.

SOLUTION:
```
import 'package:flutter/material.dart';
void main() {
  runApp(MyApp());
}
class MyApp extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      title: 'Responsive UI Demo',
      theme: ThemeData(
        primarySwatch: Colors.blue,
      ),
      home: ResponsiveHomePage(),
    );
  }
}
class ResponsiveHomePage extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
        title: Text('Responsive UI Demo'),
      ),
      body: LayoutBuilder(
        builder: (BuildContext context, BoxConstraints constraints) {
          if (constraints.maxWidth < 600) {
            return _buildNarrowLayout();
          } else {
            return _buildWideLayout();
          }
        },
      ),
    );
  }

  Widget _buildNarrowLayout() {
    return Center(
      child: Column(
        mainAxisAlignment: MainAxisAlignment.center,
        children:
        <Widget>[
        FlutterLogo(size: 100),
        SizedBox(height: 20),
        Text('Narrow Layout',
          style: TextStyle(fontSize: 24),
        ),
        SizedBox(height: 20),
        ElevatedButton( onPressed:
            () {},
          child: Text('Button'),
        ),
      ],
      ),
    );
  }

  Widget _buildWideLayout() {
    return Center(
      child: Row(
        mainAxisAlignment: MainAxisAlignment.center,
        children: <Widget>[
          FlutterLogo(size: 100),
          SizedBox(width: 20),
          Column(
            mainAxisAlignment: MainAxisAlignment.center,
            children: <Widget>[
              Text(
                'Wide Layout',
          style: TextStyle(fontSize: 24),
         ),
         SizedBox(height: 20),
         ElevatedButton( onPressed: ()
         {},
          child: Text('Button'),
         ),
        ],
       ),
      ],
     ),
    );
  }
}
```

OUTPUTS:
DESKTOP VIEW



MOBILE VIEW



Experiment - 3(b)

AIM: Implement media queries and breakpoints for responsiveness.

DESCRIPTION:
In Flutter, **MediaQuery** is a widget that provides information about the device’s screen and user preferences. It allows developers to build responsive and adaptive user interfaces by giving access to various details about the screen’s size, orientation, pixel density, and more.

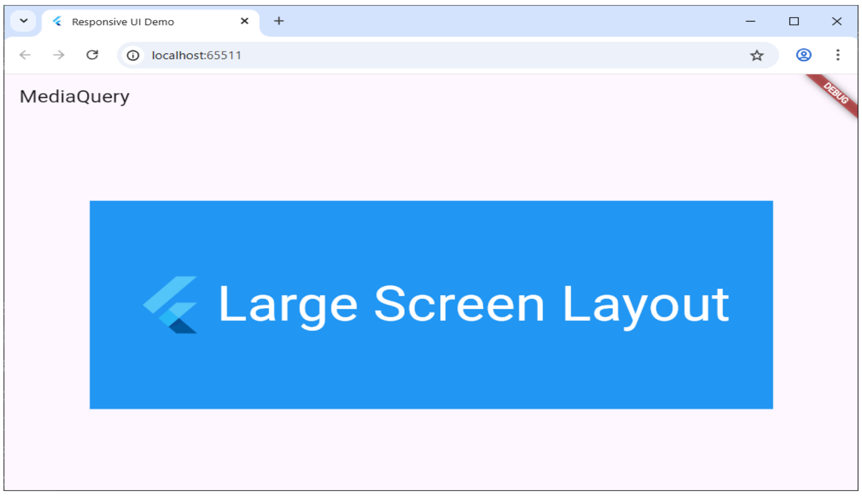
It is quite easy to use MediaQuery, just use the ***of()*** method and you are good to go, the only requirement is **BuildContext.** MediaQuery can be used anywere in the widget tree where the **BuildContext** is available.

Next, in order to access the data provided by **MediaQuery.of(context)** you can create a variable of type **MediaQueryData,** let’s say **mediaQuery**. Now once we have instance of MediaQueryData, just use “ **mediaQuery. ”** and you will be able to see everything MediaQueryData has to offer.(Assuming you are using vscode, Android Studio or IntelliJ IDE)

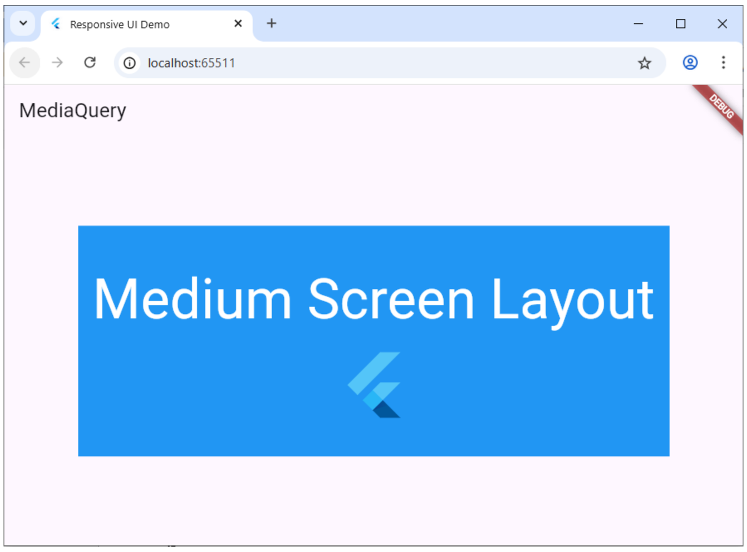
SOLUTION:
```
import 'package:flutter/material.dart';
void main() {
  runApp(MyApp());

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

class MediaQueryWidget extends StatelessWidget {
  const MediaQueryWidget({super.key});
  @override
  Widget build(BuildContext context) {
    // Use media query to get the screen size
    var screenSize = MediaQuery.of(context).size;

    // Define breakpoints for different screen sizes
    double breakpointSmall = 600.0;
    double breakpointMedium = 900.0;

    // Choose the appropriate layout based on screen width
    Widget content;
    if (screenSize.width < breakpointSmall) {
      content = buildSmallLayout();
    } else if (screenSize.width < breakpointMedium) {
      content = buildMediumLayout();
    } else {
      content = buildLargeLayout();
    }

    return SafeArea(
      child: Scaffold(
        appBar: AppBar(
          title: Text('MediaQuery'),
        ),
        body: Center(
          child: Container(
            width: screenSize.width * 0.8,
            height: screenSize.height * 0.5,
            color: Colors.blue,
            child: Center(child: content),
          ),
        ),
      ),
    );
  }

  Widget buildSmallLayout() {
    return const Text(
      'Small Screen Layout',
      textAlign: TextAlign.center,
      style: TextStyle(fontSize: 40, color: Colors.white),
    );
  }

  Widget buildMediumLayout() {
    return Column(
      mainAxisAlignment: MainAxisAlignment.center,
      children: [
        Text(
          'Medium Screen Layout',
          textAlign: TextAlign.center,
          style: TextStyle(fontSize: 60, color: Colors.white),
        ),
        SizedBox(height: 10.0),
        FlutterLogo(size: 80),
      ],
    );
  }

  Widget buildLargeLayout() {
    return Row(
      mainAxisAlignment: MainAxisAlignment.center,
      children: [
        FlutterLogo(size: 80),
        SizedBox(width: 10.0),
        Text(
          'Large Screen Layout',
          textAlign: TextAlign.center,
          style: TextStyle(fontSize: 60, color: Colors.white),
        ),
      ],
    );
  }
}
```

OUTPUTS:

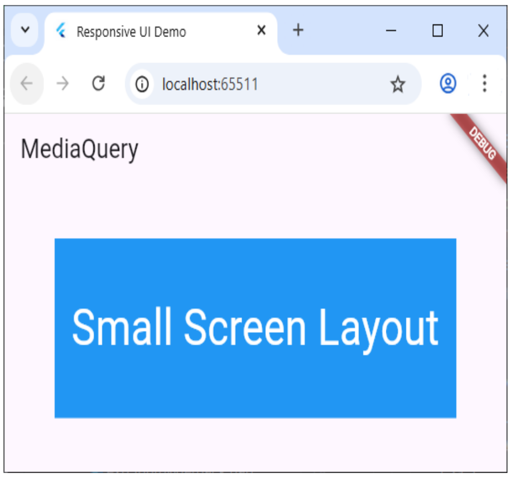
DESKTOP SCREEN SIZE



TABLET SCREEN SIZE



MOBILE SCREEN SIZE


Responsive UI Demo
localhost:65511
MediaQuery
DEBUG
Small Screen Layout