

## Day 13: CSS Grid

Date: 20-06-24

---

**Summary of the Day:** On the thirteenth day of our web development training, we delved into CSS Grid. CSS Grid is a powerful two-dimensional layout system that allows developers to create complex and responsive grid-based layouts with ease. The session focused on understanding the basics of CSS Grid, including container properties and grid item properties.

### Detailed Notes:

**1. Introduction to CSS Grid:** CSS Grid is designed to handle both rows and columns, providing a more robust layout system compared to Flexbox, which is primarily one-dimensional.

**2. CSS Grid Container Properties:** The following properties are used to define a grid container and specify its behavior.

- **display: grid;** Turns an element into a grid container.

```
? .grid-container {  
  display: grid;  
}
```

? **grid-template-columns and grid-template-rows:** Define the number and size of the columns and rows in the grid.

```
? .grid-container {  
  grid-template-columns: 1fr 1fr 1fr; /* Three equal columns */  
  grid-template-rows: 100px 200px; /* Two rows with specific heights */  
}
```

? **grid-template-areas:** Names grid areas for easier reference and layout.

```
? .grid-container {  
  grid-template-areas:  
    'header header header'  
    'sidebar content content'  
    'footer footer footer';  
}
```

? **grid-gap (or gap):** Sets the spacing between rows and columns.

```
? .grid-container {  
  grid-gap: 10px; /* Space between rows and columns */  
}
```

❓ **grid-auto-flow:** Controls how auto-placed items are inserted into the grid.

- `.grid-container {`
- `grid-auto-flow: row; /* Default */`
- `}`
- `/* Other values: column, dense, row dense, column dense */`
- 

**3. CSS Grid Item Properties:** The following properties are used to define the placement and behavior of grid items within a grid container.

- **grid-column-start, grid-column-end, grid-row-start, grid-row-end:** Specify the start and end positions of a grid item.

❓ `.grid-item {`

`grid-column-start: 1;`

`grid-column-end: 3; /* Span across the first two columns */`

`grid-row-start: 1;`

`grid-row-end: 2; /* Span across the first row */`

`}`

❓ **grid-column and grid-row:** Shorthand for setting both the start and end positions.

❓ `.grid-item {`

`grid-column: 1 / 3; /* Span across the first two columns */`

`grid-row: 1 / 2; /* Span across the first row */`

`}`

❓ **grid-area:** Assigns an item to a named grid area.

❓ `.grid-item {`

`grid-area: header;`

`}`

❓ **justify-self:** Aligns the grid item within its column.

❓ `.grid-item {`

`justify-self: center; /* Default is stretch */`

`}`

`/* Other values: start, end, stretch */`

❓ **align-self:** Aligns the grid item within its row.

- `.grid-item {`

- align-self: center; /\* Default is stretch \*/
- }
- /\* Other values: start, end, stretch \*/
- 

#### 4. Practical Examples:

##### Basic Grid Layout:

```
<!DOCTYPE html>

<html lang="en">

<head>

  <meta charset="UTF-8">

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

  <title>CSS Grid Example</title>

  <style>

    .grid-container{

      display: grid;

      grid-template-columns: 1fr 1fr 1fr;

      grid-template-rows: 100px 100px;

      grid-gap: 10px;

      background-color: lightgray;

    }

    .grid-item {

      background-color: lightcoral;

      text-align: center;

      padding: 20px;

    }

  </style>

</head>

<body>

  <div class="grid-container">

    <div class="grid-item">Item 1</div>

    <div class="grid-item">Item 2</div>
```

```
<div class="grid-item">Item 3</div>
<div class="grid-item">Item 4</div>
<div class="grid-item">Item 5</div>
<div class="grid-item">Item 6</div>
</div>
</body>
</html>
```

### **Grid Layout with Named Areas:**

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>CSS Grid Named Areas</title>
  <style>
    .grid-container {
      display: grid;
      grid-template-areas:
        'header header header'
        'sidebar content content'
        'footer footer footer';
      grid-gap: 10px;
      background-color: lightgray;
    }
    .header {
      grid-area: header;
      background-color: lightblue;
      text-align: center;
      padding: 20px;
    }
    .sidebar {
```

```
    grid-area: sidebar;

    background-color: lightgreen;

    text-align: center;

    padding: 20px;
}

.content {

    grid-area: content;

    background-color: lightcoral;

    text-align: center;

    padding: 20px;
}

.footer {

    grid-area: footer;

    background-color: lightgoldenrodyellow;

    text-align: center;

    padding: 20px;
}

</style>
</head>
<body>

<div class="grid-container">

    <div class="header">Header</div>

    <div class="sidebar">Sidebar</div>

    <div class="content">Content</div>

    <div class="footer">Footer</div>

</div>

</body>
</html>
```

## Day 14: Applying External CSS

Date: 21-06-24

---

## 1. Setting Up External CSS:

- Create a separate CSS file (e.g., styles.css).
- Link the external CSS file to the HTML document using the <link> tag within the <head> section.
- <link rel="stylesheet" href="styles.css">
- 

## 2. Structuring the HTML:

- Ensure the HTML document is well-structured with semantic tags.
- Example of a basic HTML structure:
- <!DOCTYPE html>
- <html lang="en">
- <head>
- <meta charset="UTF-8">
- <meta name="viewport" content="width=device-width, initial-scale=1.0">
- <title>My Webpage</title>
- <link rel="stylesheet" href="styles.css">
- </head>
- <body>
- <header class="header">Header Content</header>
- <nav class="navbar">Navigation Bar</nav>
- <main class="main-content">
- <section class="content">Main Content</section>
- <aside class="sidebar">Sidebar Content</aside>
- </main>
- <footer class="footer">Footer Content</footer>
- </body>
- </html>
- 

## 3. Writing External CSS:

### Basic Styling:

/\* styles.css \*/

```
body {  
  font-family: Arial, sans-serif;  
  margin: 0;  
  padding: 0;  
  background-color: #f0f0f0;  
}
```

```
.header, .navbar, .footer {  
  background-color: #333;  
  color: white;  
  text-align: center;  
  padding: 10px 0;  
}
```

```
.main-content {  
  display: flex;  
  justify-content: space-between;  
  margin: 20px;  
}
```

```
.content, .sidebar {  
  background-color: #fff;  
  padding: 20px;  
  margin: 10px;  
  box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);  
  flex: 1;  
}
```

### **Advanced Styling with Flexbox and Grid:**

/\* Flexbox for Navigation Bar \*/

```
.navbar {
```

```
display: flex;

justify-content: space-around;

align-items: center;

}
```

```
/* Grid for Main Content Layout */
```

```
.main-content {

display: grid;

grid-template-columns: 2fr 1fr;

gap: 20px;

}
```

```
.content {

grid-column: 1 / 2;

}
```

```
.sidebar {

grid-column: 2 / 3;

}
```

### **Adding Visual Enhancements:**

```
/* Adding Hover Effects */
```

```
.header, .navbar, .footer {

transition: background-color 0.3s;

}
```

```
.header:hover, .navbar:hover, .footer:hover {

background-color: #555;

}
```

```
/* Styling Links */
```

```
a {
```



```
color: #333;

text-decoration: none;

transition: color 0.3s;
}

a:hover {
    color: #007BFF;
}

/* Styling Buttons */
button {
    background-color: #007BFF;
    color: white;
    border: none;
    padding: 10px 20px;
    cursor: pointer;
    transition: background-color 0.3s;
}

button:hover {
    background-color: #0056b3;
}
```

## **Day 15: Introduction to JavaScript**

**Date**23-06-24

---

### **Detailed Notes:**

#### **1. What is JavaScript?**

- JavaScript is a scripting language used to create and control dynamic website content.
- It is an essential technology alongside HTML and CSS.
- JavaScript can be used for client-side (in the browser) and server-side (with Node.js) development.

## 2. Including JavaScript in HTML:

- **Inline JavaScript:**

❓ `<button onclick="alert('Hello, World!')">Click Me</button>`

❓ **Internal JavaScript:**

❓ `<!DOCTYPE html>`

`<html lang="en">`

`<head>`

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

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

`<title>Internal JavaScript Example</title>`

`<script>`

```
function showAlert() {  
    alert('Hello, World!');  
}
```

`</script>`

`</head>`

`<body>`

`<button onclick="showAlert()">Click Me</button>`

`</body>`

`</html>`

❓ **External JavaScript:**

- Create a separate JavaScript file (e.g., script.js).
- Link the external JavaScript file in the HTML document.

`<!DOCTYPE html>`

`<html lang="en">`

`<head>`

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

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

`<title>External JavaScript Example</title>`

`<script src="script.js" defer></script>`

`</head>`

```
<body>

  <button onclick="showAlert()">Click Me</button>

</body>

</html>
```

- script.js:
- - function showAlert() {
  - alert('Hello, World!');
  - }
  -

### 3. Basic JavaScript Syntax:

- **Variables:**

```
? var x = 5;    // ES5: function-scoped

let y = 10;    // ES6: block-scoped

const z = 15;  // ES6: block-scoped, constant
```

#### ? Data Types:

- Strings, Numbers, Booleans, Arrays, Objects, etc.

```
? let name = "John";

let age = 25;

let isStudent = true;

let hobbies = ["reading", "sports"];

let person = {
  firstName: "Jane",
  lastName: "Doe"
};
```

#### ? Operators:

- Arithmetic: +, -, \*, /, %
- Assignment: =, +=, -=, \*=, /=
- Comparison: ==, ===, !=, !==, <, >, <=, >=
- Logical: &&, ||, !

#### ? Functions:

```
? function greet(name) {  
    return "Hello, " + name;  
}
```

```
let greeting = greet("Alice");  
console.log(greeting); // Output: Hello, Alice
```

## ? **Control Structures:**

- **Conditional Statements:**

```
? if (age >= 18) {  
    console.log("Adult");  
} else {  
    console.log("Minor");  
}
```

## ? **Loops:**

- - for (let i = 0; i < 5; i++) {
  - console.log(i);
  - }
  - 
  - let j = 0;
  - while (j < 5) {
  - console.log(j);
  - j++;
  - }
  -

## **4. Practical Examples:**

### **Example: Changing Text Content:**

```
<!DOCTYPE html>  
  
<html lang="en">  
  
<head>  
  
    <meta charset="UTF-8">
```

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

<title>Change Text Example</title>

<script>

  function changeText() {

    document.getElementById('text').innerHTML = "Text has been changed!";

  }

</script>

</head>

<body>

  <p id="text">This is the original text.</p>

  <button onclick="changeText()">Change Text</button>

</body>

</html>
```

### **Example: Simple Calculator:**

```
<!DOCTYPE html>

<html lang="en">

<head>

  <meta charset="UTF-8">

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

  <title>Simple Calculator</title>

  <script>

    function calculate() {

      let num1 = parseFloat(document.getElementById('num1').value);

      let num2 = parseFloat(document.getElementById('num2').value);

      let result = num1 + num2;

      document.getElementById('result').innerHTML = "Result: " + result;

    }

  </script>

</head>

<body>

  <input type="number" id="num1" placeholder="Enter first number">
```

```
<input type="number" id="num2" placeholder="Enter second number">
<button onclick="calculate()">Calculate</button>
<p id="result"></p>
</body>
</html>
```

## Day 16: JavaScript Functions and Events

Date:25-06-24

---

### 1. JavaScript Functions:

- **Defining Functions:** Functions are blocks of code designed to perform specific tasks and can be reused throughout the code.

🔗 // Function declaration

```
function greet(name) {
  return "Hello, " + name;
}
```

// Function expression

```
const greet = function(name) {
  return "Hello, " + name;
};
```

// Arrow function (ES6)

```
const greet = (name) => "Hello, " + name;
```

### 🔗 Calling Functions:

🔗 console.log(greet("Alice")); // Output: Hello, Alice

🔗 **Parameters and Arguments:** Functions can accept parameters and use them within the function body.

```
🔗 function add(a, b) {
  return a + b;
}
```

```
console.log(add(5, 3)); // Output: 8
```

❓ **Default Parameters:** Parameters can have default values if no arguments are provided.

```
❓ function greet(name = "Guest") {  
    return "Hello, " + name;  
}
```

```
console.log(greet()); // Output: Hello, Guest
```

❓ **Returning Values:** Functions can return values using the return statement.

- function multiply(a, b) {
- return a \* b;
- }
- 
- let result = multiply(4, 5);
- console.log(result); // Output: 20
- 

## 2. JavaScript Events:

- **Understanding Events:** Events are actions or occurrences that happen in the browser, such as clicking a button, hovering over an element, or loading a page. JavaScript can be used to listen for and respond to these events.
- **Common Events:**
  - onclick - Triggered when an element is clicked.
  - onmouseover - Triggered when the mouse pointer is over an element.
  - onmouseout - Triggered when the mouse pointer leaves an element.
  - onkeydown - Triggered when a key is pressed down.
  - onkeyup - Triggered when a key is released.
  - onload - Triggered when the page has finished loading.
- **Event Listeners:** Event listeners can be added to HTML elements to handle events.
- // Inline event
- <button onclick="showAlert()">Click Me</button>
- 
- // Event listener
- document.getElementById("myButton").addEventListener("click", showAlert);
-

### 3. Practical Examples:

#### Example: Button Click Event:

```
<!DOCTYPE html>

<html lang="en">

<head>

  <meta charset="UTF-8">

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

  <title>Button Click Event</title>

  <script>

    function showAlert() {

      alert('Button was clicked!');

    }

  </script>

</head>

<body>

  <button onclick="showAlert()">Click Me</button>

</body>

</html>
```

#### Example: Mouseover and Mouseout Events:

```
<!DOCTYPE html>

<html lang="en">

<head>

  <meta charset="UTF-8">

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

  <title>Mouseover and Mouseout Events</title>

  <script>

    function changeColorOnMouseOver(element) {

      element.style.backgroundColor = 'yellow';

    }

    function changeColorOnMouseOut(element) {
```



```
        element.style.backgroundColor = "";
    }
</script>
</head>
<body>
    <p onmouseover="changeColorOnMouseOver(this)"
onmouseout="changeColorOnMouseOut(this)">

        Hover over this text to change its background color.

    </p>
</body>
</html>
```

### **Example: Form Validation:**

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Form Validation</title>
    <script>
        function validateForm() {
            let name = document.getElementById('name').value;
            if (name == "") {
                alert('Name must be filled out');
                return false;
            }
            return true;
        }
    </script>
</head>
<body>
    <form onsubmit="return validateForm()">
```

```
<label for="name">Name:</label>

<input type="text" id="name" name="name">

<button type="submit">Submit</button>

</form>

</body>

</html>
```

## Day 17:CREATE PARRALLEX WEBSITE BY USING CSS AND HTML

**DATE:26-06-24**

Today was an interesting day! Mam asked us to create tasks specifically related to **Flexbox** and **CSS Grid**, which are two important layout techniques in web design. I was a bit excited and nervous at the same time because these concepts are very powerful but can be tricky to get right.

### Tasks Given:

#### 1. Flexbox Tasks:

- We had to create a basic layout using **Flexbox**, arranging items in rows and columns.
- The main focus was to play with properties like justify-content, align-items, and flex-wrap. I created a simple layout where I had some boxes inside a flex container, adjusting their position both horizontally and vertically.

#### 2. CSS Grid Tasks:

- The second part of the task was focused on **CSS Grid**, where we had to divide a container into rows and columns.
- The task was to build a responsive grid layout that adjusts itself when the screen size changes. We worked with properties like grid-template-rows, grid-template-columns, and gap. I also learned how to use grid areas for placing items precisely in a grid structure.

### Challenges:

- **Flexbox:** It was a bit tricky at first to understand how to align items both horizontally and vertically inside a container. I had to do a lot of testing with align-items and justify-content to get the right positioning.
- **CSS Grid:** Grid felt a little easier once I understood how to define the rows and columns, but getting the grid items to fit properly without overlapping was a challenge. I had to use media queries to make the layout responsive, which was fun but took time.

### Code for parrallex website:

```
<!DOCTYPE html>

<html lang="en">
```

```
<head>

<meta charset="UTF-8">

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

<title>Document</title><style>

.container{

    height:100vh;

    width:100%;

    background-image: url(https://wallpaperset.com/w/full/b/b/a/461648.jpg);

    background-repeat:no repeat;

    background-position:center;

    background-size:cover;

    background-attachment:fixed;

    background-blend-mode:darken;

    position:relative;


}

.center{

    top:50%;

    left:90px;

    color:white;

    text-align:center;

    font-size:80px;

    font-weight:bold;

    font-family:cursive;

    position:absolute;

    text-decoration-line:underline;

    text-decoration-style:solid;
```

```
}  
  
.box{  
    text-align:justify;  
    font-size:18px;  
    line-height:1.5;  
    color:black;  
    background-color:white;  
    margin:0 20px;  
  
}.box5{  
    height:100vh;  
    width:100%;  
    background-image:  
url(https://image.tmdb.org/t/p/original/z5VKhNSQKQyxm0co68HAkCqHnmX.jpg);  
    background-repeat:no repeat;  
    background-position:center;  
    background-size:cover;  
    background-attachment:fixed;  
    background-blend-mode:darken;}  
  
.box2{  
    text-align:justify;  
    font-size:18px;  
    line-height:1.5;  
    color:black;  
    background-color:white;  
    margin:0 20px;  
  
}  
  
.box6{  
    height:100vh;  
    width:100%;
```

```
background-image:
url(https://image.tmdb.org/t/p/original/p5HxqSZr6WyXt7ARtK3PPuJehKC.jpg);
```

```
background-repeat:no repeat;
```

```
background-position:center;
```

```
background-size:cover;
```

```
background-attachment:fixed;
```

```
background-blend-mode:darken;}
```

```
.box3{
```

```
text-align:justify;
```

```
font-size:18px;
```

```
line-height:1.5;
```

```
color:black;
```

```
background-color:white;
```

```
margin:0 20px;
```

```
}.box7{
```

```
height:100vh;
```

```
width:100%;
```

```
background-image: url(https://cdn.wrytin.com/images/wrytup/r/1024/the-exorcist-
jxe4fmz4.jpeg);
```

```
background-repeat:no repeat;
```

```
background-position:center;
```

```
background-size:cover;
```

```
background-attachment:fixed;
```

```
background-blend-mode:darken;}
```

```
.box4{
```

```
text-align:justify;
```

```
font-size:18px;
```

```
line-height:1.5;
```

```
color:black;
```

```
background-color:white;
```

```
margin:0 20px;
```

```
}

.box8{

    height:100vh;

    width:100%;

    background-image: url(
https://www.americamagazine.org/sites/default/files/main\_image/2021/03/10/Screen%20Shot%202021-03-10%20at%2011.49.34%20AM.png.png);

    background-repeat:no repeat;

    background-position:center;

    background-size:cover;

    background-attachment:fixed;

    background-blend-mode:darken;
position:relative;}

.center1{

    top:50%;

    width:100%;

    color:azure;

    text-align:center;

    font-size:80px;

    font-weight:bold;

    font-family:cursive;

    position:absolute;

    text-decoration-style:solid;

    text-decoration-line:underline;

}

h1{

    text-align:center;

}

</style>
```

</head>

<body>

<div class="container"><div class="center">HORROR MOVIES</div></div>

<div class="box"><h1>1.THE CONJURING</h1>Lorem ipsum dolor sit amet, consectetur adipiscing elit. Cupiditate eveniet nihil illum nemo ex nam, aut dolorem sunt molestiae minus sint deleniti odit necessitatibus? Minus ut, alias numquam iure vel, facilis labore molestias voluptatibus dolores ex officia tempora ab fugiat. Voluptas atque vero modi deleniti magni dolor mollitia autem aliquam illo commodi maiores cum soluta aut, laudantium minima possimus ex earum eligendi harum! Earum, temporibus. Sunt nisi animi dolorem corrupti nihil assumenda cum labore id. Facere ullam provident repellendus voluptatem doloribus. Laborum ullam possimus veritatis repellat impedit quis repudiandae cupiditate tempore. Nemo accusantium facilis laborum quia vero aspernatur reprehenderit quasi deleniti voluptatem quaerat eos, nihil totam itaque assumenda ad? Lure nihil, odio accusamus eius omnis porro debitis quos, enim architecto cum quis praesentium magni tenetur numquam unde vero veritatis blanditiis? Labore similique delectus illum praesentium aliquid dolor fuga quam, voluptatibus asperiores eius facere velit dignissimos voluptatem sint impedit ab molestiae animi rem sed iure. A rem iusto laudantium sunt. Delectus quo quisquam soluta, nam aperiam sapiente distinctio eaque tempora praesentium doloribus quis ut maiores reprehenderit iure minus blanditiis dolor amet ipsam temporibus quidem. Obcaecati totam quia vel beatae, sunt, reiciendis cumque aperiam aspernatur numquam laudantium non incidunt exercitationem officiis illum expedita ad eveniet! Eveniet exercitationem enim possimus, quos libero nam excepturi neque illum fugiat, earum quod nesciunt veritatis rerum tempore, a quasi deleniti. Vel earum nesciunt consectetur, error dolore minus officia magni quaerat doloribus laborum quisquam aspernatur deserunt iusto debitis numquam tempore vitae deleniti consequuntur. Nostrum a, nesciunt culpa necessitatibus sequi illum quisquam sed delectus ipsam sunt amet obcaecati optio aliquam quibusdam repellendus provident numquam minima nobis tempora quasi praesentium! Esse officia expedita similique distinctio quasi debitis, magnam saepe, architecto laboriosam voluptatem eligendi, laudantium consequuntur incidunt quo ea! Omnis aspernatur quibusdam laborum aliquid inventore, accusamus recusandae porro neque impedit a repellat quaerat saepe sit soluta corrupti sapiente placeat illo enim, provident unde? Culpa quia nesciunt repudiandae minus quo aut cum, doloremque inventore magni esse illum nostrum, voluptas placeat nam eos! Consequuntur cumque vel officiis. Numquam nobis corrupti blanditiis quaerat. Dolores accusantium cum, rem aut saepe nemo iusto exercitationem facere magnam sit voluptatum eius repudiandae doloremque fugiat temporibus corrupti molestias quos quo reiciendis sequi. Excepturi sit non itaque dolore sint consequuntur quas voluptatum magni doloremque sequi est ea, natus, possimus ut pariat fugam nobis commodi enim harum libero. Dicta totam ipsum est delectus ea culpa, accusamus voluptates eligendi quisquam fugiat sint! </div>

<div class="box5"></div>

<div class="box2"><h1>2.INSIDIOUS</h1>Lorem ipsum dolor sit amet, consectetur adipiscing elit. Cupiditate eveniet nihil illum nemo ex nam, aut dolorem sunt molestiae minus sint deleniti odit necessitatibus? Minus ut, alias numquam iure vel, facilis labore molestias voluptatibus dolores ex officia tempora ab fugiat. Voluptas atque vero modi deleniti magni dolor mollitia autem aliquam illo commodi maiores cum soluta aut, laudantium minima possimus ex earum eligendi harum! Earum, temporibus. Sunt nisi animi dolorem corrupti nihil

assumenda cum labore id. Facere ullam provident repellendus voluptatem doloribus. Laborum ullam possimus veritatis repellat impedit quis repudiandae cupiditate tempore. Nemo accusantium facilis laborum quia vero aspernatur reprehenderit quasi deleniti voluptatem quaerat eos, nihil totam itaque assumenda ad? lure nihil, odio accusamus eius omnis porro debitis quos, enim architecto cum quis praesentium magni tenetur numquam unde vero veritatis blanditiis? Labore similique delectus illum praesentium aliquid dolor fuga quam, voluptatibus asperiores eius facere velit dignissimos voluptatem sint impedit ab molestiae animi rem sed iure. A rem iusto laudantium sunt. Delectus quo quisquam soluta, nam aperiam sapiente distinctio eaque tempora praesentium doloribus quis ut maiores reprehenderit iure minus blanditiis dolor amet ipsam temporibus quidem. Obcaecati totam quia vel beatae, sunt, reiciendis cumque aperiam aspernatur numquam laudantium non incidunt exercitationem officiis illum expedita ad eveniet! Eveniet exercitationem enim possimus, quos libero nam excepturi neque illum fugiat, earum quod nesciunt veritatis rerum tempore, a quasi deleniti. Vel earum nesciunt consecetur, error dolore minus officia magni quaerat doloribus laborum quisquam aspernatur deserunt iusto debitis numquam tempore vitae deleniti consequuntur. Nostrum a, nesciunt culpa necessitatibus sequi illum quisquam sed delectus ipsam sunt amet obcaecati optio aliquam quibusdam repellendus provident numquam minima nobis tempora quasi praesentium! Esse officia expedita similique distinctio quasi debitis, magnam saepe, architecto laboriosam voluptatem eligendi, laudantium consequuntur incidunt quo ea! Omnis aspernatur quibusdam laborum aliquid inventore, accusamus recusandae porro neque impedit a repellat quaerat saepe sit soluta corrupti sapiente placeat illo enim, provident unde? Culpa quia nesciunt repudiandae minus quo aut cum, doloremque inventore magni esse illum nostrum, voluptas placeat nam eos! Consequuntur cumque vel officiis. Numquam nobis corrupti blanditiis quaerat. Dolores accusantium cum, rem aut saepe nemo iusto exercitationem facere magnam sit voluptatum eius repudiandae doloremque fugiat temporibus corrupti molestias quos quo reiciendis sequi. Excepturi sit non itaque dolore sint consequuntur quas voluptatum magni doloremque sequi est ea, natus, possimus ut pariat fuga nobis commodi enim harum libero. Dicta totam ipsum est delectus ea culpa, accusamus voluptates eligendi quisquam fugiat sint! </div>

<div class="box6"></div>

<div class="box3"><h1>3.THE EXORCIST</h1>Lorem ipsum dolor sit amet, consectetur adipisicing elit. Cupiditate eveniet nihil illum nemo ex nam, aut dolorem sunt molestiae minus sint deleniti odit necessitatibus? Minus ut, alias numquam iure vel, facilis labore molestias voluptatibus dolores ex officia tempora ab fugiat. Voluptas atque vero modi deleniti magni dolor mollitia autem aliquam illo commodi maiores cum soluta aut, laudantium minima possimus ex earum eligendi harum! Earum, temporibus. Sunt nisi animi dolorem corrupti nihil assumenda cum labore id. Facere ullam provident repellendus voluptatem doloribus. Laborum ullam possimus veritatis repellat impedit quis repudiandae cupiditate tempore. Nemo accusantium facilis laborum quia vero aspernatur reprehenderit quasi deleniti voluptatem quaerat eos, nihil totam itaque assumenda ad? lure nihil, odio accusamus eius omnis porro debitis quos, enim architecto cum quis praesentium magni tenetur numquam unde vero veritatis blanditiis? Labore similique delectus illum praesentium aliquid dolor fuga quam, voluptatibus asperiores eius facere velit dignissimos voluptatem sint impedit ab molestiae animi rem sed iure. A rem iusto laudantium sunt. Delectus quo quisquam soluta, nam aperiam sapiente distinctio eaque tempora praesentium doloribus quis ut maiores reprehenderit iure minus blanditiis dolor amet ipsam temporibus quidem. Obcaecati totam quia vel beatae, sunt, reiciendis cumque aperiam aspernatur numquam laudantium non incidunt exercitationem



officiis illum expedita ad eveniet! Eveniet exercitationem enim possimus, quos libero nam excepturi neque illum fugiat, earum quod nesciunt veritatis rerum tempore, a quasi deleniti. Vel earum nesciunt consecetur, error dolore minus officia magni quaerat doloribus laborum quisquam aspernatur deserunt iusto debitis numquam tempore vitae deleniti consequuntur. Nostrum a, nesciunt culpa necessitatibus sequi illum quisquam sed delectus ipsam sunt amet obcaecati optio aliquam quibusdam repellendus provident numquam minima nobis tempora quasi praesentium! Esse officia expedita similique distinctio quasi debitis, magnam saepe, architecto laboriosam voluptatem eligendi, laudantium consequuntur incidunt quo ea! Omnis aspernatur quibusdam laborum aliquid inventore, accusamus recusandae porro neque impedit a repellat quaerat saepe sit soluta corrupti sapiente placeat illo enim, provident unde? Culpa quia nesciunt repudiandae minus quo aut cum, doloremque inventore magni esse illum nostrum, voluptas placeat nam eos! Consequuntur cumque vel officiis. Numquam nobis corrupti blanditiis quaerat. Dolores accusantium cum, rem aut saepe nemo iusto exercitationem facere magnam sit voluptatum eius repudiandae doloremque fugiat temporibus corrupti molestias quos quo reiciendis sequi. Excepturi sit non itaque dolore sint consequuntur quas voluptatum magni doloremque sequi est ea, natus, possimus ut pariat fuga nobis commodi enim harum libero. Dicta totam ipsum est delectus ea culpa, accusamus voluptates eligendi quisquam fugiat sint! </div>

<div class="box7"></div>

<div class="box4"><h1>4.THE NUN</h1>Lorem ipsum dolor sit amet, consecetur adipisicing elit. Cupiditate eveniet nihil illum nemo ex nam, aut dolorem sunt molestiae minus sint deleniti odit necessitatibus? Minus ut, alias numquam iure vel, facilis labore molestias voluptatibus dolores ex officia tempora ab fugiat. Voluptas atque vero modi deleniti magni dolor mollitia autem aliquam illo commodi maiores cum soluta aut, laudantium minima possimus ex earum eligendi harum! Earum, temporibus. Sunt nisi animi dolorem corrupti nihil assumenda cum labore id. Facere ullam provident repellendus voluptatem doloribus. Laborum ullam possimus veritatis repellat impedit quis repudiandae cupiditate tempore. Nemo accusantium facilis laborum quia vero aspernatur reprehenderit quasi deleniti voluptatem quaerat eos, nihil totam itaque assumenda ad? Iure nihil, odio accusamus eius omnis porro debitis quos, enim architecto cum quis praesentium magni tenetur numquam unde vero veritatis blanditiis? Labore similique delectus illum praesentium aliquid dolor fuga quam, voluptatibus asperiores eius facere velit dignissimos voluptatem sint impedit ab molestiae animi rem sed iure. A rem iusto laudantium sunt. Delectus quo quisquam soluta, nam aperiam sapiente distinctio eaque tempora praesentium doloribus quis ut maiores reprehenderit iure minus blanditiis dolor amet ipsam temporibus quidem. Obcaecati totam quia vel beatae, sunt, reiciendis cumque aperiam aspernatur numquam laudantium non incidunt exercitationem officiis illum expedita ad eveniet! Eveniet exercitationem enim possimus, quos libero nam excepturi neque illum fugiat, earum quod nesciunt veritatis rerum tempore, a quasi deleniti. Vel earum nesciunt consecetur, error dolore minus officia magni quaerat doloribus laborum quisquam aspernatur deserunt iusto debitis numquam tempore vitae deleniti consequuntur. Nostrum a, nesciunt culpa necessitatibus sequi illum quisquam sed delectus ipsam sunt amet obcaecati optio aliquam quibusdam repellendus provident numquam minima nobis tempora quasi praesentium! Esse officia expedita similique distinctio quasi debitis, magnam saepe, architecto laboriosam voluptatem eligendi, laudantium consequuntur incidunt quo ea! Omnis aspernatur quibusdam laborum aliquid inventore, accusamus recusandae porro neque impedit a repellat

quaerat saepe sit soluta corrupti sapiente placeat illo enim, provident unde? Culpa quia nesciunt repudiandae minus quo aut cum, doloremque inventore magni esse illum nostrum, voluptas placeat nam eos! Consequuntur cumque vel officiis. Numquam nobis corrupti blanditiis quaerat. Dolores accusantium cum, rem aut saepe nemo iusto exercitationem facere magnam sit voluptatum eius repudiandae doloremque fugiat temporibus corrupti molestias quos quo reiciendis sequi. Excepturi sit non itaque dolore sint consequuntur quas voluptatum magni doloremque sequi est ea, natus, possimus ut pariat fuga nobis commodi enim harum libero. Dicta totam ipsum est delectus ea culpa, accusamus voluptates eligendi quisquam fugiat sint! </div>

<div class="box8"><div class="center1">THANKS FOR VISIT US</div></div>

</body>

</html>

## Day 18: FORM DESIGN FOR ANY GROCERY STORE

**Date:** 27-06-24

Code: <!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

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

<title>Document</title>

<link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/6.5.2/css/all.min.css">

<style>\*{

padding:0;

margin:0;

box-sizing:border-box;

}

header{

height:100vh;

width:100%;

```
background:url(https://c.pxhere.com/photos/2c/e6/supermarket_nice_woma-898411.jpg!d) no-repeat center/cover fixed;
```

```
opacity:0.8;
```

```
position:relative;
```

```
overflow:hidden;
```

```
} nav .logo{
```

```
font-size:60px;
```

```
transition:all 2s;
```

```
margin-left:50px;
```

```
gap:30px;
```

```
}
```

```
nav ul {
```

```
display:flex;
```

```
flex-direction:row;
```

```
justify-content:center;
```

```
font-size:25px;
```

```
gap:70px;
```

```
flex-wrap:wrap;
```

```
float:right;
```

```
margin-right:100px;
```

```
}
```

```
a{
```

```
text-decoration:none;
```

```
font-weight:bold;
```

```
} a:hover{  
    color:grey;  
}
```

```
.box{  
height:450px;  
width:450px;  
border:2px solid yellowgreen;  
border-radius:20px;  
position:absolute;  
top:150px;  
left:150px;  
box-shadow:0 0 8px black;  
text-align:center;  
backdrop-filter:blur(1px);  
  
}
```

```
h3{  
    font-size:30px;  
    font-family:'Trebuchet MS', 'Lucida Sans Unicode', 'Lucida Grande', 'Lucida Sans', Arial,  
sans-serif;  
    text-align:center;  
    color:rgb(76, 76, 76);  
    margin:0 5px;
```

```
}input[type=text],input[type=tel],textarea{  
    text-align:center;  
    font-size:20px;
```

```
font-family:'Trebuchet MS', 'Lucida Sans Unicode', 'Lucida Grande', 'Lucida Sans', Arial,
sans-serif;

margin:20px auto;

border:2px solid transparent;

border-radius:20px;

box-shadow:0 0 8px grey;

}
```

```
</style>

</head>

<body><header>

  <div class="container">

    <nav>

      <label class="logo"> <i class="fa-solid fa-cart-shopping"></i></label>

      <ul type="none">

        <li><a href="#">HOME</a></li>

        <li><a href="#">ABOUT US</a></li>

        <li><a href="#">GROCERY LIST</a></li>

        <li><a href="#">CONTACT US</a></li> </ul>

    </nav></div></header>

    <div class="box">

      <form action="">

        <h3>Register For The Grocery service</h3>

        <input type="text" placeholder="Enter Your Name"><br/>

        <input type="tel" placeholder=" Enter Mobile Number"><br/>

        <input type="text" placeholder="Enter your Pincode"><br/>

        <textarea placeholder="Enter Full Address"></textarea><br/>
```

</form>

</div>

</body>

</html>

---

## Day 19: discussion related to project on HTML and CSS

**Date:** 28-06-24

Today was a pivotal day for my project. I had a meeting with my teacher to discuss my progress, and it turned out to be really helpful. Here's how it went:

### Preparation:

Before the meeting, I took some time to review the work I've done so far. I made a list of the main points I wanted to address during our discussion:

- What aspects of the project are going well?
- What challenges have I encountered?
- What areas of the project need improvement?
- Clarifications on the project requirements (if any).

I also gathered my project materials—notes, drafts, research, and the outline—so I'd have everything I might need for the meeting.

**The Discussion:** The conversation was really productive. I first shared my progress and explained what I've completed up to this point. My teacher asked questions to ensure I was on the right track and to see if I'd missed any important details.

### Key Points from the Meeting:

- **Strengths of the Project:** My teacher said I'm doing well with the research and the overall structure, especially the way I've organized the information.
- **Areas for Improvement:** One of the areas that needs more work is the analysis section. My teacher suggested that I should go deeper into the interpretation of my findings and explain the significance more clearly.
- **Next Steps:** Based on the feedback, I now know exactly where to focus my energy over the next few days. I need to refine the analysis, double-check my citations, and ensure that the conclusion ties everything together logically.
- **Clarifications:** We also discussed some of the project guidelines, especially the formatting requirements, which I wasn't entirely sure about. Now I have a clear understanding of what is expected.

**Personal Reflections:**

I feel more confident after today's discussion. I was a bit worried about whether I was heading in the right direction, but my teacher's feedback reassured me that I'm on track. The areas of improvement are manageable, and I now have a clearer roadmap for how to move forward.

I also appreciated the constructive criticism. It's easy to get stuck in your own perspective, but getting an outside point of view has been invaluable.

**To-Do List:**

- Revise the analysis section.
- Double-check formatting and citations.
- Finish the conclusion.
- Plan a follow-up meeting (if needed).

It feels like everything is starting to come together!

---

**Day 20: Integrating HTML, CSS, and JavaScript basic**

**Date:** 1-07-24

**1. Project Requirements:**

- Create a webpage that includes a structured HTML layout.
- Apply CSS to style the webpage and make it visually appealing.
- Add JavaScript to introduce interactivity, such as event handling and DOM manipulation.

**2. HTML Structure:**

- Basic structure with DOCTYPE, head, and body tags.
- Use of semantic elements like <header>, <nav>, <main>, and <footer>.
- Inclusion of elements like paragraphs, headings, lists, images, links, forms, and tables.

**Example HTML:**

```
<!DOCTYPE html>
```

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

```
<head>
```

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

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

```
<title>My Webpage</title>
```

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

```
</head>

<body>

  <header>

    <h1>Welcome to My Webpage</h1>

    <nav>

      <ul>

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

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

        <li><a href="#contact">Contact</a></li>

      </ul>

    </nav>

  </header>

  <main>

    <section id="home">

      <h2>Home</h2>

      <p>This is the home section.</p>

    </section>

    <section id="about">

      <h2>About</h2>

      <p>This is the about section.</p>

    </section>

    <section id="contact">

      <h2>Contact</h2>

      <form id="contactForm">

        <label for="name">Name:</label>

        <input type="text" id="name" name="name">

        <label for="email">Email:</label>

        <input type="email" id="email" name="email">

        <button type="button" onclick="submitForm()">Submit</button>

      </form>

      <div id="formMessage"></div>

    </section>

  </main>

</body>

</html>
```



</section>

<section id="data">

<h2>Data Table</h2>

<table>

<thead>

<tr>

<th>Name</th>

<th>Age</th>

<th>City</th>

</tr>

</thead>

<tbody>

<tr>

<td>John Doe</td>

<td>25</td>

<td>New York</td>

</tr>

<tr>

<td>Jane Smith</td>

<td>30</td>

<td>Los Angeles</td>

</tr>

</tbody>

</table>

</section>

</main>

<footer>

<p>&copy; 2024 My Webpage</p>

</footer>

<script src="script.js"></script>

</body>

</html>

### 3. CSS Styling:

- Use of external CSS to style the webpage.
- Application of layout properties like flexbox and grid.
- Styling of elements such as headers, navigation menus, sections, forms, and tables.

#### Example CSS (styles.css):

```
body {  
  font-family: Arial, sans-serif;  
  margin: 0;  
  padding: 0;  
  box-sizing: border-box;  
}
```

```
header {  
  background-color: #4CAF50;  
  color: white;  
  padding: 1em;  
  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;
}
```

```
main {
padding: 1em;
}
```

```
section {
margin-bottom: 2em;
}
```

```
form {
display: flex;
flex-direction: column;
}
```

```
form label, form input, form button {
margin-bottom: 10px;
}
```

```
table {
width: 100%;
border-collapse: collapse;
}
```

```
table, th, td {
border: 1px solid black;
}
```

```
th, td {
```

```
padding: 8px;
text-align: left;
}
```

```
footer {
background-color: #333;
color: white;
text-align: center;
padding: 1em;
position: fixed;
width: 100%;
bottom: 0;
}
```

#### **LAST DAY: Some tasks for revision:**

##### **Code:**

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Document</title>
  <style>*{
    box-sizing:border-box;

  }
  body{
    background: #30E8BF; /* fallback for old browsers */
background: -webkit-linear-gradient(to right, #FF8235, #30E8BF); /* Chrome 10-25, Safari 5.1-6
*/
background: linear-gradient(to right, #FF8235, #30E8BF); /* W3C, IE 10+/ Edge, Firefox 16+,
Chrome 26+, Opera 12+, Safari 7+ */
```

```
.container{  
  height:500px;  
  width:450px;  
  border:7px solid;  
  border-radius:20px;  
  position:absolute;  
  top:50%;  
  left:50%;  
  transform:translate(-50%,-50%);  
  perspective:900px;  
}
```

```
.container .box-img{  
  height:100%;  
  width:100%;  
  background-color:aliceblue;  
  border-radius:20px;  
  position:absolute;
```

```
}
```

```
.box-img img{  
  height:100%;  
  width:100%;  
  opacity:0.9;  
  border-radius:20px;
```

```
}
```

```
.container .box1{  
  height:100%;  
  width:100%;  
  border-radius:20px;  
  transition:all 1s;
```

```
border:5px solid;
position:absolute;
top:-100%;
opacity:0;
transition:all 0.5s;
background-color:lightcyan;
}
.container:hover .box1{
opacity:1;
top:0;
}
```

```
h1{
text-align:center;
font-size:60px;
font-family:cursive;
color:black;
```

```
}p{
line-height:1.5;
color:black;
font-family:cursive;
font-size:20px;
}
```

```
</style>
```

```
</head>

<body>

  <div class="container">

    <div class="box-img"></div>

    <div class="box1"><h1>PINK ROSE</h1><p>Lorem ipsum dolor sit amet consectetur
adipisicing elit. Aliquam voluptatum aperiam laboriosam accusantium! Pariatur quam eum
aliquam maxime facilis quis officiis eius eos voluptate, quas veritatis qui odit rem consectetur
exercitationem aut magnam, repellendus corrupti. Fuga fugit blanditiis fugiat in?</p></div>

  </div>

</body>

</html>
```

#### **CODE:**

```
<!DOCTYPE html>

<html lang="en">

<head>

  <meta charset="UTF-8">

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

  <title>Document</title>

  <style>

.container{

  height:100vh;

  width:100%;

  box-sizing:border-box;

  position:relative;

  animation:change 10s linear 0s infinite;

}

@keyframes change{

  0%{
```

```
background:url(https://tse2.mm.bing.net/th?id=OIP.vuFRpwPLqeVOKkfYARhKYgHaEo&pid=Api&P=0&h=180) no-repeat center/cover;
```

```
transition:all 2s;
```

```
opacity:0.8;
```

```
}50%{
```

```
background:url(https://wallpaperaccess.com/full/36301.jpg) no-repeat center/cover;
```

```
transition:all 2s;
```

```
opacity:0.5;
```

```
}100%{
```

```
background:url(https://images.pexels.com/photos/459225/pexels-photo-459225.jpeg?cs=srgb&dl=daylight-environment-forest-459225.jpg&fm=jpg) no-repeat center/cover;
```

```
transition:all 2s;
```

```
opacity:0.7;
```

```
}
```

```
}
```

```
img{
```

```
height:250px;
```

```
width:250px;
```

```
animation:bird 10s linear 0s infinite;
```

```
@keyframes bird{
```

```
0%{
```

```
position:absolute;
```

```
top:0px;
```

```
left:100px;
```

```
}20%{
```

```
position:absolute;
```

```
top:0px;
```



```
    left:300px}

    40%{

    position:absolute;

    top:0px;

    left:700px;}

    60%{

    position:absolute;

    top:0px;

    left:1100px;}

    80%{

    position:absolute;

    top:0px;

    left:1500px;

    }100%{

    position:absolute;

    top:0px;

    right:-300px;

    }

}

</style>

</head>

<body>

<div class="container">

<div class="img2"> 
```

</div>

</body>

</html>