CET-223

Web Technologies

Experiment # 05

Experiment Title

Creating the Responsive Designs with CSS Flexbox and Grid

Assessment of CLO(s): 04

Performed on _____

Student Name:			
Roll No.		**Group**	
Semester		**Session**	

Total (Max)	**Performance (03)**	**Viva (03)**	**File (04)**	**Total (10)**
Marks Obtained				
Remarks (if any)				

Experiment evaluated by

Instructor's Name	**Engr. Bilal Iqbal**		
Date		**Signature**	

Objective:

To understand how to create responsive web designs using **CSS Flexbox** and **CSS Grid** layouts. This lab will guide you through different exercises to create layouts that adapt to various screen sizes.

Flexbox

Flexbox (Flexible Box Layout) is a CSS layout mode that allows us to create flexible and responsive layouts by distributing space dynamically between items in a container.

Here are some key properties used in Flexbox:

- display: flex: This defines a flex container.
- flex-direction: Specifies the direction of the flex items (row or column).
- justify-content: Aligns the flex items horizontally.
- align-items: Aligns the flex items vertically.
- flex-wrap: Specifies whether the flex items should wrap if there isn't enough space.

Example

1. Create an HTML file named `index.html`.
2. Add a simple navigation bar layout:

```
<!DOCTYPE html>
<html >
<head>
  <title>Flexbox Navigation Bar</title>
  <link rel="stylesheet" href="style.css">
</head>
<body>
  <header>
    <nav class="navbar">
      <div class="logo">MySite</div>
      <ul class="nav-links">
        <li><a href="#">Home</a></li>
        <li><a href="#">About</a></li>
        <li><a href="#">Services</a></li>
        <li><a href="#">Contact</a></li>
      </ul>
    </nav>
  </header>
</body>
</html>
```

3. Add the following Flexbox-based CSS in a file named `style.css`

```
* {
   margin: 0;
   padding: 0;
   box-sizing: border-box;
}

body {
   font-family: Arial, sans-serif;
}

.navbar {
   display: flex;
   justify-content: space-between;
   align-items: center;
   padding: 10px 20px;
   background-color: #333;
}

.logo {
   color: white;
   font-size: 24px;
}

.nav-links {
   list-style: none;
   display: flex;
}

.nav-links li {
   margin-left: 20px;
}

.nav-links a {
   color: white;
   text-decoration: none;
}
```

CSS Grid

CSS Grid Layout is a two-dimensional system for arranging items in rows and columns. It allows for more complex and flexible layouts than Flexbox, which is more suited for one-dimensional layouts.

Here are some important properties used in CSS Grid:

- `display: grid`: This defines a grid container.
- `grid-template-columns`: Defines the number of columns and their widths.
- `grid-template-rows`: Defines the number of rows and their heights.
- `grid-gap`: Defines the space between grid items.
- `grid-column`: Specifies the number of columns an item should span.

```
<!DOCTYPE html>
<html lang="en">
<head>
   <link rel="stylesheet" href="style.css">
</head>
<body>
  <main class="portfolio">
   <div class="item">Project 1</div>
   <div class="item">Project 2</div>
   <div class="item">Project 3</div>
   <div class="item">Project 4</div>
   <div class="item">Project 5</div>
   <div class="item">Project 6</div>
</main>

</body>
</html>
```

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

.item {
   background-color: #f4f4f4;
   padding: 20px;
   text-align: center;
   border: 1px solid #ddd;
}

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

Tasks:

Create an e-store layout using Flexbox and CSS Grid for the product display, navigation, and responsive design.