

4	<p>To create a webpage that offers a seamless user experience across devices, responsive design principles must be applied. This involves using fluid grids, flexible images, and media queries to ensure the layout adjusts gracefully to different screen sizes. For instance, on a desktop, the webpage might display multiple columns with detailed content, while on a smartphone, the same content could stack vertically for easy scrolling. Navigation menus should transform into a hamburger icon on smaller screens, ensuring they remain accessible without taking up too much space. Additionally, touch-friendly elements and optimized images ensure fast loading times and a smooth experience on all devices.</p>	10	CO1	3
5	<p>Given a scenario (e.g., creating a blog post, a product listing), design a webpage using appropriate elements, tables, lists, and images for optimal readability and user experience. Also, describe step-by-step the sequence of HTTP requests and responses that occur when a user accesses a webpage containing multiple resources (HTML, CSS, JavaScript, images).</p>	10	CO1	3

## ASSIGNMENT - 2

4. To Create a webpage that ensure a seamless user experience across devices, responsive design principles must be applied.

Fluid grid:

Fluid grid use percentage - based widths rather than fixed pixel values, this allows your layout to adapt to different screen sizes gracefully.

Example CSS:

```
container {  
  display: grid;  
  grid-template-columns: repeat(3, 1fr);  
  gap: 20px;  
  padding: 20px;  
}
```

flexible Images:

Images should resize according to the container they're in maintaining their aspect ratio while adapting to various screen sizes.

Example: CSS

```
img {
```

```
  max-width: 100%;
```

```
  height: auto;
```

```
}
```

## Media Queries

Media queries are essential for adjusting the layout based on the device's screen size.

Example: CSS

```
@media (max-width: 768px) {  
  .container {  
    grid-template-columns: 1fr;  
  }  
}
```

Responsive Navigation Menus: On smaller screens, it's crucial to transform navigation menus into a more space-efficient design like a hamburger menu.

Eg.

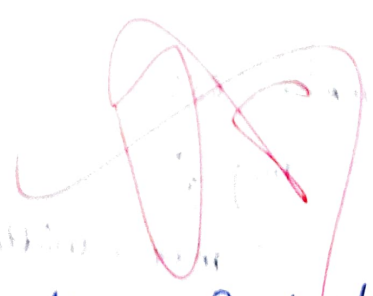
html

```
<nav class="nav-bar">  
  <div class="hamburger" onclick="toggleMenu()">#9776</div>  
  <ul class="Menu">  
    <li><a href="#">Home</a></li>  
  </ul>  
</nav>
```

Javascript for toggling:

Javascript

```
function toggleMenu() {  
  document.querySelector(".menu").classList.  
    toggle('active');  
}
```



5. Given a Scenario (eg. Creating a Blog post a product listing), design a webpage using appropriate element, tables, lists and images for optimal readability & user experience

Layout & Structure:

Header main Content Area, Sidebar, Footer.

HTML Structure:

```
<!DOCTYPE html>
```

```
<html lang = "en">
```

```
<head>
```

```
<meta charset = "UTF-8">
```

```
<meta name = "viewport" content = "width = device-  
width, initial, scale = 1.0">
```

```
<title> responsive Blog post </title>
```

```
<link rel = "stylesheet" href = "styles.css">
```

```
</head>
```

```
<body>
```

```
<header>
```

```
<h1> My Blog </h1>
```

```
<nav>
```

```
<ul>
```

```
<li> <a href = "#"> Home </a> </li>
```

```
<li> <a href = "#"> about </a> </li>
```



</a> <a href="#"> Contact </a> </div>

</div>

</nav>

</header>

<div class="content">

<article class="post">

<thead>

<tr>

<th> feature </th>

<th> file box </th>

<th> Grid </th>

</th>

</thead>

</tbody>

<tr>

<td> 1D / 2D layout </td>

<td> 1D </td>

<td> 2D </td>

</tr>

<tr>

<td> use

</div> </div> @ media (max-width: 768px) {

content {  
flex-direction: column;

post &  
margin - right : 0;

}  
css for styling:

CSS

body {

font-family : Arial, sans-serif;

line-height : 1.6;

margin : 0;

padding : 0;

color : #333;

}

header {

background-color : #333;

color : white;

padding : 20px 0;

text-align : center;

}

nav ul {

list-style-type : none;

padding : 0;

}

nav ul li {

display : inline;

margin : 0 10px;

nav ul li a {

color: white;

text-decoration: none;

Content {

display: flex;

flex-wrap: wrap;

padding: 20px;

}

post {

flex: 3;

margin-right: 20px;

}

sidebar {

flex: 1;

}

sidebar h3 {

margin-top: 0;

}

table {

width: 100%;

border-collapse: collapse;

margin: 20px 0;

}

@media (max-width: 768px) {

.content {

flex-direction: column

.part {

margin-right: 0;

}

}

JavaScript for interactivity:

JavaScript

Example: Add interactive or track events

document.addEventListener('DOMContentLoaded',

{

console.log('page loaded');

});