**­­**

**­­­­­­­**

**LAB # 03**

|  |  |
| --- | --- |
| **SUBMITTED TO­** | **­­­O­H** |
| **SUBMITTED BY** |  |
| **REGISTRATION NO.** |  |

**­**

# **Task # 1:**

**1. You are required to explain the theoretical concepts of Dropdowns, Flexbox, Grid and Media Queries.**

**ANSWER:**

1. Dropdowns

A dropdown is a common user interface element used on websites to display a list of options that appear when the user clicks or hovers over a button, text, or icon. It helps organize multiple options in a compact way without taking up too much space on the page.

Dropdowns are often used in navigation menus, filter systems, or forms. For example, a website’s navigation bar might have a "Services" tab, and when clicked, it reveals a list of available services. Similarly, in a product gallery, a dropdown might allow users to filter items by category or price range.

They can be created using simple HTML <select> tags or through a combination of HTML, CSS, and JavaScript for more advanced interactive dropdown menus. The main purpose of a dropdown is to improve usability and make the interface more organized and user-friendly.

2. Flexbox

Flexbox, short for Flexible Box Layout, is a layout model in CSS that helps developers easily arrange elements within a container, either horizontally (in a row) or vertically (in a column). It is a one-dimensional layout system, meaning it works best for managing space and alignment in a single direction.

Flexbox is particularly useful for creating responsive layouts that adjust automatically to different screen sizes. It simplifies tasks such as centering elements, distributing space evenly, or reordering items without changing the HTML structure. Some important Flexbox properties include:

* display: flex; – activates Flexbox on a container.
* justify-content – controls how items are aligned horizontally.
* align-items – controls how items are aligned vertically.
* flex-direction – sets whether items are placed in a row or column.

Flexbox is often used for building navigation bars, aligning cards or sections, and creating responsive layouts where elements need to adjust dynamically.

3. CSS Grid

CSS Grid is a more powerful and flexible layout system compared to Flexbox, as it allows control over both rows and columns simultaneously. It is a two-dimensional layout model designed for building more complex web page structures like galleries, dashboards, and complete page templates.

With CSS Grid, developers can divide a page into major sections and precisely position elements within a layout. It offers more control over spacing, alignment, and placement than older layout methods like floats or tables. Key properties of Grid include:

* display: grid; – activates the grid layout.
* grid-template-columns and grid-template-rows – define the structure of the grid.
* grid-gap – controls the spacing between grid items.
* grid-column and grid-row – specify where an element should be placed.

For example, a product gallery can be easily created using a grid layout by defining a number of columns and rows. On larger screens, products can appear in a 4-column layout, while on smaller screens, the layout can adapt automatically with fewer columns.

4. Media Queries

Media queries are a core feature of CSS used to make websites responsive, meaning they adjust their layout and design based on the device’s screen size, resolution, or orientation. They allow developers to write conditional CSS rules that only apply when certain conditions are met, such as a specific screen width.

The syntax of a media query typically looks like this:

@media (max-width: 768px) {

/\* CSS rules for tablets or smaller devices \*/

}

With media queries, developers can change font sizes, hide or show elements, adjust grid layouts, or rearrange content for different screen sizes. For example, a product gallery that displays four products in a row on a desktop might automatically switch to two products per row on a tablet and one product per row on a mobile device.

Media queries are essential for building responsive websites that offer a good user experience across desktops, tablets, and smartphones.

# **LAB TASK # 02:**

­­­­ **Write question here.**

**HTML:**

|  |
| --- |
| <!DOCTYPE html>  <html lang="en">    <head>      <meta charset="UTF-8" />      <meta name="viewport" content="width=device-width, initial-scale=1.0" />      <title>Product Gallery</title>      <link rel="stylesheet" href="style.css" />    </head>    <body>      <!-- Header & Navigation -->      <header>        <div class="logo">Cartio</div>        <nav>          <a href="#">Home</a>          <a href="#">Shop</a>          <a href="#">Contact</a>          <a href="#">About</a>        </nav>        <div class="search-bar">          <input type="text" placeholder="Search products..." />        </div>      </header>      <main>        <!-- Hidden radio inputs (must be siblings of .product-grid) -->        <input type="radio" id="all" name="category" checked />        <input type="radio" id="electronics" name="category" />        <input type="radio" id="clothing" name="category" />        <input type="radio" id="books" name="category" />        <!-- Sidebar Filter (labels reference the radios above) -->        <aside class="sidebar">          <h3>Filter by Category</h3>          <label for="all" class="filter-label">All</label>          <label for="electronics" class="filter-label">Electronics</label>          <label for="clothing" class="filter-label">Clothing</label>          <label for="books" class="filter-label">Books</label>        </aside>        <!-- Product Grid -->        <section class="product-grid">          <!-- Electronics -->          <div class="product electronics">            <img src="assets/1.jpeg" alt="Smartphone" />            <h4>Smartphone</h4>            <p>$499</p>            <button>Add to Cart</button>          </div>          <div class="product electronics">            <img src="assets/2.jpeg" alt="Headphones" />            <h4>Headphones</h4>            <p>$99</p>            <button>Add to Cart</button>          </div>          <div class="product electronics">            <img src="assets/3.jpeg" alt="Laptop" />            <h4>Laptop</h4>            <p>$899</p>            <button>Add to Cart</button>          </div>          <div class="product electronics">            <img src="assets/4.jpeg" alt="Camera" />            <h4>Camera</h4>            <p>$299</p>            <button>Add to Cart</button>          </div>          <!-- Clothing -->          <div class="product clothing">            <img src="assets/5.jpeg" alt="T-Shirt" />            <h4>T-Shirt</h4>            <p>$25</p>            <button>Add to Cart</button>          </div>          <div class="product clothing">            <img src="assets/6.jpeg" alt="Jacket" />            <h4>Jacket</h4>            <p>$120</p>            <button>Add to Cart</button>          </div>          <div class="product clothing">            <img src="assets/7.jpeg" alt="Jeans" />            <h4>Jeans</h4>            <p>$60</p>            <button>Add to Cart</button>          </div>          <div class="product clothing">            <img src="assets/8.jpeg" alt="Shoes" />            <h4>Shoes</h4>            <p>$80</p>            <button>Add to Cart</button>          </div>          <!-- Books -->          <div class="product books">            <img src="assets/9.jpeg" alt="Fiction Novel" />            <h4>Fiction Novel</h4>            <p>$15</p>            <button>Add to Cart</button>          </div>          <div class="product books">            <img src="assets/10.jpeg" alt="Science Book" />            <h4>Science Book</h4>            <p>$30</p>            <button>Add to Cart</button>          </div>          <div class="product books">            <img src="assets/11.jpeg" alt="History Book" />            <h4>History Book</h4>            <p>$20</p>            <button>Add to Cart</button>          </div>          <div class="product books">            <img src="assets/12.jpeg" alt="Comics" />            <h4>Comics</h4>            <p>$10</p>            <button>Add to Cart</button>          </div>        </section>      </main>    </body>  </html> |

**CSS:**

|  |
| --- |
| /\* Basic reset \*/  \* {    box-sizing: border-box;    margin: 0;    padding: 0;    font-family: Cambria;  body {    background: hsl(210, 70%, 75%);    color: #000000;  }  /\* Header \*/  header {    display: flex;    align-items: center;    justify-content: space-between;    background: hsl(210, 72%, 64%);    color: #fffefe;    padding: 14px 24px;  }  header .logo {    font-weight: 700;    font-size: 1.4rem;  }  header nav a {    color: #fff;    text-decoration: none;    margin: 0 10px;  }  .search-bar input {    padding: 6px 10px;    border-radius: 6px;    border: none;  }  /\* Layout \*/  main {    display: flex;    gap: 20px;    padding: 20px;    align-items: flex-start;    /\* radios must be direct children of main (hidden) \*/  }  /\* hide the radios visually but keep accessible \*/  main > input[type="radio"] {    position: absolute;    left: -9999px;    width: 1px;    height: 1px;    overflow: hidden;  }  /\* Sidebar \*/  .sidebar {    width: 220px;    background: #fff;    padding: 16px;    border-radius: 8px;    box-shadow: 0 1px 5px rgba(0, 0, 0, 0.05);    height: fit-content;  }  .sidebar h3 {    margin-bottom: 12px;    font-size: 1rem;  }  .sidebar .filter-label {    display: block;    padding: 10px 8px;    cursor: pointer;    border-radius: 6px;    margin-bottom: 6px;    border: 1px solid transparent;  }  .sidebar .filter-label:hover {    background: #f3f7ff;  }  /\* show active label using the radios (general sibling selector) \*/  #all:checked ~ .sidebar .filter-label[for="all"],  #electronics:checked ~ .sidebar .filter-label[for="electronics"],  #clothing:checked ~ .sidebar .filter-label[for="clothing"],  #books:checked ~ .sidebar .filter-label[for="books"] {    background: #e9f0ff;    color: #0b5ed7;    font-weight: 700;    border-color: #d6e4ff;  }  /\* Product grid \*/  .product-grid {    flex: 1;    display: grid;    grid-template-columns: repeat(4, 1fr);    gap: 18px;  }  /\* Card \*/  .product {    background: #fff;    border-radius: 8px;    padding: 12px;    text-align: center;    box-shadow: 0 1px 4px rgba(0, 0, 0, 0.04);    transition: transform 0.18s ease, box-shadow 0.18s ease;  }  .product img {    width: 100%;    height: 150px;    object-fit: cover;    border-radius: 6px;    margin-bottom: 10px;  }  .product h4 {    margin-bottom: 6px;    font-size: 1rem;  }  .product p {    font-weight: 700;    margin-bottom: 10px;  }  .product button {    padding: 8px 12px;    border: none;    border-radius: 6px;    background: #0b5ed7;    color: #fff;    cursor: pointer;  }  .product:hover {    transform: translateY(-6px);    box-shadow: 0 6px 18px rgba(10, 10, 10, 0.06);  }  /\* Advanced selector example: style every 4th product slightly differently \*/  .product:nth-child(4n) {    border-top: 3px solid #f1f3f5;  }  /\* CSS-only filtering rules:     When a radio is checked, hide products that do not match that category.     We use the general sibling selector (~) so the radio inputs (which are     direct children of <main>) can affect .product-grid below them.  \*/  /\* When electronics selected -> hide non-electronics \*/  #electronics:checked ~ .product-grid .product:not(.electronics) {    display: none;  }  /\* When clothing selected -> hide non-clothing \*/  #clothing:checked ~ .product-grid .product:not(.clothing) {    display: none;  }  /\* When books selected -> hide non-books \*/  #books:checked ~ .product-grid .product:not(.books) {    display: none;  }  /\* All selected -> show all \*/  #all:checked ~ .product-grid .product {    display: block;  }  /\* Responsive behavior \*/  @media (max-width: 1024px) {    .product-grid {      grid-template-columns: repeat(2, 1fr);    }  }  @media (max-width: 768px) {    main {      flex-direction: column;      padding: 14px;    }    .sidebar {      width: 100%;      order: 0;      margin-bottom: 14px;    }    .product-grid {      grid-template-columns: repeat(1, 1fr);    }  } |

**SCREEN SHOT:**

**OUTPUT:**