

## WEEK-3

**Aim:**

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

**Description:**

- A responsive UI design ensures a website or application adapts seamlessly to various screen sizes, providing an optimal user experience across all devices. This approach uses flexible layouts, images, and media queries to adjust content and elements based on the user's screen size, resolution, and orientation.
- Responsive/adaptive layouts provide an optimized user experience regardless of screen size. Implement responsive/adaptive layouts to enable your view-based app to support all display sizes, orientations, and configurations, including resizable configurations such as [multi-window mode](#).

**Source Code:**

```
import 'package:flutter/material.dart';
void main() => runApp(
  MaterialApp(
    home:Scaffold(
      appBar: AppBar(title:const Text('Simple Responsive UI')),
      body:LayoutBuilder(
        builder:(context,constraints){
          final width= constraints.maxWidth;
          String text= 'Mobile View';
          IconData icon=Icons.phone_android;
          double fontSize =24;

          if(width >= 600 && width <1200){
            text='Tablet View';
            icon = Icons.tablet;
            fontSize=28;
          }
          else if(width>=1200){
            text='Desktop View';
            icon=Icons.desktop_windows;
            fontSize = 32;
          }
          return Center(
            child:Column(
              mainAxisAlignment:MainAxisSize.min,
              children:[
                Icon(icon,size:80),
                const SizedBox(height:20),
                Text(text,style:TextStyle(fontSize: fontSize)),
```

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

### Output:

Simple Responsive UI



Mobile View

Simple Responsive UI



Tablet View

Simple Responsive UI



Desktop View

Feedback 

Dart 3.8.1 • Flutter 3.32.6  Stable channel

Date:

**Aim:**

- b) Implement media queries and breakpoints for responsiveness.

**Description:**

- Implementing media queries and breakpoints is crucial for creating responsive websites that adapt to different screen sizes and devices. Media queries allow developers to apply specific CSS styles based on the characteristics of the user's device, such as screen width, height, or orientation.
- Breakpoints, defined within media queries, are specific screen widths at which the website's layout changes to provide an optimal viewing experience.
- Media queries are a feature of CSS that allows you to apply styles based on the characteristics of the device being used to view the website.

**Source Code:**

```
import 'package:flutter/material.dart';
void main() => runApp(
  MaterialApp(
    home:Scaffold(
      appBar: AppBar(title:const Text('Responsive with MediaQuery')),
      body:Builder(
        builder:(context){
          final width= MediaQuery.of(context).size.width;
          String text= 'Small Screen';
          IconData icon=Icons.phone_android;
          double fontSize =24;

          if(width >= 600 && width <1200){
            text='Medium Screen(Tablet)';
            icon = Icons.tablet;
            fontSize=28;
          }
          else if(width>=1200){
            text='Large Screen(Desktop)';
            icon=Icons.desktop_windows;
            fontSize = 32;
          }
          return Center(
            child:Column(
              mainAxisAlignment:MainAxisSize.min,
              children:[
                Icon(icon,size:80),
                const SizedBox(height:20),
                Text(text,style:TextStyle(fontSize: fontSize)),
                const SizedBox(height:10),
                Text('Screen width: ${width.toStringAsFixed(0)} px'),
```

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

**Output:**

Responsive with MediaQuery



Small Screen

Screen width: 500 px

Responsive with MediaQuery



Medium Screen(Tablet)

Screen width: 740 px

Responsive with MediaQuery



Large Screen(Desktop)

Screen width: 1342 px