HTML ASSIGNMENT

• Question 1: Define HTML. What is the purpose of HTML in web development?

ans => HTML, or HyperText Markup Language, is the standard markup language used to create and design documents on the World Wide Web. It provides the basic structure for web pages and web applications, allowing developers to define elements such as headings, paragraphs, links, images, tables, and other content.

• Question 2: Explain the basic structure of an HTML document. Identify the mandatory tagsand their purposes.

ans =>
```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Document Title</title>
</head>
<body>
    <h1>Hello, World!</h1>
    <p>This is a sample HTML document.</p>
</body>
</html>
```

• Question 3: What is the difference between block-level elements and inline elements inHTML? Provide examples of each ?

ans => Block-Level Elements

Examples:

: A generic container for grouping content.
<h1>, <h2>, <h3>, <h4>, <h5>, <h6>: Headings of different levels.
<p>: Paragraphs of text.
: Unordered lists.
: Ordered lists.
: List items (used within <ul> or <ol>).
<header>, <footer>, <section>, <article>: Semantic elements that define different parts of a web page.

Inline Elements

Examples:

: A generic inline container for text or other inline elements.
<a>: Anchor tag for hyperlinks.
: Indicates strong importance (usually displayed as bold).
: Indicates emphasized text (usually displayed as italic).
<img>: Embeds an image.

: Inserts a line break.
<code>: Represents a fragment of computer code.

Question 4: Discuss the role of semantic HTML. Why is it important for accessibility andSEO? Provide examples of semantic elements?

ans => Importance of Semantic HTML Accessibility:

Improved Screen Reader Interpretation:
Semantic elements provide context to assistive technologies, such as screen readers, which help visually impaired users navigate and understand web content. For example, using <header>, <nav>, and <footer> allows screen readers to identify the structure of the page easily.
Better User Experience: By using semantic HTML, developers can create a more intuitive experience for all users, including those with disabilities. This can lead to better navigation and comprehension of the content.

Search Engine Optimization (SEO):
Enhanced Search Engine Understanding: Search engines use semantic HTML to better understand the content and context of a web page. This can improve the page's ranking in search results. For instance, using <article> for blog posts helps search engines recognize that the content is an article, which can be indexed accordingly.
Rich Snippets: Semantic HTML can help in generating rich snippets in search results, which can improve click-through rates. For example, using <time> to mark up dates can allow search engines to display the date of an article in search results.
Maintainability:

LAB ASSIGNMENT :

• Task: Create a simple HTML webpage that includes:

Q 5 => • A header (<header>), footer (<footer>), main section (<main>), and aside section(<aside>). ?

ans =>
```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Sample Page</title>
    <link rel="stylesheet" href="styles.css">
</head>
<body>

    <header>
        <h1>Welcome to My Website</h1>
        <nav>
            <ul>
                <li><a href="#home">Home</a></li>
                <li><a href="#about">About</a></li>
                <li><a href="#services">Services</a></li>
                <li><a href="#contact">Contact</a></li>
            </ul>
        </nav>
```

```
    </header>

    <main>
        <h2>Main Content Area</h2>
        <p>This is where the main content of the page goes. You can include articles,
images, and other relevant information here.</p>
    </main>

    <aside>
        <h3>Related Links</h3>
        <ul>
            <li><a href="#link1">Link 1</a></li>
            <li><a href="#link2">Link 2</a></li>
            <li><a href="#link3">Link 3</a></li>
        </ul>
    </aside>

    <footer>
        <p>&copy; 2023 My Website. All rights reserved.</p>
        <p><a href="#privacy">Privacy Policy</a> | <a href="#terms">Terms of
Service</a></p>
    </footer>

</body>
</html>
```

Q 6 => • A paragraph with some basic text.?

ans =>

```
<p>
    Lorem ipsum dolor sit amet, consectetur adipiscing elit. Sed do eiusmod tempor
incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud
exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure
dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur.
Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit
anim id est laborum.
</p>
```

Q 7 => • A list (both ordered and unordered).?

ans =>

```
<h3>Unordered List</h3>
<ul>
    <li>Apples</li>
    <li>Bananas</li>
    <li>Cherries</li>
    <li>Dates</li>
</ul>
```

Q 8 • A link that opens in a new tab.?

ans =>

```
<a href="https://www.example.com" target="_blank">Visit Example.com</a>
```

• HTML FORM

Question 1: What are HTML forms used for? Describe the purpose of the input, textarea, select, and buttonelements.?

ans=>1. Input Element
The <input> element is a versatile form element that can be used to create various types of input fields. The type of input is specified using the type attribute, which can take values such as:

text: A single-line text input.
password: A single-line input that hides the text entered (for passwords).
email: A field for entering an email address, with built-in validation.
number: A field for entering numeric values, with optional constraints (like min and max).
checkbox: A box that can be checked or unchecked, allowing for binary choices.
radio: A set of options where only one can be selected at a time.
file: A field for uploading files.
submit: A button that submits the form.

2. Textarea Element
The <textarea> element is used for multi-line text input. It allows users to enter larger amounts of text, such as comments or messages. The size of the textarea can be controlled using the rows and cols attributes, or through CSS for more precise styling. Unlike the <input> element, which is typically for single-line input, <textarea> is ideal for longer text entries.

3. Select Element
The <select> element creates a dropdown list from which users can select one or more options. It can contain multiple <option> elements, each representing a choice. The multiple attribute can be added to allow users to select more than one option. This element is useful for scenarios where space is limited or when you want to present a list of options without taking up too much screen real estate.

4. Button Element
The <button> element is used to create clickable buttons within a form. It can be used for various purposes, such as submitting the form, resetting the form fields, or triggering JavaScript functions. The type attribute can specify the button's behavior:

submit: Submits the form data to the server.
reset: Resets all form fields to their default values.
button: A generic button that can be programmed to perform custom actions using JavaScript.

Question 2 : Explain the difference between the GETand POSTmethods in form submission. When should each be used?

ans=> GET Method

Data Transmission: The GET method appends form data to the URL in the form of a query string. For example, if a user submits a form with the GET method, the data is sent as part of the URL, like this: http://example.com/form?name=John&age=30.

Visibility: Since the data is included in the URL, it is visible to the user. This means that sensitive information (like passwords) should not be sent using GET.

Data Length Limit: URLs have length limitations (typically around 2000 characters in many browsers), which restricts the amount of data that can be sent using the GET method.

Caching: GET requests can be cached by browsers and are generally considered safe and idempotent, meaning they can be repeated without causing side effects.

Use Cases: The GET method is best used for:

Retrieving data from the server (e.g., search queries, filtering data).
Bookmarking or sharing URLs since the parameters are part of the URL.
When the data being sent is not sensitive and is relatively small.

POST Method:
Data Transmission: The POST method sends form data in the body of the HTTP request, not in the URL. This allows for larger amounts of data to be sent without being visible in the URL.

Visibility: Data sent via POST is not displayed in the URL, making it more suitable for sensitive information (like passwords or personal data).

Data Length Limit: There is no strict limit on the amount of data that can be sent using POST, making it suitable for larger payloads, such as file uploads.

Caching: POST requests are not cached by default, and they are not idempotent, meaning that submitting the same POST request multiple times can result in different outcomes (e.g., creating multiple records).

Use Cases: The POST method is best used for:

Submitting forms that change server state (e.g., creating, updating, or deleting resources).
Sending sensitive information that should not be exposed in the URL.
Uploading files or large amounts of data.

Question 3: What is the purpose of the labelelement in a form, and how does it improve accessibility?

ans=> Purpose of the <label> Element:

Descriptive Text:
The <label> element provides a clear description of the form control, helping users understand what data they need to enter. For example, a label like "Email Address" next to an input field indicates that the user should enter their email address.

Association with Form Controls:
The <label> element can be associated with a specific form control using the for attribute. The value of the for attribute should match the id of the corresponding form control. This association enhances usability by allowing users to click on the label to focus on the associated input field.

How <label> Improves Accessibility
Screen Reader Support: Screen readers, which are used by visually impaired users, can read the text of the <label> element when the associated form control is focused. This provides context and helps users understand what information they need to provide.

Clickable Labels:
When a label is associated with a form control, clicking on the label will focus the corresponding input field. This is particularly helpful for users with motor impairments or those using touch devices, as it makes it easier to interact with form elements.

Improved Usability:
Labels help all users, including those with cognitive disabilities, by providing clear instructions and context for each form field. This reduces confusion and enhances the overall user experience.

Visual Clarity:
Labels can improve the visual layout of forms, making them more organized and easier to read. This is especially important for long forms, where clear labeling can help users navigate through the fields more effectively.

LAB ASSIGNMENT:

• Task: Create a contact form with the following fields:

1 • Full name (text input):

```
<form>
    <label for="fullName">Full Name:</label>
    <input type="text" id="fullName" name="fullName" placeholder="Enter your full name"
required>
    <input type="submit" value="Submit">
</form>
```

2• Email (email input)

```
<form>
    <label for="email">Email:</label>
    <input type="email" id="email" name="email" placeholder="Enter your email" required>
    <input type="submit" value="Submit">
</form>
```

3• Phone number (tel input)

```
<form>
    <label for="phone">Phone Number:</label>
    <input type="tel" id="phone" name="phone" placeholder="Enter your phone number"
required>
    <input type="submit" value="Submit">
</form>
```

4• Subject (dropdown menu)

```
<form>
    <label for="subject">Subject:</label>
    <select id="subject" name="subject" required>
        <option value="" disabled selected>Select a subject</option>
        <option value="general_inquiry">General Inquiry</option>
        <option value="support">Support</option>
        <option value="sales">Sales</option>
        <option value="feedback">Feedback</option>
    </select>
    <input type="submit" value="Submit">
</form>
```

5• Message (textarea)

```
<form>
    <label for="message">Message:</label>
```

```
    <textarea id="message" name="message" rows="4" cols="50" placeholder="Enter your
message here..." required></textarea>
    <input type="submit" value="Submit">
</form>
```

6 • Submit button

```
<form>
    <label for="name">Full Name:</label>
    <input type="text" id="name" name="name" placeholder="Enter your full name" required>

    <label for="email">Email:</label>
    <input type="email" id="email" name="email" placeholder="Enter your email" required>

    <label for="phone">Phone Number:</label>
    <input type="tel" id="phone" name="phone" placeholder="Enter your phone number"
required>

    <label for="subject">Subject:</label>
    <select id="subject" name="subject" required>
        <option value="" disabled selected>Select a subject</option>
        <option value="general_inquiry">General Inquiry</option>
        <option value="support">Support</option>
        <option value="sales">Sales</option>
        <option value="feedback">Feedback</option>
    </select>

    <label for="message">Message:</label>
    <textarea id="message" name="message" rows="4" cols="50" placeholder="Enter your
message here..." required></textarea>

    <input type="submit" value="Submit">
</form>
```

Additional Requirements:

Q1 • Use appropriate form validation using required, minlength, maxlength, and pattern?

ANS =>

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Contact Form with Validation</title>
</head>
<body>

    <h2>Contact Form</h2>
    <form>
        <label for="name">Full Name:</label>
        <input type="text" id="name" name="name" placeholder="Enter your full name"
required minlength="3" maxlength="50">

        <label for="email">Email:</label>
        <input type="email" id="email" name="email" placeholder="Enter your email"
required>

        <label for="phone">Phone Number:</label>
        <input type="tel" id="phone" name="phone" placeholder="Enter your phone number"
required pattern="[0-9]{10}" title="Please enter a 10-digit phone number without spaces
or dashes.">
```

```
        <label for="subject">Subject ●
```

Q2 • Link form labels with their corresponding inputs using the forattribute.

ANS =>

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Contact Form with Linked Labels</title>
</head>
<body>

    <h2>Contact Form</h2>
    <form>
        <label for="name">Full Name:</label>
        <input type="text" id="name" name="name" placeholder="Enter your full name"
required minlength="3" maxlength="50">

        <label for="email">Email:</label>
        <input type="email" id="email" name="email" placeholder="Enter your email"
required>

        <label for="phone">Phone Number:</label>
        <input type="tel" id="phone" name="phone" placeholder="Enter your phone number"
required pattern="[0-9]{10}" title="Please enter a 10-digit phone number without spaces
or dashes.">

        <label for="subject">Subject:</label>
        <select id="subject" name="subject" required>
            <option value="" disabled selected>Select a subject</option>
            <option value="general_inquiry">General Inquiry</option>
            <option value="support">Support</option>
            <option value="sales">Sales</option>
            <option value="feedback">Feedback</option>
        </select>

        <label for="message">Message:</label>
        <textarea id="message" name="message" rows="4" cols="50" placeholder="Enter your
message here..." required minlength="10" maxlength="500"></textarea>

        <input type="submit" value="Submit">
    </form>

</body>
</html>
```

Lab Assignment

• Task: Create a product catalog table that includes the following columns:

Q1 : Product Name
• Product Image (use placeholder image URLs)
• Price
• Description
• Availability (in stock, out ofstock)

ANS=>

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Product Information Form</title>
</head>
<body>

    <h2>Product Information</h2>
    <form>
        <label for="productName">Product Name:</label>
        <input type="text" id="productName" name="productName" placeholder="Enter product
name" required minlength="2" maxlength="100">

        <label for="productImage">Product Image URL:</label>
        <input type="url" id="productImage" name="productImage" placeholder="Enter image
URL" required>
        <img id="imagePreview" src="https://via.placeholder.com/150" alt="Product Image
Preview" style="display: block; margin-top: 10px;">

        <label for="price">Price ($):</label>
        <input type="number" id="price" name="price" placeholder="Enter product price"
required min="0" step="0.01">

        <label for="description">Description:</label>
        <textarea id="description" name="description" rows="4" cols="50"
placeholder="Enter product description" required minlength="10"
maxlength="500"></textarea>

        <label for="availability">Availability:</label>
        <select id="availability" name="availability" required>
            <option value="" disabled selected>Select availability</option>
            <option value="in_stock">In Stock</option>
            <option value="out_of_stock">Out of Stock</option>
        </select>

        <input type="submit" value="Submit">
    </form>

</body>
</html>
```

• HTML Tables

• Question 1: Explain the structure of an HTML table and the purpose of each of the followingelements: <table>, <tr>, <th>, <td>, and <thead>.?

ans=>1. <table>

Purpose: The <table> element is the container for the entire table. It defines the boundaries of the table and encompasses all the other table-related elements.

Usage: It is used to create a table structure in HTML. You can also use attributes like border, cellpadding, and cellspacing (though CSS is preferred for styling) to control the appearance of the table.

2. <tr>

Purpose: The <tr> element stands for "table row." It is used to define a row within the table.
Usage: Each <tr> element contains one or more <th> (table header) or <td> (table data) elements, representing the cells in that row. You can have multiple <tr> elements within a <table> to create multiple rows.

3. <th>

Purpose: The <th> element stands for "table header." It is used to define a header cell in a table, which typically contains headings for the columns.
Usage: Header cells are usually bold and centered by default. They help to describe the content of the corresponding column or row, making the table easier to understand. <th> elements can be used within <tr> elements to create header rows.

4. <td>

Purpose: The <td> element stands for "table data." It is used to define a standard cell in a table that contains data.
Usage: Each <td> element represents a data point within a row. It can contain text, images, links, or other HTML elements. <td> elements are used within <tr> elements to create the body of the table.

5. <thead>

Purpose: The <thead> element is used to group the header content in a table. It is a semantic element that helps to define the header section of the table.
Usage: The <thead> element typically contains one or more <tr> elements, which in turn contain <th> elements. Using <thead> improves accessibility and allows for better styling and manipulation of the table header, especially when combined with <tbody> and <tfoot> for organizing the table structure.

• Question 2: What is the difference between colspanand rowspanin tables? Provide examples?

ans=> 1. colspan:

Definition: The colspan attribute specifies the number of columns a cell should span across. It allows a single cell to extend horizontally across multiple columns in a table.

Usage: This is useful when you want to create a header that covers several columns or when you want to combine data from multiple columns into a single cell.

Example of colspan:

```
<table border="1">
    <tr>
        <th colspan="3">Student Information</th>
    </tr>
    <tr>
        <th>Name</th>
        <th>Age</th>
```

```
        <th>City</th>
    </tr>
    <tr>
        <td>John Doe</td>
        <td>30</td>
        <td>New York</td>
    </tr>
    <tr>
        <td>Jane Smith</td>
        <td>25</td>
        <td>Los Angeles</td>
    </tr>
</table>
```

2. rowspan:

Definition: The rowspan attribute specifies the number of rows a cell should span vertically. It allows a single cell to extend across multiple rows in a table.

Usage: This is useful when you want to combine data from multiple rows into a single cell or when you want to create a hierarchical structure in the table.

Example of rowspan:

```
<table border="1">
    <tr>
        <th>Name</th>
        <th>Age</th>
        <th>City</th>
    </tr>
    <tr>
        <td rowspan="2">John Doe</td>
        <td>30</td>
        <td>New York</td>
    </tr>
    <tr>
        <td>31</td>
        <td>Los Angeles</td>
    </tr>
    <tr>
        <td>Jane Smith</td>
        <td>25</td>
        <td>Chicago</td>
    </tr>
</table>
```

• Question 3: Why should tables be used sparingly for layout purposes? What is a betteralternative?

Reasons to Avoid Tables for Layout

Semantic Meaning: Tables are designed to display tabular data, not for layout. Using tables for layout purposes can confuse the semantic meaning of the HTML, making it harder for search engines and assistive technologies (like screen readers) to interpret the content correctly.

Accessibility Issues:
Screen readers and other assistive technologies expect tables to contain data. If tables are used for layout, it can lead to a poor user experience for individuals with disabilities, as they may not be able to navigate the content effectively.

Maintenance Challenges:
Tables can become complex and difficult to maintain, especially as the layout changes. Modifying a table-based layout often requires significant changes to the HTML structure, making it less flexible and more prone to errors.

Responsive Design Limitations:
Tables are not inherently responsive. Adapting a table layout for different screen sizes can be challenging, often requiring additional CSS or JavaScript to achieve a fluid design. This can lead to a suboptimal user experience on mobile devices.

Performance Concerns:
Complex table structures can lead to larger HTML files, which may impact page load times. This is particularly relevant for users on slower connections.

Better Alternatives:
CSS for Layout:
The best alternative to using tables for layout is to use CSS (Cascading Style Sheets). CSS provides powerful layout techniques that allow for flexible and responsive designs. Some popular CSS layout methods include:

Flexbox:
A one-dimensional layout model that allows for easy alignment and distribution of space among items in a container, making it ideal for building responsive layouts.
CSS Grid:
A two-dimensional layout system that enables the creation of complex grid-based layouts with rows and columns, providing greater control over the design.
Float and Positioning:
While less common in modern design, floats and positioning can still be used for specific layout needs.
Semantic HTML Elements:
Use semantic HTML elements like <header>, <nav>, <main>, <section>, <article>, and <footer> to structure the content of a webpage. These elements provide meaning to the content and improve accessibility.

Responsive Design Frameworks:
Consider using responsive design frameworks like Bootstrap or Foundation, which provide pre-built CSS classes and components for creating responsive layouts without the need for tables.

Additional Requirements:

Q1 : • Use theadfor the table header.?

ANS=>

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Product Table Example</title>
    <style>
        table {
            width: 100%;
            border-collapse: collapse;
        }
        th, td {
            border: 1px solid #ddd;
            padding: 8px;
```

```
            text-align: left;
        }
        th {
            background-color: #f2f2f2;
        }
    </style>
</head>
<body>

    <h2>Product List</h2>
    <table>
        <thead>
            <tr>
                <th>Product Name</th>
                <th>Product Image</th>
                <th>Price ($)</th>
                <th>Description</th>
                <th>Availability</th>
            </tr>
        </thead>
        <tbody>
            <tr>
                <td>Product 1</td>
                <td><img src="https://via.placeholder.com/50" alt="Product 1 Image"></td>
                <td>19.99</td>
                <td>This is a description of Product 1.</td>
                <td>In Stock</td>
            </tr>
            <tr>
                <td>Product 2</td>
                <td><img src="https://via.placeholder.com/50" alt="Product 2 Image"></td>
                <td>29.99</td>
                <td>This is a description of Product 2.</td>
                <td>Out of Stock</td>
            </tr>
            <tr>
                <td>Product 3</td>
                <td><img src="https://via.placeholder.com/50" alt="Product 3 Image"></td>
                <td>39.99</td>
                <td>This is a description of Product 3.</td>
                <td>In Stock</td>
            </tr>
        </tbody>
    </table>

</body>
</html>
```

Q2 : Add a border and some basic styling using inline CSS.?

ANS=>

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Product Table Example</title>
</head>
<body>

    <h2 style="text-align: center; color: #333;">Product List</h2>
    <table style="width: 100%; border-collapse: collapse; margin: 20px 0;">
        <thead style="background-color: #f2f2f2;">
            <tr>
```

```
                <th style="border: 1px solid #ddd; padding: 8px; text-align:
left;">Product Name</th>
                <th style="border: 1px solid #ddd; padding: 8px; text-align:
left;">Product Image</th>
                <th style="border: 1px solid #ddd; padding: 8px; text-align: left;">Price
($)</th>
                <th style="border: 1px solid #ddd; padding: 8px; text-align:
left;">Description</th>
                <th style="border: 1px solid #ddd; padding: 8px; text-align:
left;">Availability</th>
            </tr>
        </thead>
        <tbody>
            <tr>
                <td style="border: 1px solid #ddd; padding: 8px;">Product 1</td>
                <td style="border: 1px solid #ddd; padding: 8px;"><img
src="https://via.placeholder.com/50" alt="Product 1 Image"></td>
                <td style="border: 1px solid #ddd; padding: 8px;">19.99</td>
                <td style="border: 1px solid #ddd; padding: 8px;">This is a description
of Product 1.</td>
                <td style="border: 1px solid #ddd; padding: 8px;">In Stock</td>
            </tr>
            <tr>
                <td style="border: 1px solid #ddd; padding: 8px;">Product 2</td>
                <td style="border: 1px solid #ddd; padding: 8px;"><img
src="https://via.placeholder.com/50" alt="Product 2 Image"></td>
                <td style="border: 1px solid #ddd; padding: 8px;">29.99</td>
                <td style="border: 1px solid #ddd; padding: 8px;">This is a description
of Product 2.</td>
                <td style="border: 1px solid #ddd; padding: 8px;">Out of Stock</td>
            </tr>
            <tr>
                <td style="border: 1px solid #ddd; padding: 8px;">Product 3</td>
                <td style="border: 1px solid #ddd; padding: 8px;"><img
src="https://via.placeholder.com/50" alt="Product 3 Image"></td>
                <td style="border: 1px solid #ddd; padding: 8px;">39.99</td>
                <td style="border: 1px solid #ddd; padding: 8px;">This is a description
of Product 3.</td>
                <td style="border: 1px solid #ddd; padding: 8px;">In Stock</td>
            </tr>
        </tbody>
    </table>

</body>
</html>
```

Q3 :Use colspanor rowspanto merge cells where applicable. ?

ANS =>

```
 <!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Product Table Example with Merged Cells</title>
</head>
<body>

    <h2 style="text-align: center; color: #333;">Product List</h2>
    <table style="width: 100%; border-collapse: collapse; margin: 20px 0;">
        <thead style="background-color: #f2f2f2;">
            <tr>
                <th style="border: 1px solid #ddd; padding: 8px; text-align:
left;">Product Name</th>
```

```
                <th style="border: 1px solid #ddd; padding: 8px; text-align:
left;">Product Image</th>
                <th style="border: 1px solid #ddd; padding: 8px; text-align: left;">Price
($)</th>
                <th style="border: 1px solid #ddd; padding: 8px; text-align:
left;">Description</th>
                <th style="border: 1px solid #ddd; padding: 8px; text-align:
left;">Availability</th>
            </tr>
        </thead>
        <tbody>
            <tr>
                <td style="border: 1px solid #ddd; padding: 8px;">Product 1</td>
                <td style="border: 1px solid #ddd; padding: 8px;"><img
src="https://via.placeholder.com/50" alt="Product 1 Image"></td>
                <td style="border: 1px solid #ddd; padding: 8px;">19.99</td>
                <td style="border: 1px solid #ddd; padding: 8px;" colspan="2">This is a
description of Product 1 that spans two columns.</td>
            </tr>
            <tr>
                <td style="border: 1px solid #ddd; padding: 8px;">Product 2</td>
                <td style="border: 1px solid #ddd; padding: 8px;"><img
src="https://via.placeholder.com/50" alt="Product 2 Image"></td>
                <td style="border: 1px solid #ddd; padding: 8px;">29.99</td>
                <td style="border: 1px solid #ddd; padding: 8px;">This is a description
of Product 2.</td>
                <td style="border: 1px solid #ddd; padding: 8px;">Out of Stock</td>
            </tr>
            <tr>
                <td style="border: 1px solid #ddd; padding: 8px;" rowspan="2">Product
3</td>
                <td style="border: 1px solid #ddd; padding: 8px;" rowspan="2"><img
src="https://via.placeholder.com/50" alt="Product 3 Image"></td>
                <td style="border: 1px solid #ddd; padding: 8px;">39.99</td>
                <td style="border: 1px solid #ddd; padding: 8px;">This is a description
of Product 3.</td>
                <td style="border: 1px solid #ddd; padding: 8px;">In Stock</td>
            </tr>
            <tr>
                <td style="border: 1px solid #ddd; padding: 8px;">Discounted Price</td>
                <td style="border: 1px solid #ddd; padding: 8px;">34.99</td>
            </tr>
        </tbody>
    </table>

</body>
</html>
```

COMPLETE HTML ASSIGNMENT