

## **Report Day: 3.2**

**Submitted By: Ayesha Abid**

**Submitted To: Mr. Ali Hyder Hidayat**

**Group: Frontend Development @ProSensia**

**Dated: 08/07/25**

### **Learning Outcomes:**

#### **Responsive Web Design:**

Responsive web design makes your web page look good on all devices. Responsive web design uses only HTML and CSS.

#### **Setting the Viewport:**

The viewport is the user's visible area of a web page. We should include the following <meta> viewport element in the <head> section of all our web pages.

The width=device-width part sets the width of the page to follow the screen-width of the device (which will vary depending on the device).

#### **Media Query:**

A **media query** lets you apply CSS **only when certain conditions are true**, like:

- Screen width is smaller than 600px (mobile)
- Device is in landscape mode
- Screen resolution is high

#### **Basic Syntax:**

```
@media (condition) {
```

```
/* CSS rules here */}
```

- Media queries can help to add a breakpoint where certain parts of the design will behave differently on each side of the breakpoint.
- Media queries can also be used to change layout of a page depending on the orientation of the browser.
- You can also use media queries to change the font size of an element on different screen size

```

1  <!DOCTYPE html>
2  <html>
3  <head>
4  <meta name="viewport" content="width=device-width, initial-scale=1.0">
5  <style>
6  body {
7    background-color: lightgreen;
8  }
9
10 @media only screen and (max-width: 600px) {
11   body {
12     background-color: lightblue;
13   }
14 }
15 </style>
16 </head>
17 <body>
18
19 <p>Resize the browser window. When the width of this document is 600 pixels or less, the background-color i
20
21 </body>
22 </html>

```

```

1  <<!DOCTYPE html>
2  <html>
3  <head>
4  <meta name="viewport" content="width=device-width, initial-scale=1.0">
5  <style>
6  * {
7    margin: 0;
8    box-sizing: border-box;
9  }
10
11 body {
12   font-family: "Lucida Sans", sans-serif;
13 }
14
15 .grid-container {
16   display: grid;
17   grid-template-areas:
18     'header header header header header header'
19     'menu main main main main right'
20     'footer footer footer footer footer footer';
21   gap: 10px;
22   background-color: white;
23   padding: 10px;
24 }
25
26 .grid-container > div {
27   padding: 10px;
28   font-size: 16px;
29 }

```

```

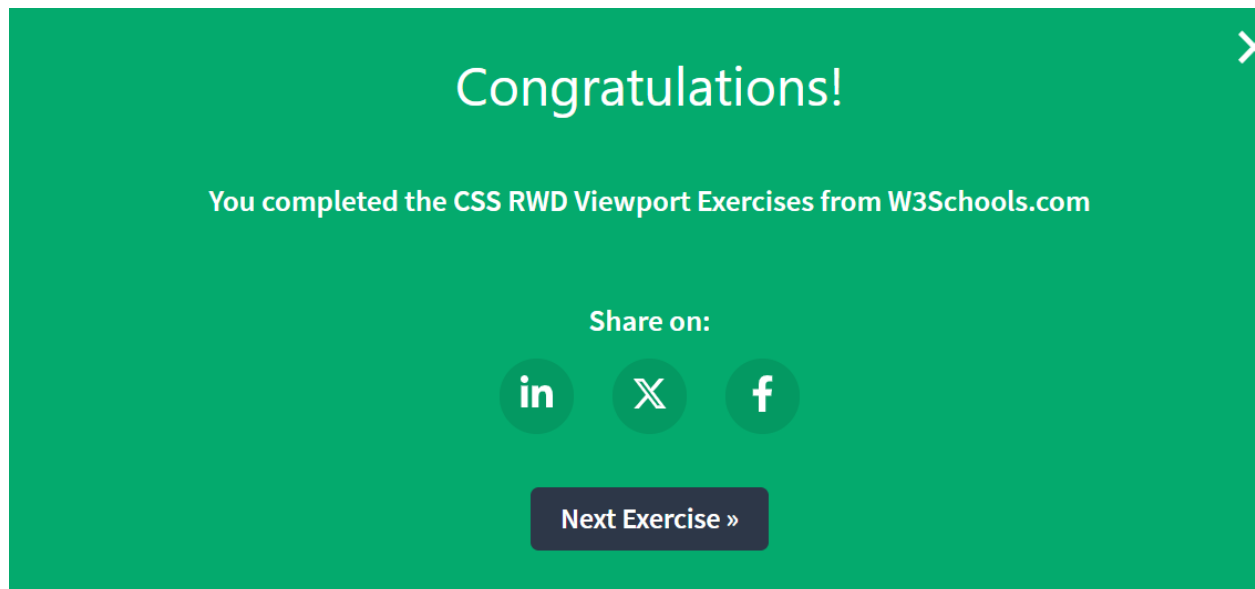
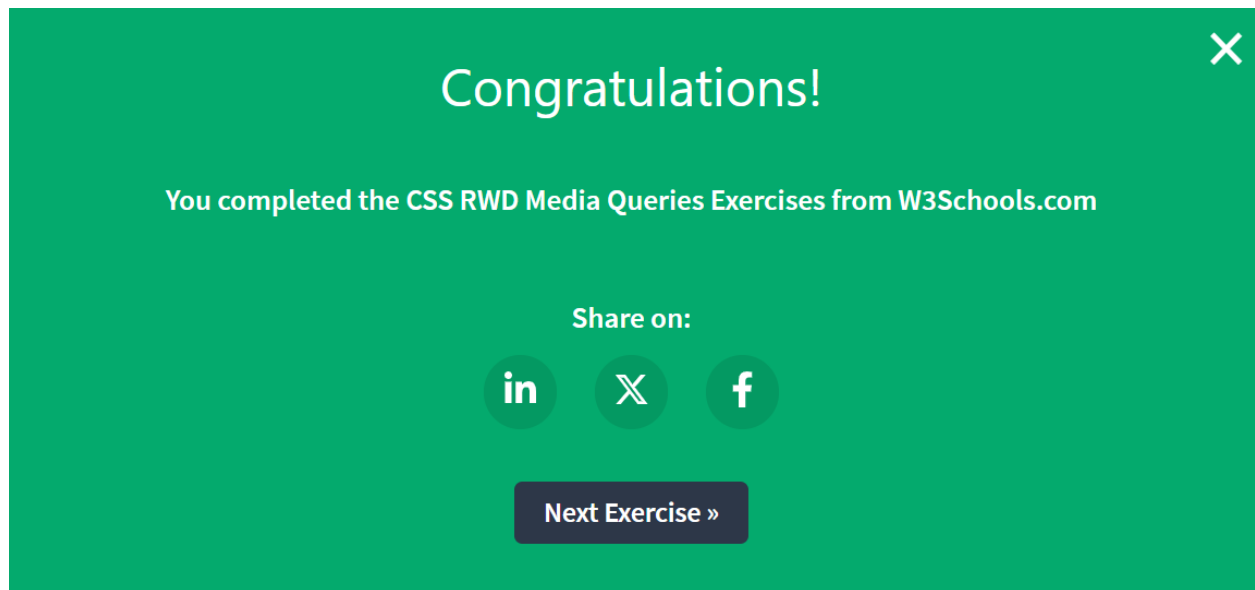
5   <style>
40  }
41
42  .item2 {
43    grid-area: menu;
44  }
45
46  .item2 ul {
47    list-style-type: none;
48    margin: 0;
49    padding: 0;
50  }
51
52  .item2 li {
53    padding: 8px;
54    margin-bottom: 7px;
55    background-color: #33b5e5;
56    color: #ffffff;
57  }
58
59  .item2 li:hover {
60    background-color: #0099cc;
61  }
62
63  .item3 {
64    grid-area: main;
65  }
66

```

```

101  @media only screen and (max-width: 600px) {
102    .item1 {grid-area: 1 / span 6;}
103    .item2 {grid-area: 2 / span 6;}
104    .item3 {grid-area: 3 / span 6;}
105    .item4 {grid-area: 4 / span 6;}
106    .item5 {grid-area: 5 / span 6;}
107  }
108  </style>
109  </head>
110  <body>
111
112  <div class="grid-container">
113
114    <div class="item1"><h1>Chania</h1></div>
115
116    <div class="item2">
117      <ul>
118        <li>The Flight</li>
119        <li>The City</li>
120        <li>The Island</li>
121        <li>The Food</li>
122      </ul>
123    </div>
124
125    <div class="item3">

```



### Conclusion:

Today we have learned about media queries and responsive web design elements as media queries are a powerful feature in CSS that allow us to make our website **responsive** meaning it can **adjust its layout and styles** based on different screen sizes, devices, or orientations.