**Module 3 - CSS**

**Ans (1) :-**

**What is a CSS selector:-**

A **CSS selector** is a pattern used to choose and style specific HTML elements on a web page.

**1. Element Selector:-**The element selector targets all HTML elements of a specific type.

### 2. ****Class Selector:-****The class selector is used to style elements with a specific class attribute. it start with a dot (.) in CSS.

### 3. ****ID Selector:-**** The ID selector targets a single element with a specific id attribute. it start with (#) in CSS.

### Example:-

### OUTPUT:-

### 

### Ans(2):-

### CSS specificity:-

### CSS specificity is a set of rules used by browsers to determine which CSS rule to apply when multiple rules target the same element.

### 🡪 More specific selectors taking precedence over less specific ones.

### Resolving Conflicts:-

🡪CSS rule has a specificity score based on its selectors. The rule with the highest specificity takes precedence.

🡪 Some properties, like color and font-family, are **inherited** by child elements by default, while others, like margin and border, are not.

### Ans(3):-

|  |  |  |
| --- | --- | --- |
| Inline | Internal | External |
| Placed directly in the HTML element. | Placed in the head section of the HTML. | Liked to the html page. |
| No selector used. | Use the <style>tag | Use the <link>tag in the <head>tag. |
| Only apply to the HTML element. | Only apply to the current HTML page. | All the style needed for all page. |
| Overrides both internal and external styles | Overrides external CSS | separation of content in HTML. |

### Advantages and Disadvantages :-

### 1.Inline CSS:

**Quick and Simple:** Easy to implement for small changes or specific elements.

### ****No External Files Required:**** Styles are embedded directly, reducing dependency on additional files.

### Disadvantages:-

**Redundant Code:** Leads to code duplication, as styles are repeated for each element.

**Not Scalable:** Unsuitable for larger projects due to cluttered HTML and lack of separation between content and presentation.

### ****Low Reusability:**** Styles cannot be reused for multiple elements.

### 2.Internal CSS:

**Easy to Apply to a Single Page:** Useful for styling a single page without affecting others.

**No External File Required:** All styles are included in the HTML file, reducing dependency on external resources.

**Higher Specificity than External Styles:** Useful when overriding external CSS.

### Disadvantages:-

**Limited Reusability:** Styles are confined to one document, requiring duplication for other pages.

### 2.External CSS:

**Reusable Styles:** A single style sheet can be used across multiple pages, improving consistency.

**Scalability:** Well-suited for large projects and teams due to centralized styling.

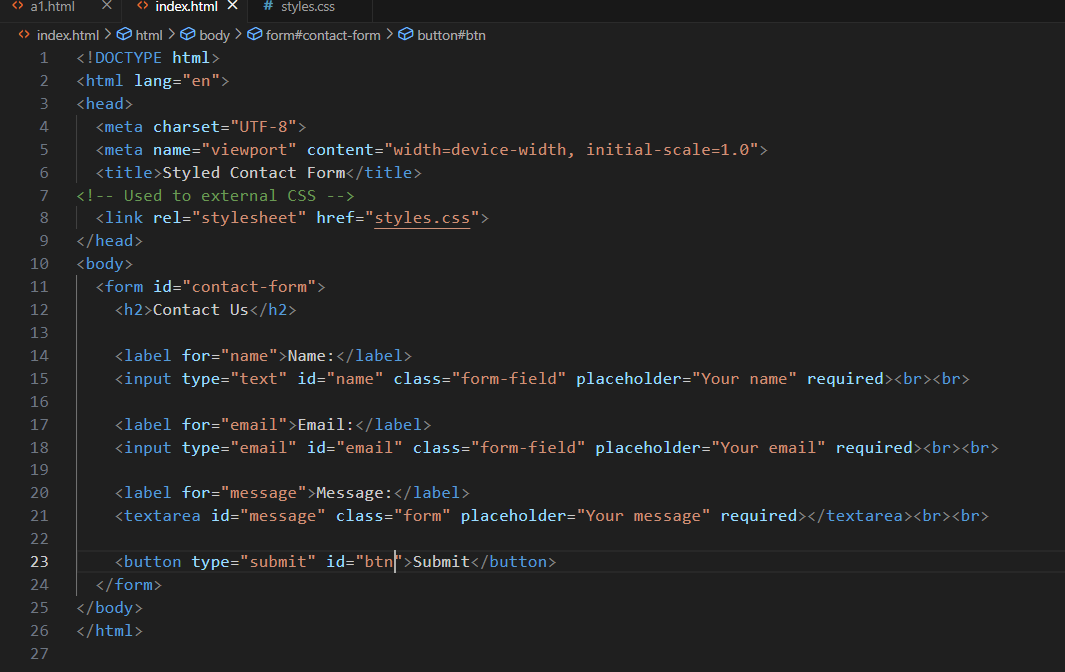
**Faster Maintenance:** Changes in one CSS file automatically reflect across all linked pages.

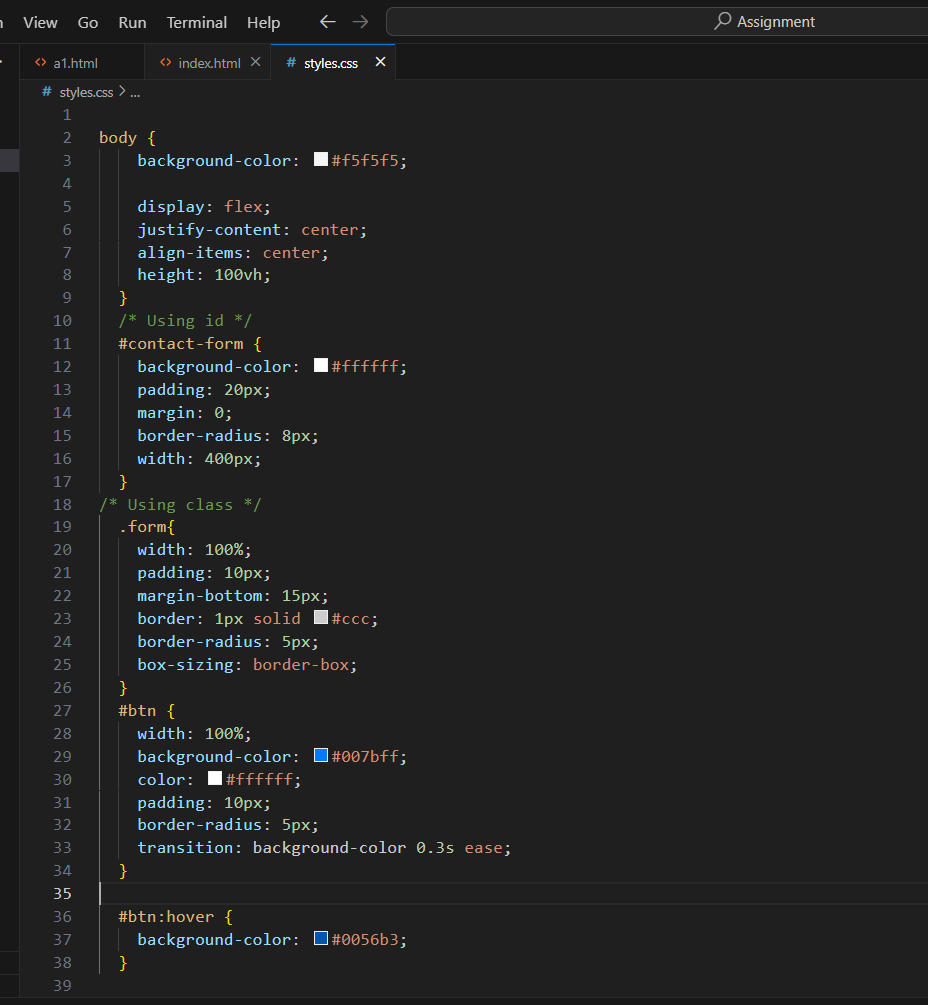
### Disadvantages:-

**Initial Load Time:** Requires an additional HTTP request to fetch the CSS file, which may impact page load time.

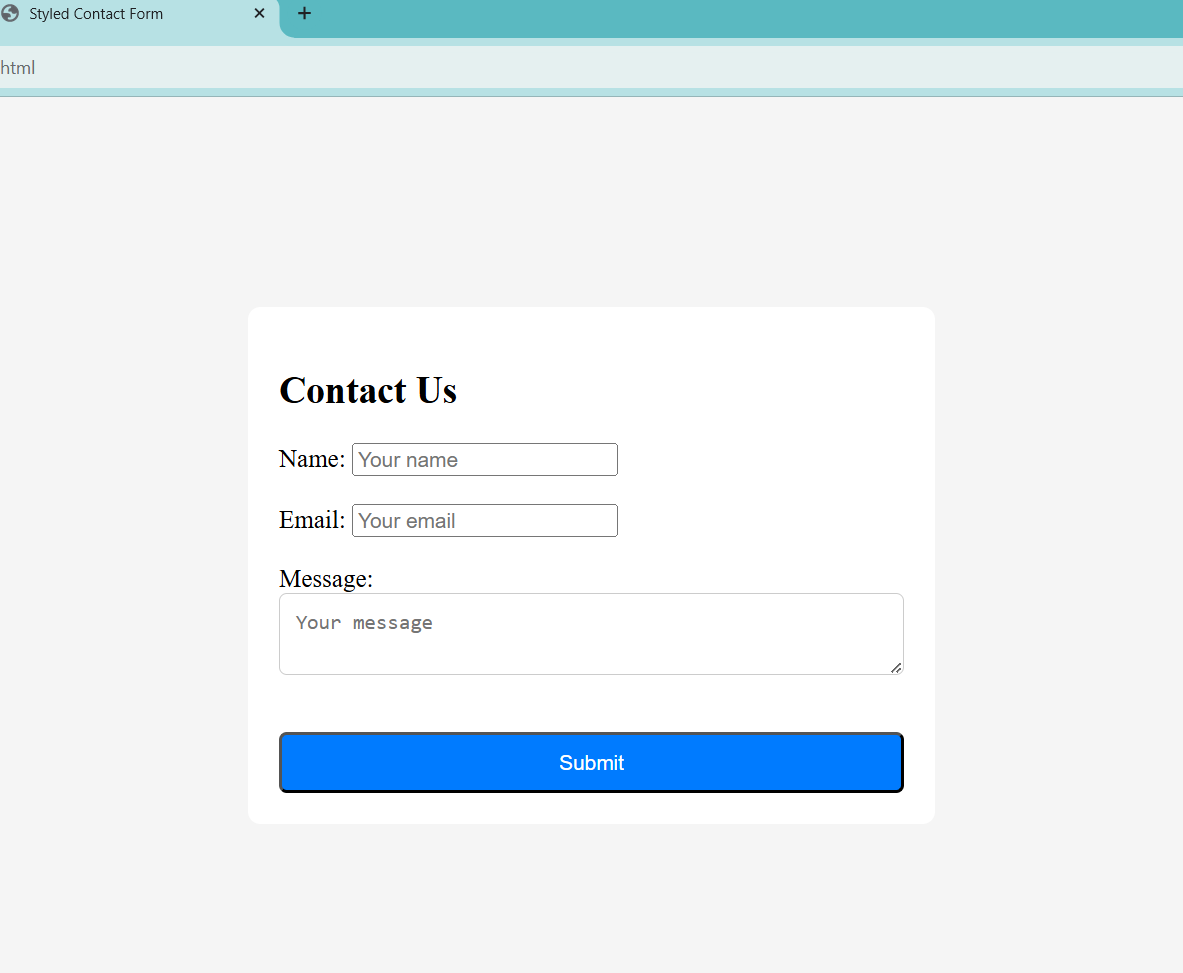
**Complex Debugging:** When multiple style sheets are used, debugging specificity conflicts can become challenging.

|  |
| --- |
| **Lab Assignment** |

****

****

**OUTPUT:-**

****

**CSS Box Model:-**

**Ans (1):-**

The **CSS Box Model** is a fundamental concept in web design that describes how every element on a webpage is structured and sized.

### Content:-The innermost part of the box where text, images, or other content is displayed.

### 🡪Size :- The size of the content depends on the width and height properties you set for the element.

1. **Padding:-**The space between the content and the element's border.

🡪 **Size** :**-** because it pushes the border outward. For instance, if your content is 100px wide and you add padding: 10px, the total width becomes **120px.**

### 3.Borders: - The edge surrounding the padding and content. Think of it as a frame around the element.

🡪**Size:-** If your content is 100px wide, with padding: 10px and border: 5px, the total width is now **130px.**

### **4.**Margin:-The space outside the border that separates the element from other elements.

🡪Size:- If you set margin: 20px, the element will have 20px of space between its border and nearby elements.

**Ans (2):-**

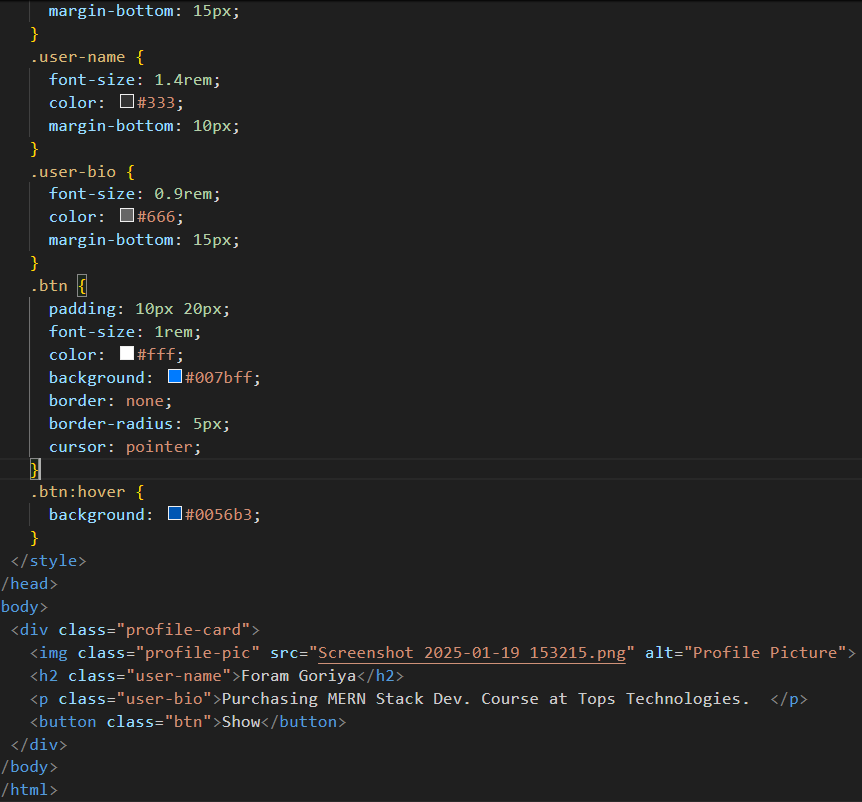
**Difference between border-boxand content-boxbox-sizing:-**

The box-sizing **property** in CSS determines how the total size of an element is calculated. It controls whether padding and borders are included in the element's specified width and height.

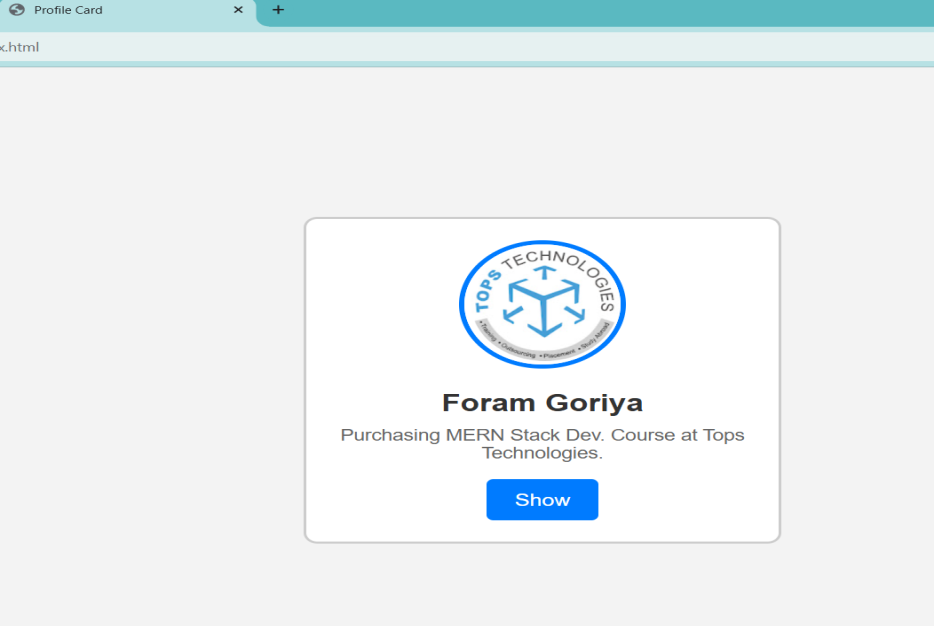
🡪Content Box are defult.

|  |
| --- |
| **Lab Assignment** |





**OUTPUT:-**



**CSS Flex box:-**

**Ans (1):-**

 Flex-box is a layout method for arranging items in rows or columns.

1. **Flex-container:**- The flex container is the parent element that holds the flex items. The display: flex; or display: inline-flex; property is applied to the container to enable Flex-box. A flex container organizes its child elements (flex items) along either a row or a column, based on the defined direction.
2. **Flex-item:**- Flex items are the direct children of the flex container. These items are arranged according to the Flex-box model, which allows them to grow, shrink, or remain at their initial size depending on available space and flex properties.

**How it is useful:-**

**🡪Alignment and Distribution:** Flex-box makes it easier to align items vertically and horizontally within a container, both in terms of space distribution and alignment along the main and cross axes.

**🡪Responsiveness:** Flex-box helps design responsive layouts by making items adapt to different screen sizes and available space.

**🡪Flexibility:** Flex-box provides a flexible way to arrange elements within a container without needing to explicitly define widths or heights for each item.

**Ans(2):-**

**Justify-content,align-items,flex-directionproperties:-**

**Justify-content:-**

The justify-content property controls the alignment of flex items along the main axis.It distributes space between the items and aligns them relative to the container.

Values:-

🡪flex-start: Items are aligned to the start of the container.

🡪flex-end: Items are aligned to the end of the container.

🡪center: Items are centered within the container.

🡪space-between: Items are spaced out evenly with the first item at the start and the last item at the end.

🡪space-around: half the space between the items is at the ends of the container.

🡪space-evenly: equal space between them and at the edges of the container.

**Example:-**

.container {

display: flex;

justify-content: center;

}

**align-items:-**The align-items propertyallows you to align items within the flex container based on their height.

**Values:**

🡪stretch: Items stretch to fill the container.

🡪flex-start: Items are top if the main axis is horizontal.

🡪flex-end: Items are aligned bottom if the main axis is horizontal.

🡪center: Items are aligned in the center of the cross axis.

🡪baseline: Items are aligned the text baseline of the items.

**Example:**

.container {

display: flex;

align-items: center;

}

**3.flex-direction:-**The flex-direction property defines the direction of the main axiswhich the flex items are placed within the container.

**Values:**

🡪row (default): Items arehorizontally from left to right.

🡪row-reverse: Items are horizontally but right to left.

🡪column: Items are vertically from top to bottom.

🡪column-reverse: Items are vertically bottom to top.

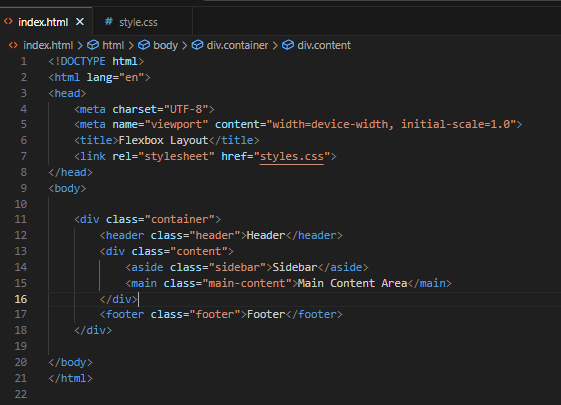
**Example:**

.container {

display: flex;

flex-direction: column;

}

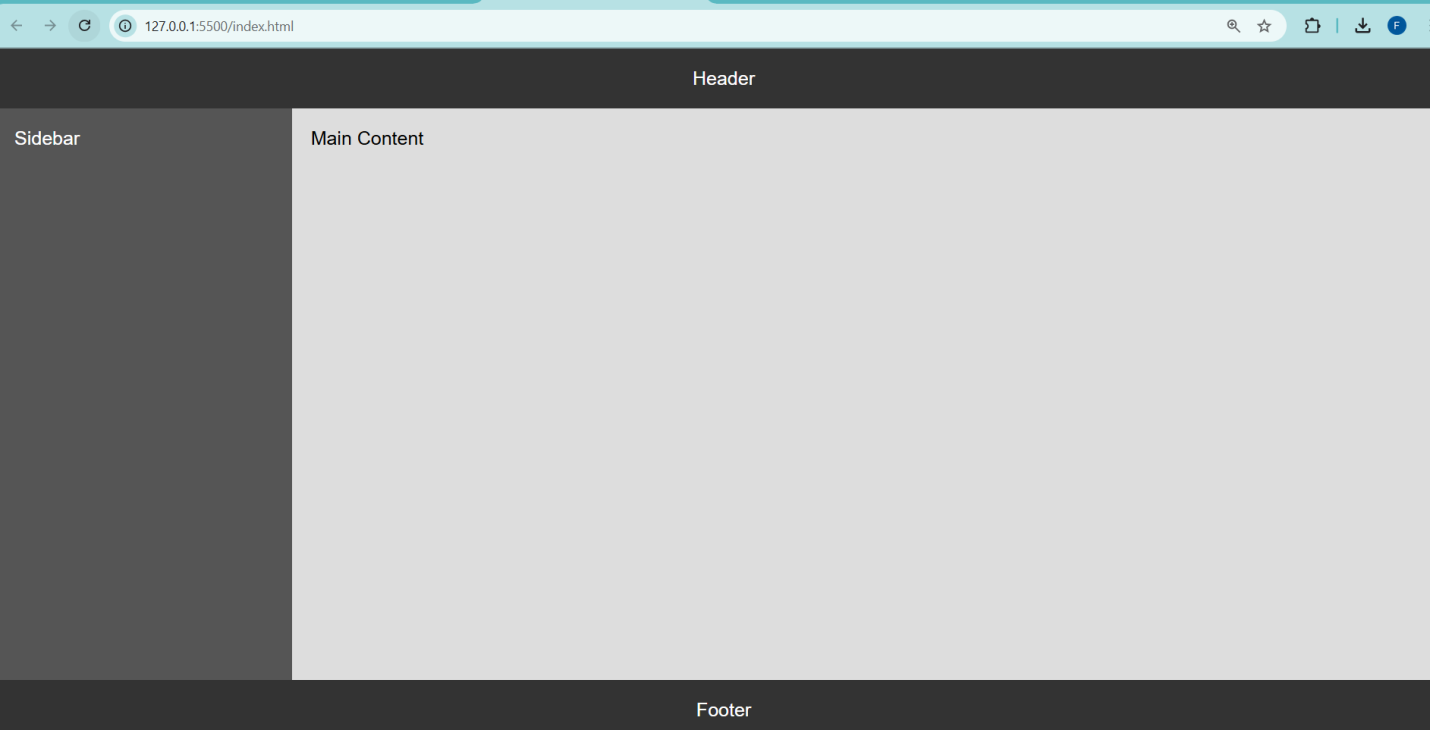


|  |
| --- |
| **Lab Assignment** |

****

****

**OUTPUT:-**

****

**CSS Grid**

**Ans (1):-**

**CSS Grid and how it differs from Flex box:-**

**CSS Grid** and **Flex box** are both layout systems in CSS that help design responsive and dynamic web layouts.

#### ****Key Features of CSS Grid:-****

**🡪Rows & Columns:** Allows precise placement of items in both directions.

**🡪Explicit and Implicit Grids:** grid-template-rows & grid-template-columns or let Grid automatically create one.

🡪**Grid Areas:** You can assign multiple elements to specific named areas.

**🡪Flexible Sizing:** Supports auto-sizing, and min-max().

**🡪Alignment Controls:** Supports align-items, justify-items, place-items, and gap

**When to Use Grid vs. Flex box:-**

🡪Layout type in grid 2-dimensional(row columns).

|  |
| --- |
| 🡪Layout type in grid 2-dimensional (row columns). |
|  |

🡪 Complex layouts in grid system.

🡪 Simple layouts in Flex box.

**Ans (2):-**

**Grid Example:-**

## 1. grid-template-columns:- This property defines the number and size of the columns in a CSS grid.

**Example:-**

.container {

display: grid;

grid-template-columns: 100px 200px auto;

}

## 2. grid-template-rows:- This property defines the number and size of the rows in a CSS grid.

**Example:-**

.container {

display: grid;

grid-template-rows: 100px 200px auto;

}

## grid-gap ****:-****The grid-gap property defines spacing between rows and columns.

**Example:-**

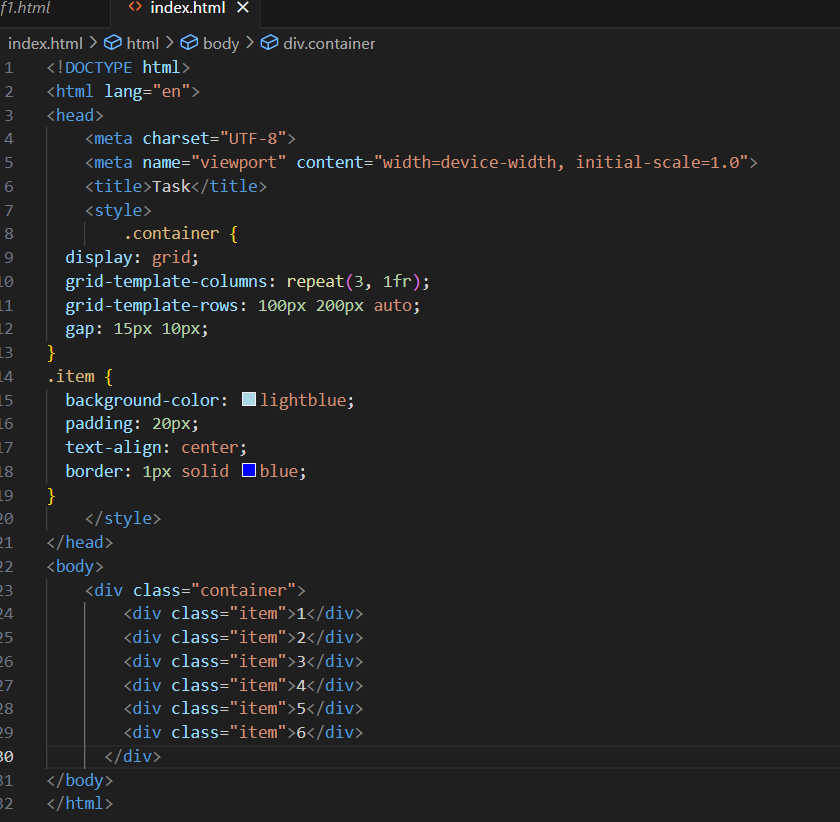
.container {

display: grid;

grid-template-columns: repeat(3, 1fr);

grid-template-rows: repeat(2, 150px);

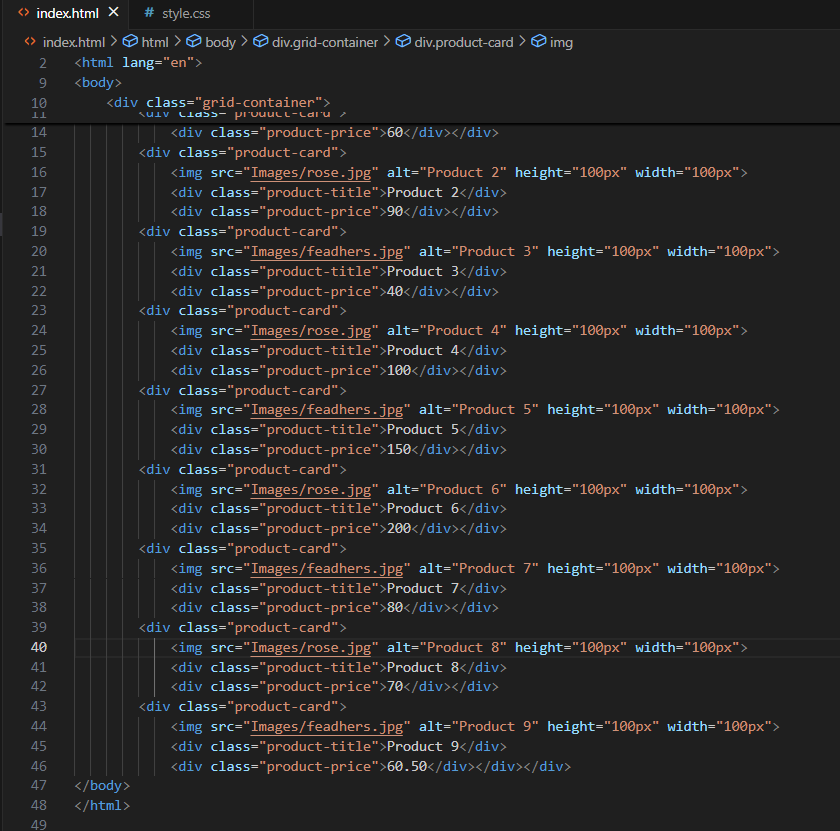
gap: 20px; }

****

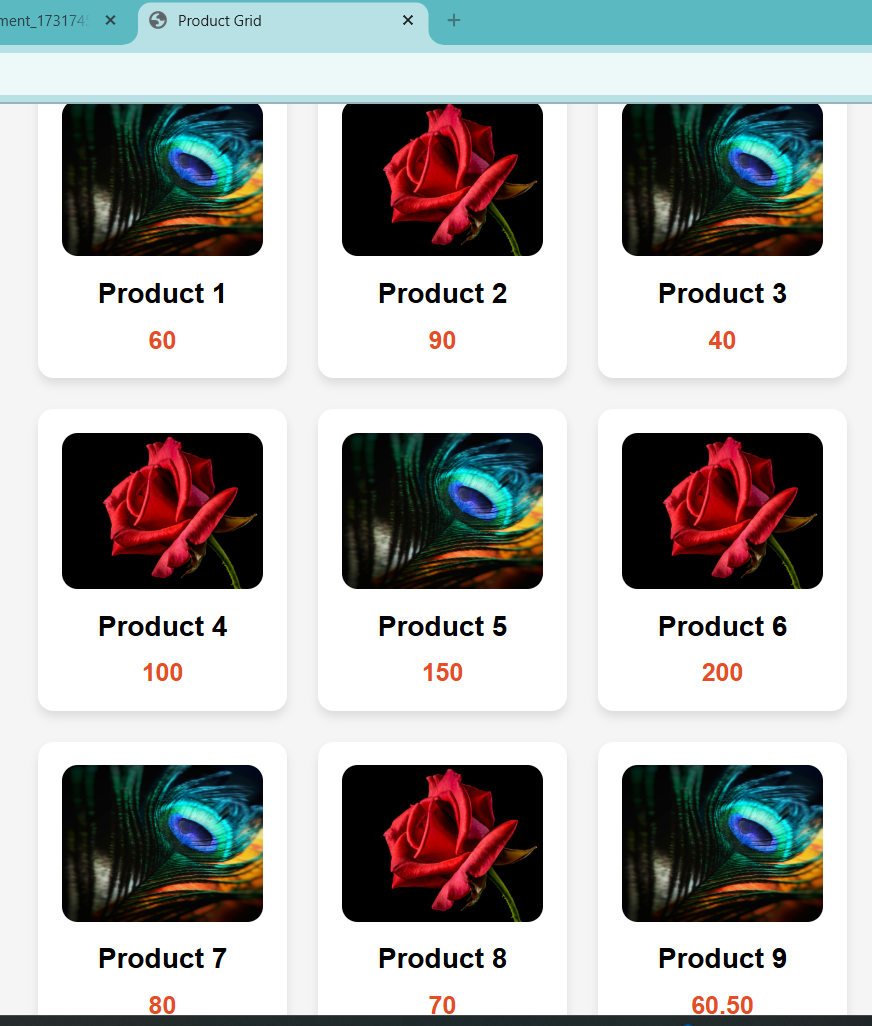
**Output:-**



|  |
| --- |
| **Lab Assignment** |



**OUTPUT:-**

****