

# Internship Report – Frontend Dev(Week3)

---

**Name: Zainab**

---

**Father Name: Assad Qayyum**

---

**Date: 8<sup>th</sup> July, 2025**

---

**Internship Domain: Front-end Intern**

---

**Task: CSS Media Queries: Responsive Design Techniques**

---

## Task Overview: (Day2)

Today's task was focused on understanding and applying **CSS Media Queries** as part of responsive design techniques. The objective was to learn how to make web pages adapt to different screen sizes and devices using CSS only.

## Content Covered:

- Introduction to Responsive Web Design
- Purpose and use of Media Queries
- Syntax and key parts of media queries
- Hands-on coding practice with various screen sizes

## Introduction:

Modern websites must look good on all devices—from desktops to smartphones. That’s where responsive design comes in. **Media Queries** in CSS help developers adapt the **layout and styling** of web content depending on the **screen size, device type, or orientation**.

## CSS Media Queries:

**Media Queries** are CSS techniques used to apply styles **only when certain conditions are true**, such as screen width, device type, or orientation. They help create a fluid, **responsive design** that adjusts to various screen sizes.

Media queries can be used to check many things, such as:

- width and height of the viewport
- orientation of the viewport (landscape or portrait)
- resolution

## Syntax:

```
@media (condition) {  
  
  /* CSS styles that apply if the condition is true */  
}
```

## Common Media features:

Value	Description
orientation	Orientation of the viewport. Landscape or portrait
max-height	Maximum height of the viewport
min-height	Minimum height of the viewport
height	Height of the viewport (including scrollbar)
max-width	Maximum width of the viewport
min-width	Minimum width of the viewport
width	Width of the viewport (including scrollbar)

**Example:** If the screen is 600 pixels wide or less, change the body background to light blue.

```
@media (max-width: 600px) {  
  body {  
    background-color: lightblue;  
  }  
}
```

## Key Parts of Media Query:

Here are the main things media queries include:

### 1. Media Type (optional)

It includes the media type – e.g., screen, print

### 2. Media Features:

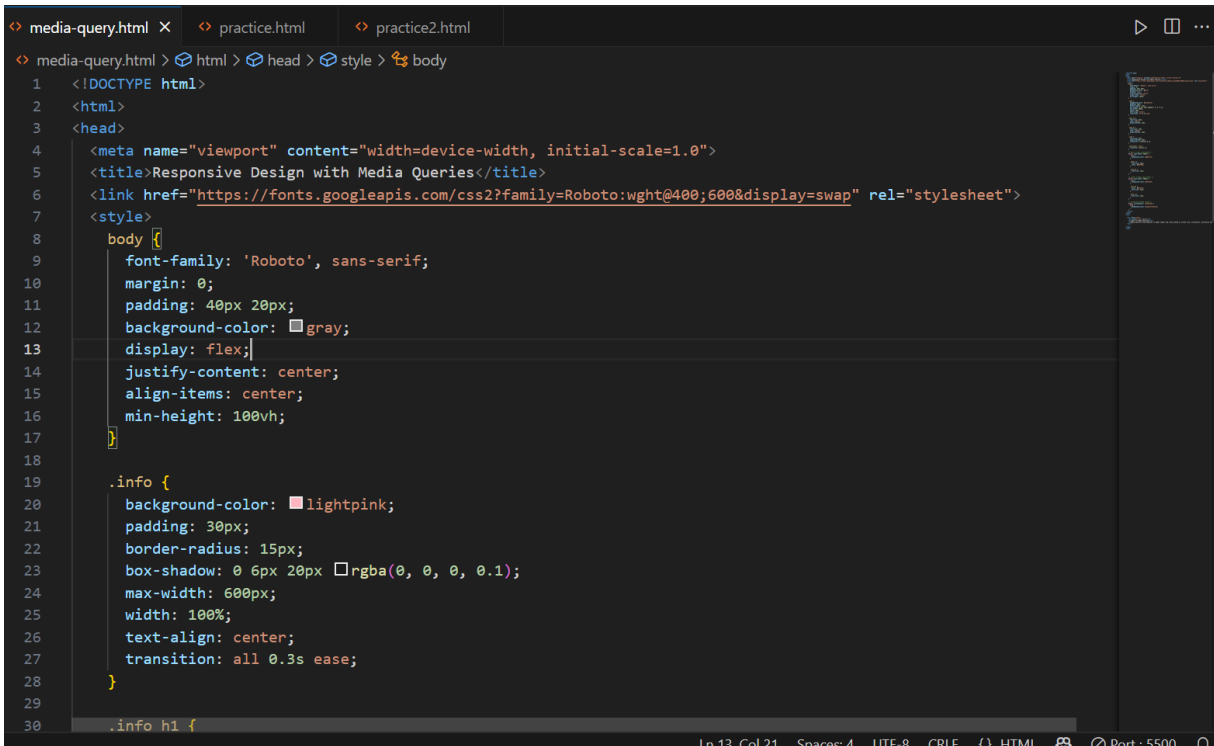
These are the conditions:

- **max-width:** applies when screen width is equal to or less than a value
- **min-width:** applies when screen width is equal to or more than a value
- **orientation:** landscape or portrait
- **max-height** or **min-height**

## Common Usage Scenarios

- Adjust layout and grid/flex direction
- Resize fonts and images
- Hide or show elements on mobile
- Change background or colors

## Practice Code:



```
media-query.html X practice.html practice2.html
media-query.html > html > head > style > body
1 <!DOCTYPE html>
2 <html>
3 <head>
4   <meta name="viewport" content="width=device-width, initial-scale=1.0">
5   <title>Responsive Design with Media Queries</title>
6   <link href="https://fonts.googleapis.com/css2?family=Roboto:wght@400;600&display=swap" rel="stylesheet">
7   <style>
8     body {
9       font-family: 'Roboto', sans-serif;
10      margin: 0;
11      padding: 40px 20px;
12      background-color: gray;
13      display: flex;
14      justify-content: center;
15      align-items: center;
16      min-height: 100vh;
17    }
18
19    .info {
20      background-color: lightpink;
21      padding: 30px;
22      border-radius: 15px;
23      box-shadow: 0 6px 20px rgba(0, 0, 0, 0.1);
24      max-width: 600px;
25      width: 100%;
26      text-align: center;
27      transition: all 0.3s ease;
28    }
29
30    .info h1 {
```

```
media-query.html X practice.html practice2.html
media-query.html > html > head > style > body
2 <html>
3 <head>
7 <style>
19 .info {
28 }
29
30 .info h1 {
31   font-size: 28px;
32   color: #333;
33   margin-bottom: 10px;
34 }
35
36 .info p {
37   font-size: 18px;
38   color: #555;
39   margin-bottom: 15px;
40 }
41
42 .icon {
43   font-size: 40px;
44   margin-bottom: 10px;
45   transition: transform 0.3s;
46 }
47
48 .info:hover .icon {
49   transform: scale(1.2);
50 }
51
52 /* Small screen (max-width) */
53 @media (max-width: 500px) {
```

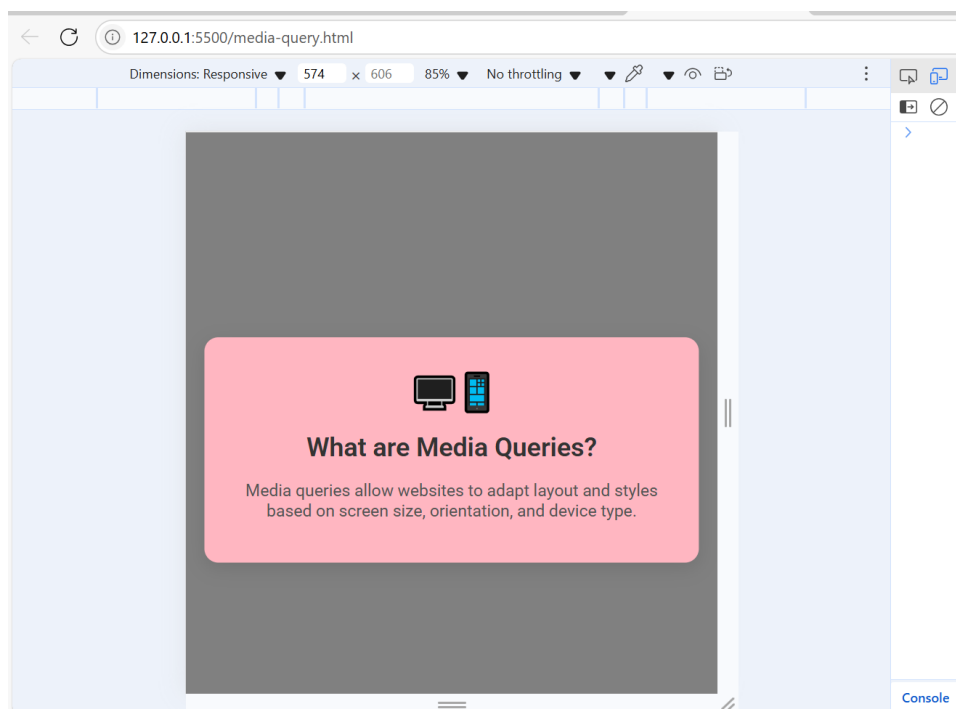
```
media-query.html X practice.html practice2.html
media-query.html > html > head > style > body
2 <html>
3 <head>
7 <style>
52 /* Small screen (max-width) */
53 @media (max-width: 500px) {
54   .info {
55     background-color: #e0f7fa;
56   }
57
58   .info p {
59     font-size: 16px;
60     color: #00796b;
61   }
62
63   .info h1 {
64     font-size: 24px;
65   }
66 }
67
68 /* Larger screens (min-width) */
69 @media (min-width: 800px) {
70   .info {
71     background-color: #f3e5f5;
72   }
73
74   .info h1 {
75     color: #6a1b9a;
76     font-size: 32px;
77   }
78 }
```

```
media-query.html X practice.html practice2.html
media-query.html > html > head > style > body
2 <html>
3 <head>
7 <style>
69 @media (min-width: 800px) {
74 .info h1 {
77 }
78
79 .info p {
80 font-size: 19px;
81 }
82 }
83
84 /* Orientation-based change */
85 @media (orientation: landscape) {
86 body {
87 background-color: lightsteelblue;
88 }
89 }
90 </style>
91 </head>
92 <body>
93
94 <div class="info">
95 <div class="icon"><img alt="Icon of a monitor and a smartphone" data-bbox="455 745 505 765"/></div>
96 <h1>What are Media Queries?</h1>
97 <p>Media queries allow websites to adapt layout and styles based on screen size, orientation, and device type.</p>
98 </div>
99
100 </body>
101 </html>
```

## Explanation:

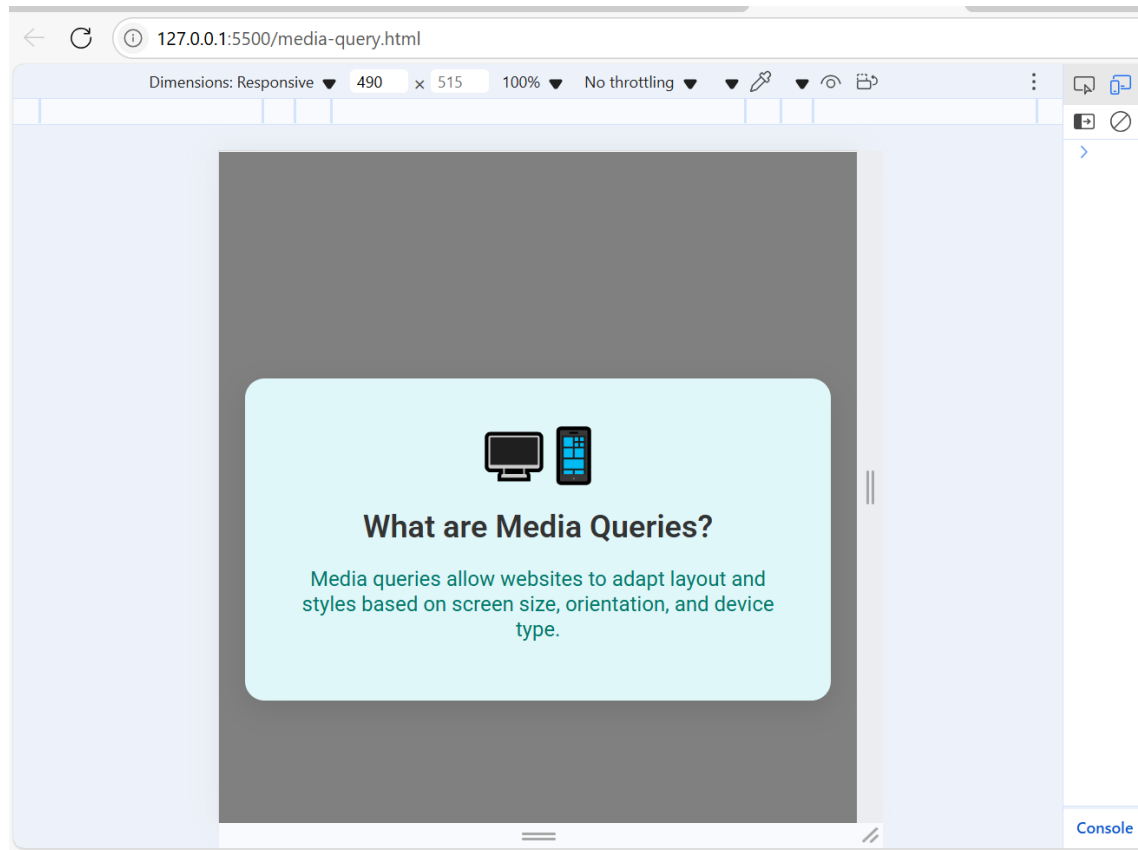
Base Style (All Screens) includes:

- Gray background, centered pink info box
- Includes icon, heading, and paragraph
- Hovering icon slightly scales up



### On Small Screens (**max-width: 500px**)

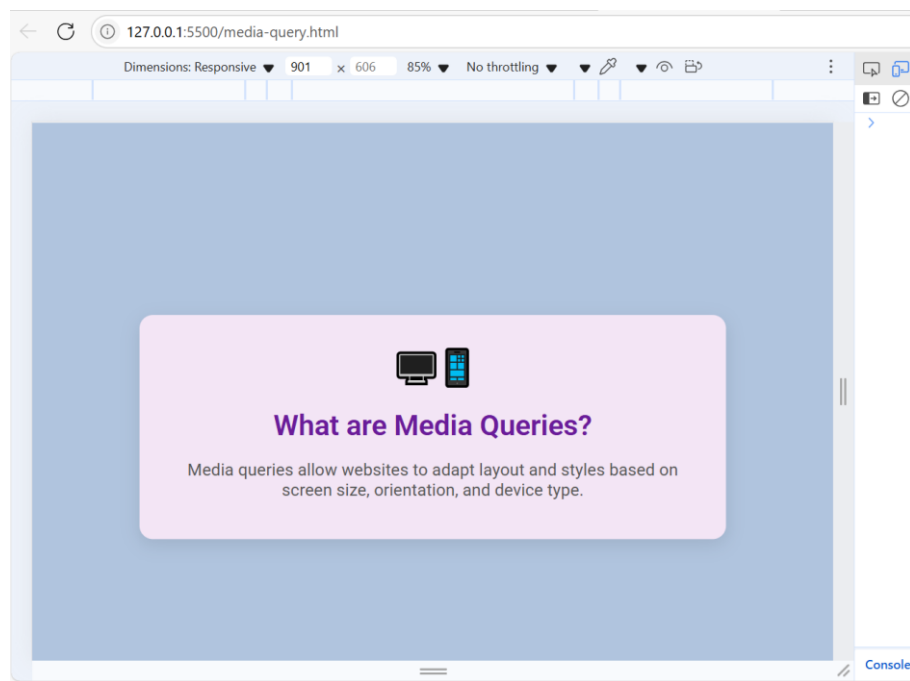
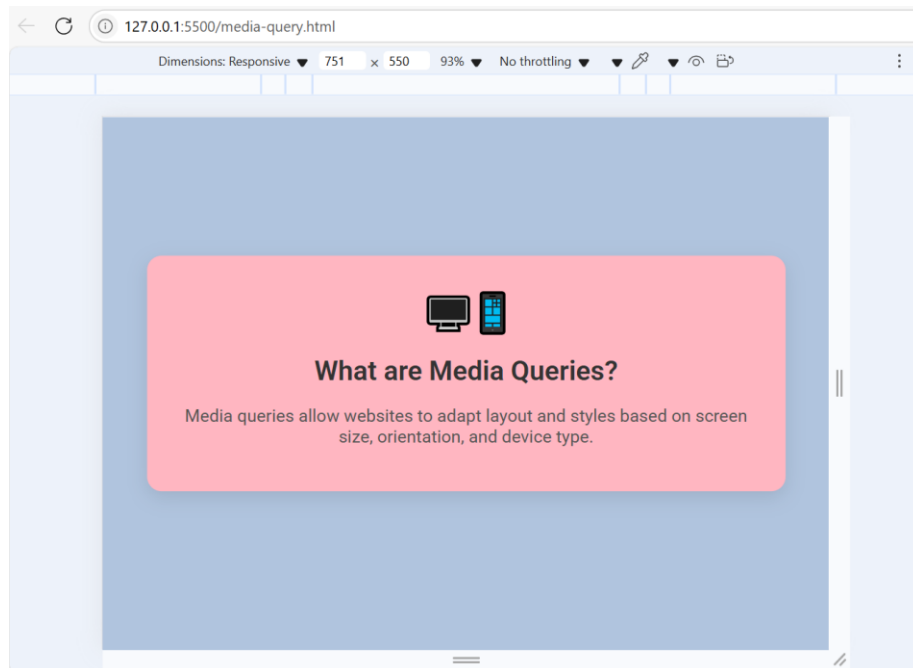
- Info box background: light teal
- Text color: dark teal
- Smaller font sizes



### On Large Screens (**min-width: 800px**)

- Info box background: light purple
- Heading becomes larger and purple
- Slightly bigger paragraph text

**Landscape Orientation:** Page background changes to light steel blue



### Conclusion:

Media queries are essential in creating websites that are flexible and user-friendly on all devices. They let us write conditional CSS that respond to different screen sizes and orientations. Practicing with different breakpoints and layouts helps build an intuitive understanding of how to design responsively.