

HTML & CSS

PART-2

HTML 5.0: Components of HTML, Text Formatting tags, Quotations, Links, Images, CSS: syntax, Box Model, CSS outline, Links in CSS, Responsiveness, Position Property, Navigation Bars, Dropdown, Forms.

Introduction to HTML5

HTML (HyperText Markup Language) is the standard language used to create and structure web pages.

- ❖ **HTML5** is the **latest version** of HTML, introduced in 2014, with new features for modern web development.
- ❖ It provides:
 - Better support for **multimedia** (audio, video, canvas for graphics).
 - New **semantic elements** for meaningful structure.
 - Improved **form controls** and input types.
 - **APIs** for offline storage, geolocation, drag-and-drop, etc.

Basic Components of HTML5

1. Document Type Declaration

- Tells the browser which version of HTML the page is written in.

```
<!DOCTYPE html>
```

2. HTML Document Structure

- Every HTML5 page follows this structure:

```
<!DOCTYPE html>
<html>
<head>
  <title>My First HTML5 Page</title>
</head>
<body>
  <h1>Welcome to VJIT!</h1>
</body>
</html>
```

Output:**3. Head Section (<head>)**

- Contains metadata about the document (not displayed on the page).
- Includes:
 - <title> → Page title
 - <meta> → Information like charset, description
 - <link> → External CSS files
 - <script> → External JavaScript files

Example:

```
<head>
  <title>HTML5 Example</title>
  <meta charset="UTF-8">
  <meta name="description" content="Introduction to
HTML5">
  <link rel="stylesheet" href="style.css">
</head>
```

4. Body Section (<body>)

- The **visible part** of the web page.
- Contains **text, images, links, tables, forms, multimedia**.

Example:

```
<body>
  <h1>Welcome to HTML5</h1>
  <p>This is a paragraph.</p>
  
</body>
```

5. Semantic Elements

- New in HTML5 → Add meaning to web pages.
- Examples:
 - <header> → Page or section header
 - <nav> → Navigation menu
 - <article> → Independent content
 - <section> → Content section
 - <footer> → Page or section footer

Example:

```
<header>
  <h1>My Website</h1>
</header>
<nav>
  <a href="home.html">Home</a> |
  <a href="about.html">About</a>
</nav>
<section>
  <article>
    <h2>Introduction</h2>
    <p>This is an article inside a section.</p>
  </article>
</section>
<footer>
  <p>© 2025 My Website</p>
</footer>
```

6. Multimedia Support

- HTML5 supports **video and audio** without plugins.

```
<video width="320" height="240" controls>
  <source src="movie.mp4" type="video/mp4">
</video>
<audio controls>
  <source src="song.mp3" type="audio/mpeg">
</audio>
```

7. Forms and Input Types

- HTML5 introduces new **input types** like email, date, range, color, etc.

```
<form>
  Name: <input type="text"><br>
  Email: <input type="email"><br>
  Birthdate: <input type="date"><br>
  Favorite Color: <input type="color"><br>
  <input type="submit" value="Submit">
</form>
```

Text Formatting Tags in HTML5

Text formatting tags are used to **style, highlight or emphasize text** in a web page.

1. Bold Text

- `` → Bold text (no extra meaning, only styling).
- `` → Important text (semantic meaning + bold style).

```
<p>This is <b>bold</b> text.</p>
<p>This is <strong>important</strong> text.</p>
```

2. Italic / Emphasized Text

- `<i>` → Italic text (styling only).
- `` → Emphasized text (semantic meaning, usually italic).

```
<p>This is <i>italic</i> text.</p>
<p>This is <em>emphasized</em> text.</p>
```

3. Underline & Inserted Text

- `<u>` → Underlined text.
- `<ins>` → Inserted text (underlined, semantic).

```
<p>This is <u>underlined</u> text.</p>
<p>This is <ins>newly inserted</ins> text.</p>
```

4. Deleted / Strikethrough Text

- `` → Deleted text (semantic).
- `<s>` → Strikethrough text (styling only).

```
<p>This is <del>deleted</del> text.</p>
<p>This is <s>striketrough</s> text.</p>
```

5. Superscript and Subscript

- `<sup>` → Superscript (above baseline).
- `<sub>` → Subscript (below baseline).

```
<p>Formula: H<sub>2</sub>O</p>
<p>Math: x<sup>2</sup> + y<sup>2</sup></p>
```

6. Highlighting

- `<mark>` → Highlights text with a yellow background.

```
<p>This is <mark>highlighted</mark> text.</p>
```

7. Small Text

- `<small>` → Renders smaller text.

```
<p>This is <small>small</small> text.</p>
```

8. Monospace / Code

- `<code>` → Inline code (monospace font).
- `<pre>` → Preformatted text (preserves spaces & line breaks).

```
<p>Use <code>printf()</code> in C.</p>
<pre>
Line 1
  Line 2 (Indented)
Line 3
</pre>
```

9. Quotation and Citation

- `<q>` → Short quotation (inline, adds quotes).
- `<blockquote>` → Long quotation (block-level).
- `<cite>` → Citation (usually italic).

```
<p>He said, <q>Hello World!</q></p>
<blockquote>
  "This is a blockquote for long quotations."
</blockquote>
<p><cite>— Albert Einstein</cite></p>
```

Example:

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>HTML Text Formatting Tags</title>
</head>
<body>
  <h1>HTML Text Formatting Tags</h1>

  <!-- 1. Bold Text -->
  <h2>1. Bold Text</h2>
  <p>This is <b>bold</b> text.</p>
  <p>This is <strong>important</strong> text.</p>

  <!-- 2. Italic / Emphasized Text -->
  <h2>2. Italic / Emphasized Text</h2>
  <p>This is <i>italic</i> text.</p>
  <p>This is <em>emphasized</em> text.</p>

  <!-- 3. Underline & Inserted Text -->
  <h2>3. Underline & Inserted Text</h2>
  <p>This is <u>underlined</u> text.</p>
  <p>This is <ins>newly inserted</ins> text.</p>
```

<!-- 4. Deleted / Strikethrough Text -->

<h2>4. Deleted / Strikethrough Text</h2>

<p>This is deleted text.</p>

<p>This is <s>strikethrough</s> text.</p>

<!-- 5. Superscript and Subscript -->

<h2>5. Superscript and Subscript</h2>

<p>Formula: H₂O</p>

<p>Math: x² + y²</p>

<!-- 6. Highlighting -->

<h2>6. Highlighting</h2>

<p>This is <mark>highlighted</mark> text.</p>

<!-- 7. Small Text -->

<h2>7. Small Text</h2>

<p>This is <small>small</small> text.</p>

<!-- 8. Monospace / Code -->

<h2>8. Monospace / Code</h2>

<p>Use <code>printf()</code> in C.</p>

<pre>

Line 1

Line 2 (Indented)

Line 3

</pre>

<!-- 9. Quotation and Citation -->

<h2>9. Quotation and Citation</h2>

<p>He said, <q>Hello World!</q></p>

<blockquote>

"This is a blockquote for long quotations."

</blockquote>

<p><cite>— Albert Einstein</cite></p>

</body>

</html>

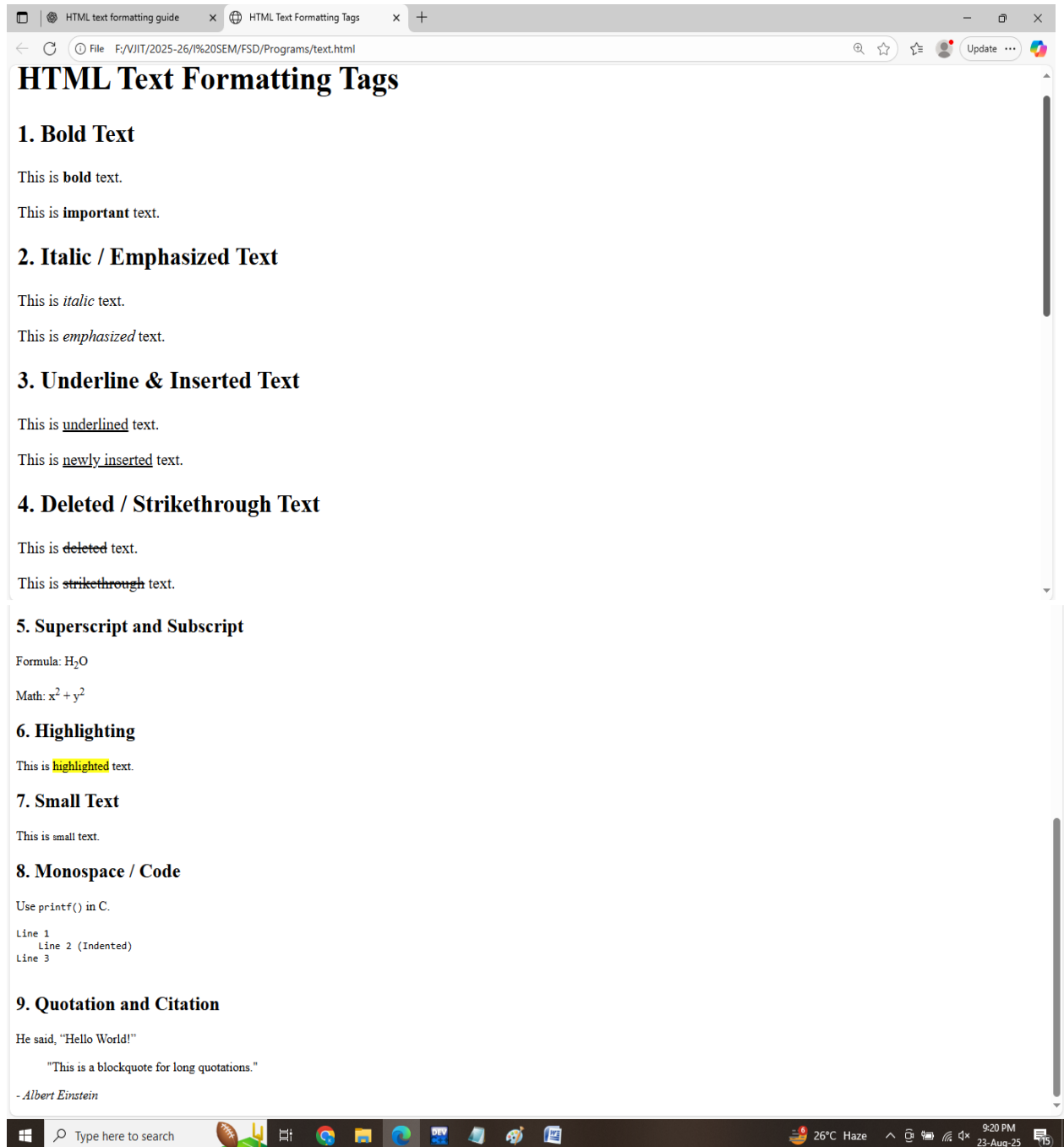
Output:

Table of Text Formatting Tags

Tag	Meaning / Use	Example Output
	Bold (style only)	Bold
	Important (semantic, bold)	Important
<i>	Italic (style only)	<i>Italic</i>
	Emphasis (semantic, italic)	<i>Emphasized</i>
<u>	Underline	<u>Underlined</u>
<ins>	Inserted (underlined, semantic)	<u>Inserted</u>
	Deleted (semantic)	Deleted
<s>	Strikethrough (style only)	Striked
<sup>	Superscript	x ²
<sub>	Subscript	H ₂ O
<mark>	Highlight	Marked
<small>	Smaller text	small
<code>	Inline code (monospace)	printf()
<pre>	Preformatted text	(Keeps spaces & lines)
<q>	Short quotation (inline)	“quoted”
<blockquote>	Long quotation (block)	Block Quote
<cite>	Citation / Reference	<i>Citation</i>

Quotation Tags in HTML5

Quotations are used to represent text taken from another source or spoken by someone.

1. <q> – Short Quotation

- Inline quotation.
- Browser automatically adds **quotation marks** (" ") around the text.

Example:

```
<p>Einstein once said, <q>Imagination is more important than knowledge.</q></p>
```

Output:

Einstein once said, "*Imagination is more important than knowledge.*"

2. <blockquote> – Long Quotation

- Block-level element.
- Used for long passages.
- Usually indents the quoted text.

Example:

```
<blockquote>
  "The greatest glory in living lies not in never falling,
  but in rising every time we fall."
</blockquote>
```

Output:

"The greatest glory in living lies not in never falling, but in rising every time we fall."

3. <cite> – Citation / Source

- Defines the **title of a creative work** (book, article, movie, etc.).
- Usually displayed in *italic*.

Example:

```
<p><cite>The Theory of Relativity</cite> by Albert Einstein.</p>
```

Output:

The Theory of Relativity by Albert Einstein.

4. <abbr> (Optional for quoted work with abbreviation)

Sometimes abbreviations are used inside quotations.

- <abbr> gives the **full form** when hovered.

Example:

```
<p>The founder of <abbr title="World Wide Web">WWW</abbr> said,  
<q>The web should be for everyone.</q></p>
```

Summary of Quotation Tags

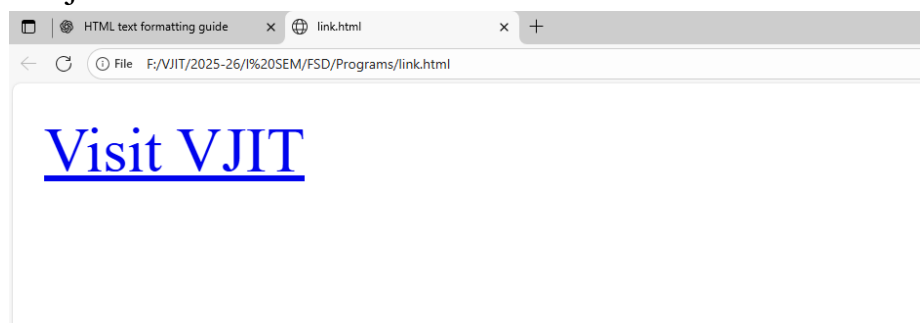
Tag	Purpose	Example
<q>	Short inline quotation	"Hello"
<blockquote>	Long block quotation	Indented block
<cite>	Source / Title	<i>Book Name</i>
<abbr>	Abbreviation in quote	WWW → World Wide Web

Links in HTML5

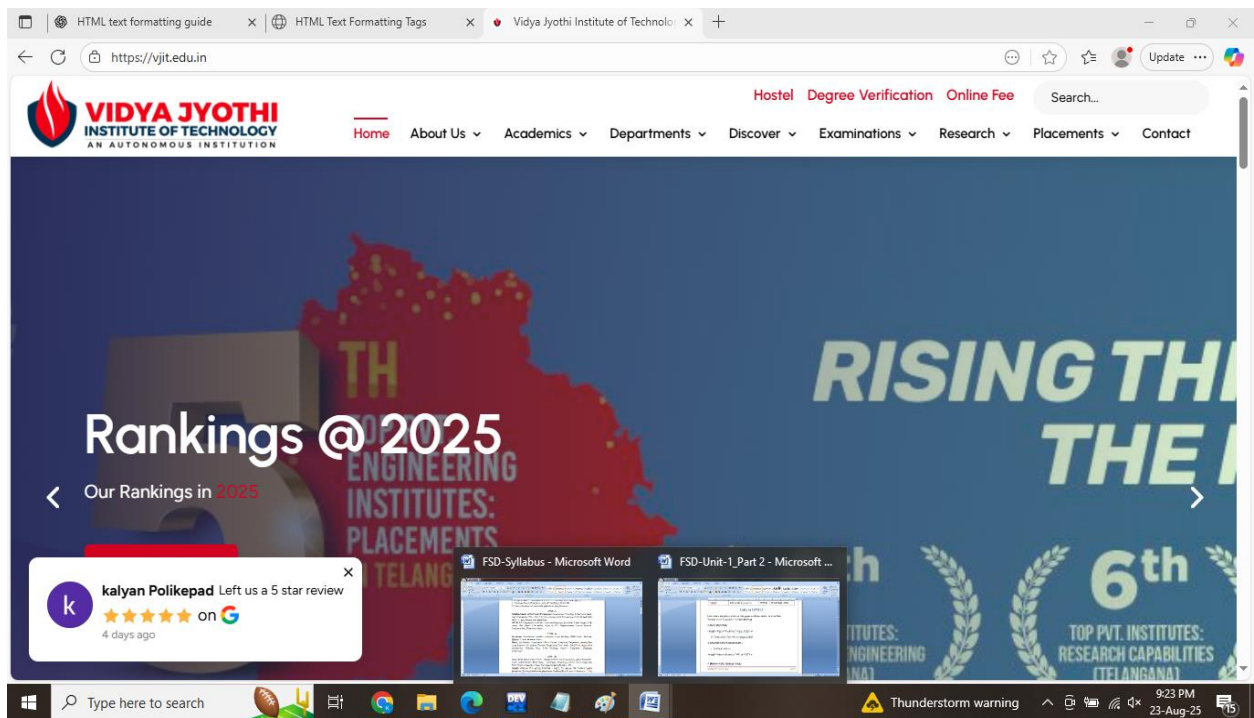
Links allow navigation between web pages, sections, emails, or even files. They are created using the <a> (anchor) tag.

1. Basic Hyperlink

```
<a href="https://vjit.edu.in/">Visit VJIT</a>
```



- Opens <https://vjit.edu.in/> when clicked.



2. Absolute URL (External Link)

- Full web address.

`Go to VJIT`

3. Relative URL (Internal Link)

- Links to pages **within the same website/project**.

`CSE(DS)`

4. Open Link in New Tab

- Add `target="_blank"`.

`Open CSE(DS) in new tab`

5. Email Link

- Opens the user's email app with pre-filled address.

`Email Us`

6. Telephone Link

- Used on mobile to directly dial.

```
<a href="tel:+917373 637637">Call Us</a>
```

7. Link to a Section (Bookmark / Anchor Link)

- Use id to mark a section, then link to it.

```
<a href="#contact">Go to Contact</a>
```

```
<h2 id="contact">Contact Us</h2>
```

```
<p>Here are our contact details...</p>
```

8. Download Link

- Force download instead of opening.

```
<a href="files/report.pdf" download>Download Report</a>
```

Summary of Link Types

Link Type	Example Code
External website	<code>Link</code>
Internal page	<code>About</code>
New tab	<code>Open</code>
Email	<code>Email</code>
Telephone	<code>Call</code>
Bookmark / Section link	<code>Jump</code>
File download	<code>Download</code>

Images in HTML5

Images make a webpage more attractive and informative.

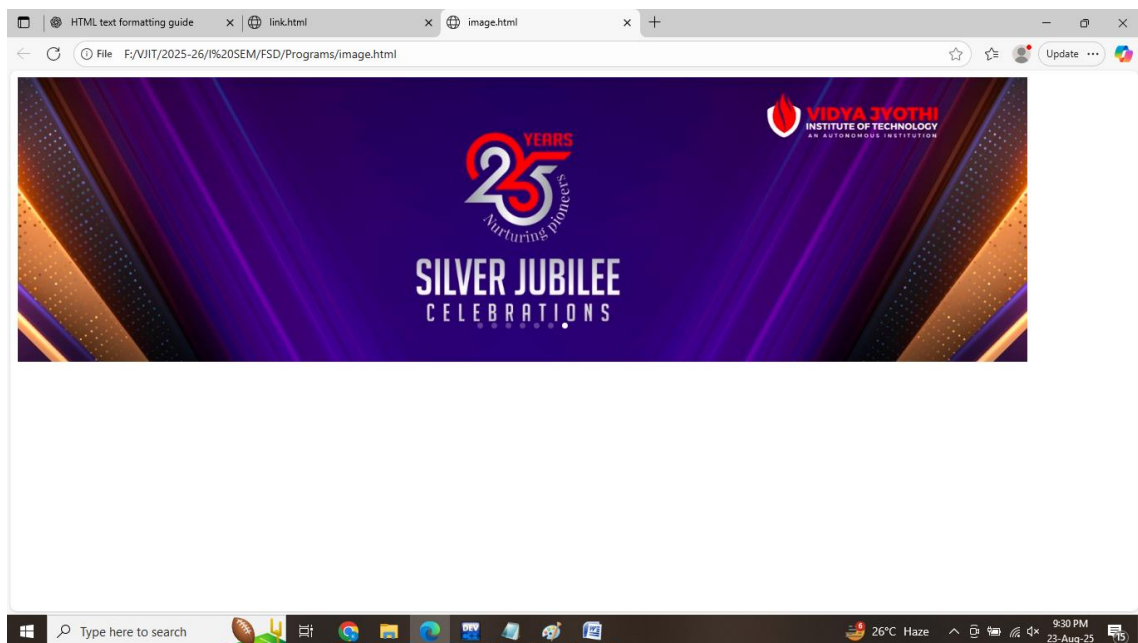
- ❖ They are inserted using the ** tag**, which is a **self-closing tag** (does not need ``).

1. Basic Syntax

```

```

- **src** → path of the image file
- **alt** → alternative text (important for accessibility & SEO, shows if image fails to load)



2. Image from Local Folder

```

```

3. Image from Internet (Absolute URL)

```

```

4. Resize Image (Width & Height)

```

```

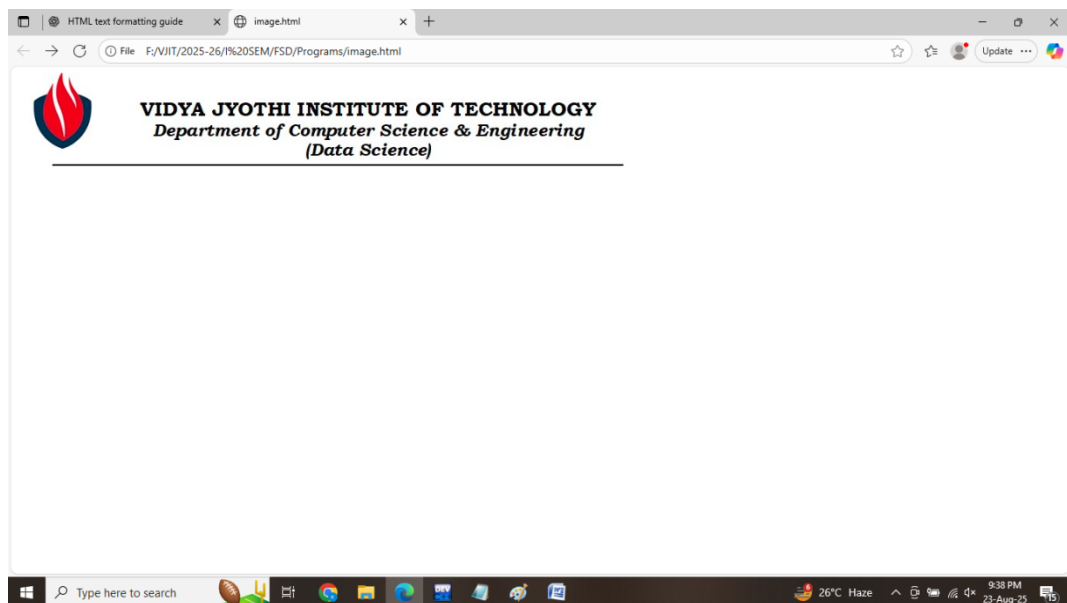
5. Image with Border & Styling (using CSS)

```

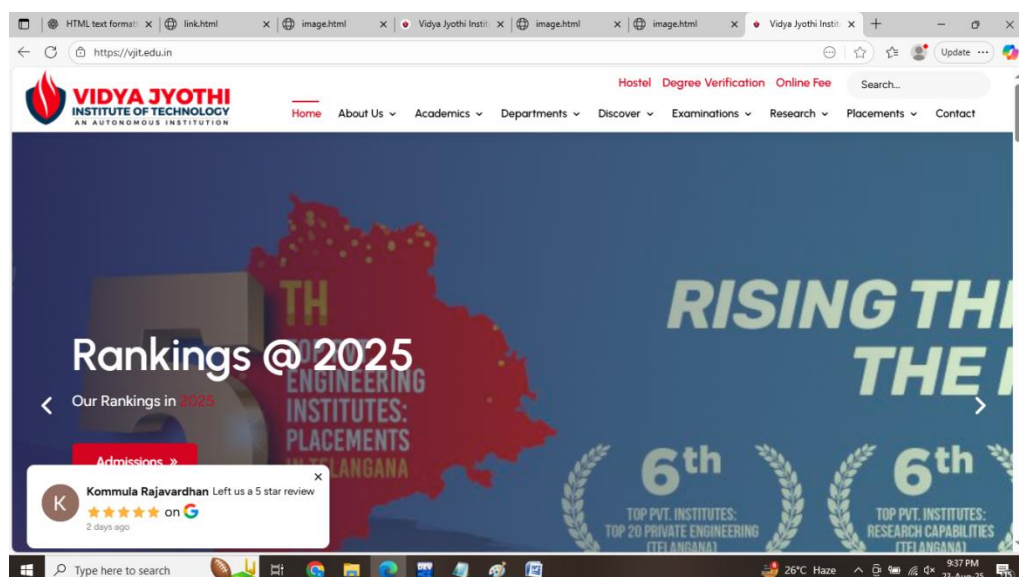
```

6. Image as a Link

```
<a href="https://www.wikipedia.org">  
  
</a>
```



Clicking the **image** opens the link.



7. Responsive Image (Fit Screen Size)

```

```

8. Image with Tooltip (Title Attribute)

```

```

Summary of Tag Attributes

Attribute	Purpose
src	Image source (file path or URL)
alt	Alternative text (SEO & accessibility)
width	Width of the image
height	Height of the image
title	Tooltip text when hovering
style	Inline CSS styling
loading="lazy"	Loads image only when needed (improves performance)

CSS – Cascading Style Sheets

CSS (Cascading Style Sheets) is used to **style and format HTML documents**.

- ❖ It controls **layout, colors, fonts, spacing, borders, backgrounds, responsiveness**, and more.
- ❖ While **HTML provides structure** (headings, paragraphs, images), **CSS provides design** (look & feel).

Why CSS?

- Separates **content (HTML)** from **presentation (CSS)**
- Makes websites **beautiful, consistent, and responsive**
- Saves time: one CSS file can style multiple pages
- Allows **themes and reusable styles**

Types of CSS

There are **three main ways** to use CSS:

1. Inline CSS
2. Internal CSS
3. External CSS

1. Inline CSS

Applied **inside an HTML tag** using the style attribute.

```
<p style="color:blue; font-size:20px;">This is blue text.</p>
```

✓ Useful for **quick styling**

✗ Not recommended for large projects (hard to maintain)

2. Internal CSS

Written inside a <style> tag in the <head> section.

```
<!DOCTYPE html>
<html>
<head>
  <style>
    body { background-color: lightyellow; }
    h1 { color: green; text-align: center; }
    p { font-size: 18px; }
  </style>
</head>
<body>
  <h1>Welcome to CSS</h1>
  <p>This is styled with internal CSS.</p>
</body>
</html>
```

✓ Good for **single-page styling**

3. External CSS

Written in a separate **.css file** and linked using **<link>**.

<pre><!-- index.html --> <!DOCTYPE html> <html> <head> <link rel="stylesheet" href="styles.css"> </head> <body> <h1>Hello CSS</h1> <p>This text is styled using an external file.</p> </body> </html></pre>
<pre>/* styles.css */ body { background-color: beige; } h1 { color: darkblue; font-family: Arial; } p { font-size: 16px; color: brown; }</pre>

✓ Best for **large projects**

✓ Reusable and maintainable

CSS Syntax

CSS (Cascading Style Sheets) is used to style HTML elements (colors, fonts, layout, spacing, etc.).

❖ The **basic CSS syntax** looks like this:

<pre>selector { property: value; }</pre>
--

Example:

```
p {
  color: red;
  font-size: 18px;
}
```

- **Selector** → chooses HTML element (p)
- **Property** → style feature (color)
- **Value** → how it looks (red)

Selectors in CSS

Selector	Example	Meaning
Universal	* { margin:0; }	Selects all elements
Element	p { color:blue; }	Selects all <p>
ID	#main { color:red; }	Selects element with id="main"
Class	.highlight { background:yellow; }	Selects elements with class="highlight"
Grouping	h1, h2 { color:green; }	Styles multiple elements
Descendant	div p { font-size:14px; }	Selects <p> inside <div>

CSS Properties**❖ Colors & Backgrounds**

```
body { background-color: lightblue; }
h1 { color: navy; }
```

❖ Fonts & Text

```
p { font-family: Arial, sans-serif; font-size: 18px; text-align: justify; }
```

❖ Box Model (Margin, Border, Padding, Content)

```
div {
  width: 200px;
  padding: 20px;
  border: 2px solid black;
  margin: 10px;
}
```

❖ Responsive Design

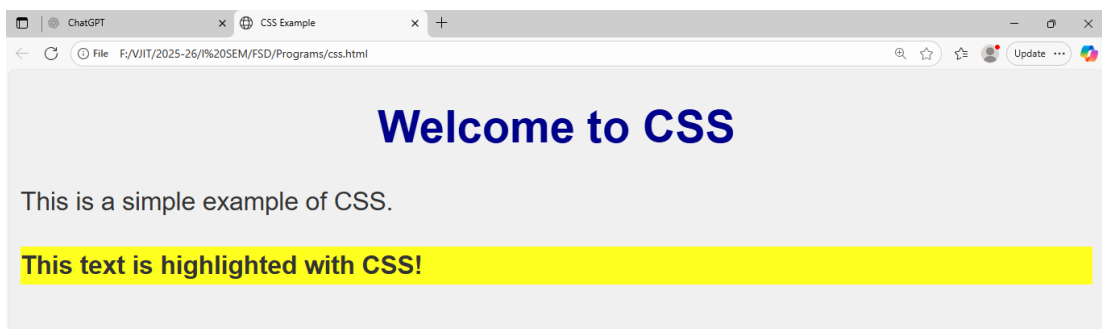
```
@media screen and (max-width: 600px) {  
  body { background-color: lightgreen; }  
}
```

Advantages of CSS

1. **Separation of content & style** → Cleaner code
2. **Consistency** → One change reflects on all pages
3. **Improved performance** → Faster page loads with external CSS
4. **Better design** → Fonts, colors, layouts
5. **Responsive** → Adapts to devices (mobile, tablet, desktop)

Example

```
<!DOCTYPE html>  
<html>  
<head>  
  <title>CSS Example</title>  
  <style>  
    body { background-color: #f0f0f0; font-family: Arial; }  
    h1 { color: darkblue; text-align: center; }  
    p { color: #333; font-size: 18px; line-height: 1.5; }  
    .highlight { background: yellow; font-weight: bold; }  
  </style>  
</head>  
<body>  
  <h1>Welcome to CSS</h1>  
  <p>This is a simple example of CSS.</p>  
  <p class="highlight">This text is highlighted with CSS!</p>  
</body>  
</html>
```



CSS Box Model

The **CSS Box Model** is the fundamental concept that defines how elements are displayed and how space is calculated around them in a web page. Every HTML element is considered a **box**, and the box model describes the structure of that box.

It consists of **four main components** (from inside to outside):

1. Content

- The actual content of the box (text, images, or other elements).
- Controlled with properties like width, height.

2. Padding

- The transparent area between the content and the border.
- Increases the clickable/visible area around content without affecting other elements.
- Controlled with padding, padding-top, padding-right, etc.

3. Border

- Surrounds the padding (if any) and content.
- Visible styling such as line thickness, style, and color.
- Controlled with border, border-width, border-style, border-color.

4. Margin

- The outermost layer; space between the element's border and the neighboring elements.
- Creates separation between elements.
- Controlled with margin, margin-top, margin-right, etc.

Formula for Total Space Taken by an Element:

$$\text{Total Width} = \text{content-width} + \text{padding-left} + \text{padding-right} + \text{border-left} + \text{border-right} + \text{margin-left} + \text{margin-right}$$
$$\text{Total Height} = \text{content-height} + \text{padding-top} + \text{padding-bottom} + \text{border-top} + \text{border-bottom} + \text{margin-top} + \text{margin-bottom}$$

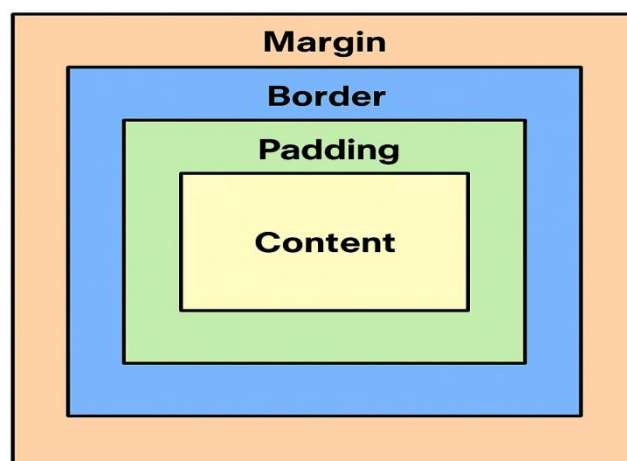
Example in CSS:

```
<!DOCTYPE html>
<html>
<head>
<style>
.box {
  width: 200px;
  height: 100px;
  padding: 20px;
  border: 5px solid blue;
  margin: 15px;
  background-color: lightyellow;
}
</style>
</head>
<body>
<div class="box">
  This is a box model example!
</div>
</body>
</html>
```

In this example:

- Content = 200×100 px
- Padding = 20 px
- Border = 5 px
- Margin = 15 px

So, **Total width** = 200 + 20+20 + 5+5 + 15+15 = 280 px.



CSS Outline

In **CSS**, an **outline** is a line drawn around an element, outside the border, to make it stand out. Unlike borders, outlines do not take up space and do not affect the element's size or layout.

Key Properties of CSS Outline:

1. **outline-style** → Defines the style of the outline.

Example: solid, dashed, dotted, double, groove, none.

```
p {  
  outline-style: solid;  
}
```

2. **outline-color** → Sets the color of the outline.

Example: red, blue, #00ff00, rgb(0,0,255).

```
p {  
  outline-color: red;  
}
```

3. **outline-width** → Sets the thickness of the outline.

Values: thin, medium, thick, or in px (e.g., 5px).

```
p {  
  outline-width: thick;  
}
```

4. **outline (shorthand)** → Combines width, style, and color.

```
p {  
  outline: 3px dashed blue;  
}
```

5. **outline-offset** → Adds space between the outline and the border.

```
p {  
  outline: 2px solid green;  
  outline-offset: 5px;  
}
```

Example:

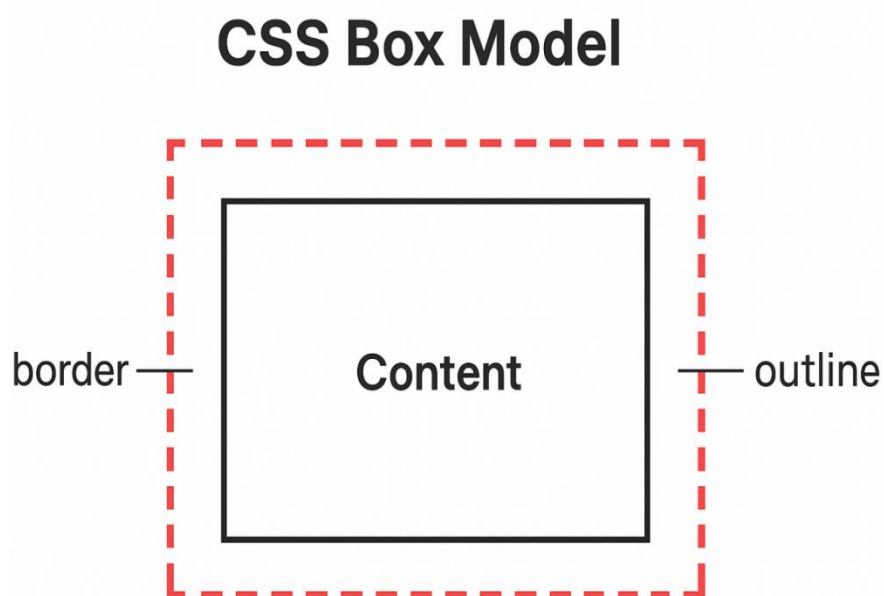
```
<!DOCTYPE html>
<html>
<head>
<style>
div {
  border: 2px solid black;
  padding: 20px;
  outline: 5px dashed red;
  outline-offset: 10px;
}
</style>
</head>
<body>

<h2>CSS Outline Example</h2>
<div>This box has a border and an outline.</div>

</body>
</html>
```

In this example:

- The **border** is inside.
- The **outline** is outside, dashed red, with spacing.



Links in CSS

In CSS, **links** are styled to control how they appear in different states (normal, hovered, visited, active, etc.).

- ❖ By default, HTML links (<a> tags) are underlined and blue, but CSS lets you customize them.

Link States in CSS

1. **a:link** → Normal, unvisited link.
2. **a:visited** → Link that has been visited.
3. **a:hover** → When the mouse pointer is over the link.
4. **a:active** → When the link is clicked.

Example

```
<!DOCTYPE html>
<html>
<head>
<style>
  /* Normal link */
  a:link {
    color: blue;
    text-decoration: none;
  }

  /* Visited link */
  a:visited {
    color: purple;
  }

  /* Hover effect */
  a:hover {
    color: red;
    text-decoration: underline;
  }
```

```
/* Active (while clicking) */
a:active {
  color: green;
}
</style>
</head>
<body>
  <h2>CSS Link Example</h2>
  <p><a href="https://www.google.com" target="_blank">Visit
Google</a></p>
</body>
</html>
```

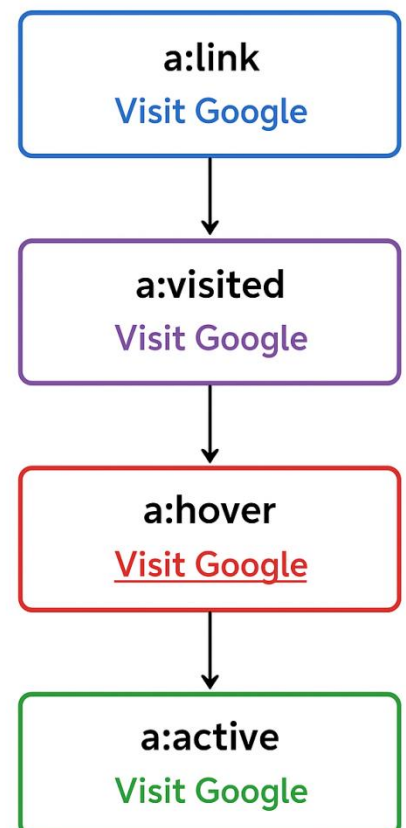
Output Explanation

- When the link has **not been visited**, it will be **blue**.
- After clicking, the link becomes **purple** (visited).
- When hovered, it turns **red** and underlined.
- When actively clicked, it appears **green**.

The **order of states in CSS**:

a:link → **a:visited** → **a:hover** → **a:active**

CSS Link States



Responsiveness in Web Design (CSS & HTML5)

What is Responsiveness?

Responsiveness means that a website automatically **adapts its layout and design** to different screen sizes and devices (desktop, tablet, mobile). It ensures good readability, usability, and user experience across all devices.

Key Features of Responsiveness:

1. **Fluid Grids** – Using relative units (% , em, rem, vh, vw) instead of fixed pixels.
2. **Flexible Images** – Images resize automatically (max-width: 100%).
3. **Media Queries** – CSS rules applied based on screen size or device type.
4. **Viewport Meta Tag** – Tells the browser how to scale the page.
`<meta name="viewport" content="width=device-width, initial-scale=1.0">`
5. **Responsive Frameworks** – Bootstrap, Tailwind CSS, Foundation, etc.

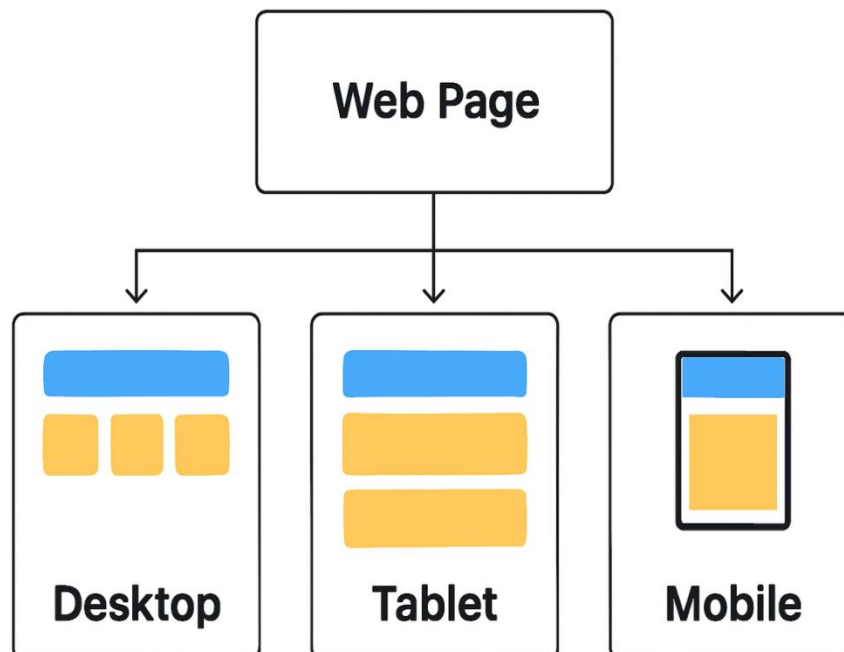
Example of a Responsive Web Page

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Responsive Example</title>
<style>
  body {
    font-family: Arial, sans-serif;
    margin: 0;
    padding: 0;
  }
  .container {
    display: flex;
    flex-wrap: wrap;
  }
  .box {
    flex: 1;
    min-width: 200px;
    margin: 10px;
    padding: 20px;
    background-color: lightblue;
    text-align: center;
  }
```

```
/* Media Query for smaller screens */
@media (max-width: 600px) {
  .box {
    flex: 100%;
  }
}
</style>
</head>
<body>
  <h2 style="text-align:center;">Responsive Boxes</h2>
  <div class="container">
    <div class="box">Box 1</div>
    <div class="box">Box 2</div>
    <div class="box">Box 3</div>
  </div>
</body>
</html>
```

How It Works

- On **large screens** → boxes are side by side.
- On **small screens (mobile)** → boxes stack vertically.



CSS Position Property

The **CSS position property** is used to control the positioning of elements on a webpage.

- ❖ It defines how an element is placed in the document flow and how it interacts with other elements.

Types of Position Values in CSS

1. **static (default)**

- Elements are positioned normally in the document flow.
- No special positioning is applied.

```
div {  
  position: static;  
}
```

2. **relative**

- Element is positioned **relative to its normal position**.
- You can shift it using top, right, bottom, left.

```
div {  
  position: relative;  
  top: 20px;  
  left: 30px;  
}
```

3. **absolute**

- Element is positioned **relative to the nearest positioned ancestor** (not static).
- Removed from normal document flow.

```
div {  
  position: absolute;  
  top: 50px;  
  right: 20px;  
}
```

4. **fixed**

- Element is positioned relative to the **viewport**.
- It stays fixed in place even when scrolling.

```
div {  
  position: fixed;  
  bottom: 10px;  
  right: 10px;  
}
```

5. **sticky**

- Acts like relative until a certain scroll point, then becomes fixed.
- Useful for sticky headers or navigation bars.

```
div {  
  position: sticky;  
  top: 0;  
}
```

Example: Different Positions

```
<!DOCTYPE html>  
<html>  
<head>  
<style>  
.box {  
  width: 150px;  
  height: 100px;  
  background: lightblue;  
  margin: 10px;  
  padding: 10px;  
  border: 2px solid navy;  
}  
.relative { position: relative; top: 20px; left: 30px; }  
.absolute { position: absolute; top: 50px; left: 50px; background: orange; }  
.fixed { position: fixed; bottom: 10px; right: 10px; background: lightgreen; }  
.sticky { position: sticky; top: 0; background: pink; }  
</style>
```

```
</head>
```

```
<body>
```

```
<div class="box static">Static</div>
```

```
<div class="box relative">Relative</div>
```

```
<div class="box absolute">Absolute</div>
```

```
<div class="box sticky">Sticky (scroll to see effect)</div>
```

```
<div class="box fixed">Fixed</div>
```

```
<p style="height:1000px;">Scroll down to see sticky and fixed positions in action.</p>
```

```
</body>
```

```
</html>
```

position

Static

Sticky

top: 0;

Relative

top: 20px;
left: 30px;

Absolute

top: 50px;
left: 50px;

Fixed

bottom: 10px;
right: 10px;

Navigation Bars

A **Navigation Bar (Navbar)** is a user interface element that provides links to different pages or sections of a website.

- ❖ It acts like a **map** for your site so users can quickly move around.

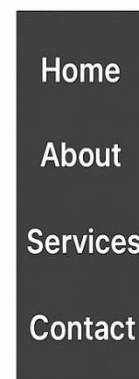
Navigation Bars

Horizontal Navbar



Links are placed side by side.

Vertical Navbar



Links are placed one below another.

Types of Navbars

1. Horizontal Navbar

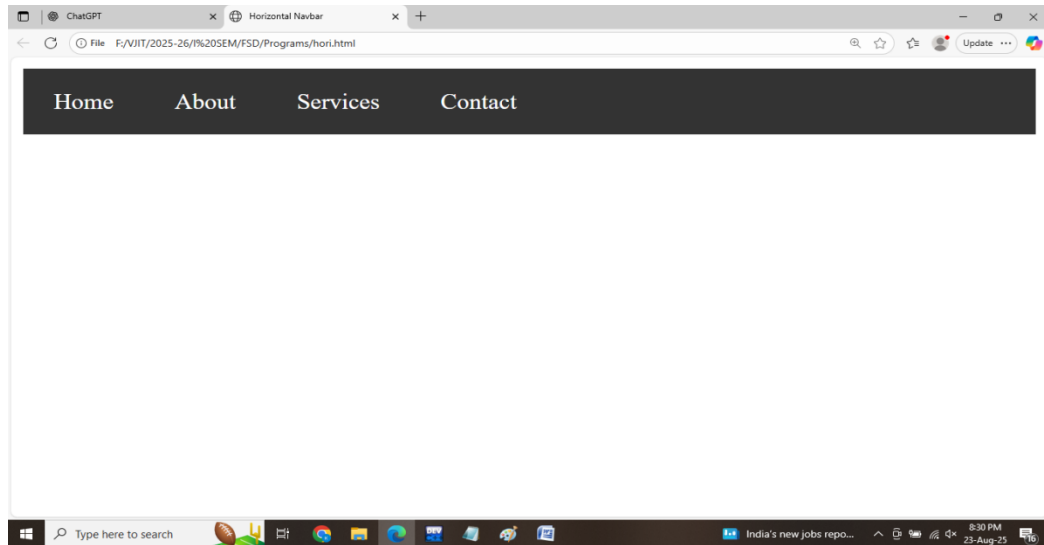
- Links are placed side by side (float: left; or display: inline-block;).
- Usually found **at the top** of a website.
- Common for **headers/menus**.

2. Vertical Navbar

- Links are stacked **one below the other**.
- Usually found on the **left/right side** of a page (sidebar).
- Useful for **dashboards or admin panels**.

Example : Horizontal Navbar

```
<!DOCTYPE html>
<html>
<head>
  <title>Horizontal Navbar</title>
  <style>
    ul {
      list-style-type: none;
      margin: 0;
      padding: 0;
      background-color: #333;
      overflow: hidden;
    }
    li {
      float: left;
    }
    li a {
      display: block;
      color: white;
      text-align: center;
      padding: 14px 20px;
      text-decoration: none;
    }
    li a:hover {
      background-color: #111;
    }
  </style>
</head>
<body>
  <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>
</body>
</html>
```

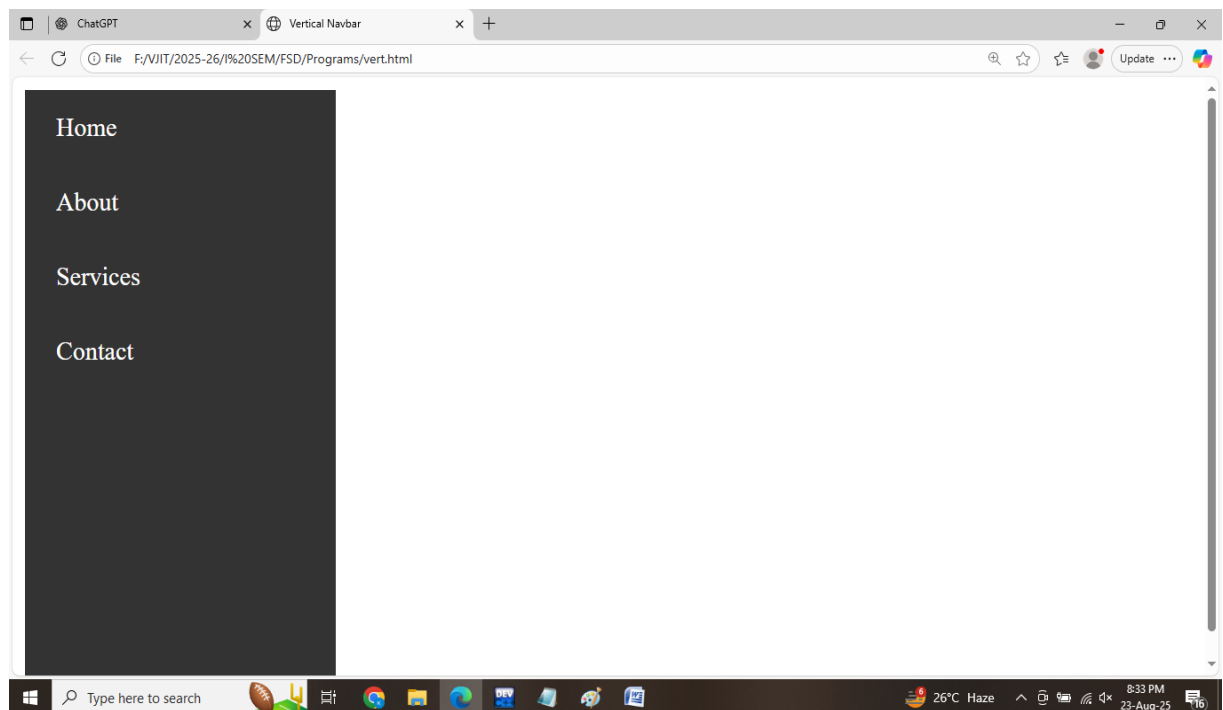
Output:

✓ **Use Case:** Top navigation bar for a company website.

Example : Vertical Navbar

```
<!DOCTYPE html>
<html>
<head>
  <title>Vertical Navbar</title>
  <style>
    ul {
      list-style-type: none;
      margin: 0;
      padding: 0;
      width: 200px;
      background-color: #333;
      height: 100vh; /* full height */
    }
    li a {
      display: block;
      color: white;
      padding: 14px 20px;
      text-decoration: none;
    }
    li a:hover {
      background-color: #111;
    }
  </style>
</head>
```

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

Output:

Use Case: Sidebar for an admin panel or dashboard.

Key Points Navbars usually use `` + `` + `<a>` for structure.

- CSS controls **layout, colors, and hover effects**.
- Can be made **responsive** using:
 - **Media queries** in plain CSS
 - **Frameworks** (e.g., Bootstrap, Tailwind CSS, Material UI)

Dropdown

A **Dropdown** is a UI component that lets a user pick one option from multiple available choices.

- ❖ It saves space and organizes options neatly, making it common in **forms, navigation bars, and settings menus**.

Types of Dropdowns

1. HTML <select> Dropdown (Form Dropdown)

- Standard dropdown used in **forms**.
- Provides a list of options to select from.

Example:

```
<!DOCTYPE html>
<html>
<head>
  <title>Form Dropdown Example</title>
</head>
<body>
  <h2>Choose Your Course:</h2>
  <select>
    <option value="html">HTML</option>
    <option value="css">CSS</option>
    <option value="js">JavaScript</option>
    <option value="python">Python</option>
  </select>
</body>
</html>
```

- ✓ Used in forms for user input (e.g., country, course, gender).

2. Custom Dropdown (Hover Menu with CSS)

- Styled with **CSS + HTML**.
- Appears on **hover** (or can be made to appear on click with JavaScript).

Example:

```
<!DOCTYPE html>
<html>
<head>
  <title>Custom Dropdown Example</title>
  <style>
    .dropdown {
      position: relative;
      display: inline-block;
    }

    .dropbtn {
      background-color: #4CAF50;
      color: white;
      padding: 10px 20px;
      border: none;
      cursor: pointer;
      font-size: 16px;
    }

    .dropdown-content {
      display: none;
      position: absolute;
      background-color: #f9f9f9;
      min-width: 160px;
      box-shadow: 0px 8px 16px rgba(0,0,0,0.2);
      z-index: 1;
    }

    .dropdown-content a {
      color: black;
      padding: 10px 15px;
      text-decoration: none;
      display: block;
    }
  </style>
</head>
<body>
  <div>
    <button class="dropbtn">Dropdown Menu
    <div class="dropdown-content">
      <a href="#1">Link 1
      <a href="#2">Link 2
      <a href="#3">Link 3
    </div>
  </div>
</body>
</html>
```

```
.dropdown-content a:hover {  
  background-color: #ddd;  
}  
  
.dropdown:hover .dropdown-content {  
  display: block;  
}  
  
.dropdown:hover .dropbtn {  
  background-color: #3e8e41;  
}  
</style>  
</head>  
<body>  
  <h2>CSS Dropdown Menu</h2>  
  <div class="dropdown">  
    <button class="dropbtn">Menu</button>  
    <div class="dropdown-content">  
      <a href="#">Home</a>  
      <a href="#">About</a>  
      <a href="#">Contact</a>  
    </div>  
  </div>  
</body>  
</html>
```

✓Used in navigation menus.

3. Navigation Bar Dropdown

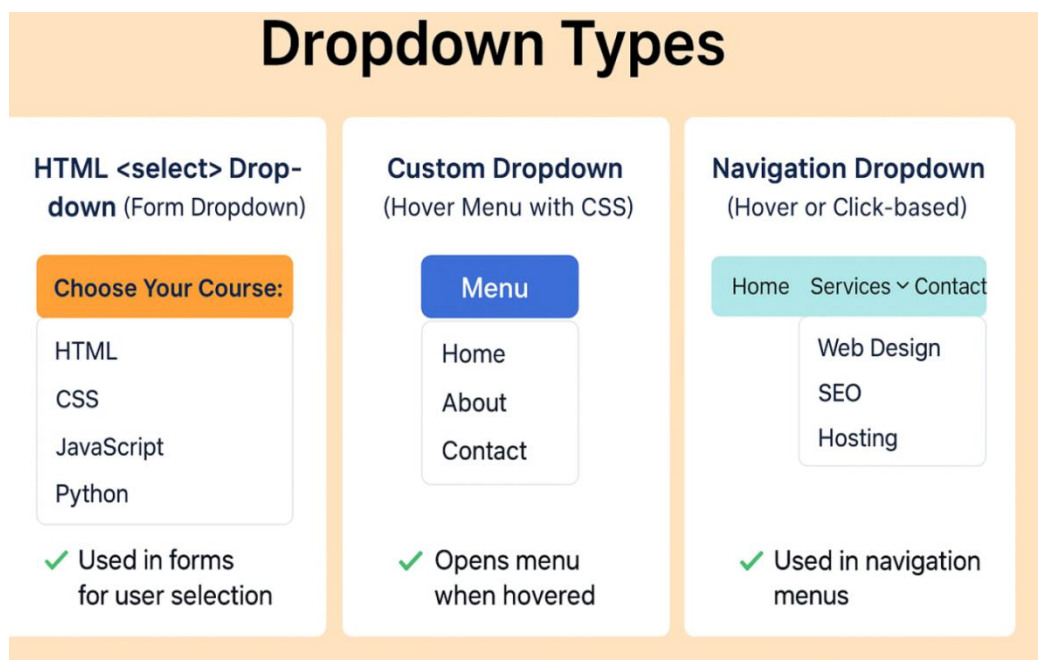
- Integrated **inside navbar menus**.
- Example:
 - **Services**
 - Web Design
 - SEO
 - Hosting

Example :

```
<ul>
  <li><a href="#home">Home</a></li>
  <li class="dropdown">
    <a href="#">Services ▼</a>
    <div class="dropdown-content">
      <a href="#">Web Design</a>
      <a href="#">SEO</a>
      <a href="#">Hosting</a>
    </div>
  </li>
  <li><a href="#contact">Contact</a></li>
</ul>
```

Key Points:

- Use **<select>** for **forms** (data selection).
- Use **CSS Dropdowns** for **navigation menus**.
- Dropdowns can be shown **on hover (CSS)** or **on click (JS)**.



Forms

In HTML, **forms** are used to collect user input and send it to a server for processing.

- ❖ They are widely used for registrations, logins, searches, surveys, feedback, and more.

Basic Structure of a Form

```
<form action="submit.php" method="post">  
  
  <!-- Form elements go here -->  
  
</form>
```

- **<form>** → Defines the form.
- **action** → URL where the form data will be sent.
- **method** → HTTP method (GET or POST) used for sending data.

Common Form Elements

1. Text Input

```
<label for="name">Name:</label>  
<input type="text" id="name" name="name">
```

2. Password Input

```
<label for="password">Password:</label>  
<input type="password" id="password" name="password">
```

3. Radio Buttons (single choice)

```
<p>Gender:</p>  
<input type="radio" id="male" name="gender" value="male">  
<label for="male">Male</label>  
  
<input type="radio" id="female" name="gender" value="female">  
<label for="female">Female</label>
```


4. Checkboxes (multiple choices)

```
<p>Hobbies:</p>
<input type="checkbox" id="sports" name="hobbies" value="sports">
<label for="sports">Sports</label>

<input type="checkbox" id="music" name="hobbies" value="music">
<label for="music">Music</label>
```

5. Dropdown (Select Menu)

```
<label for="course">Choose a Course:</label>
<select id="course" name="course">
  <option value="html">HTML</option>
  <option value="css">CSS</option>
  <option value="js">JavaScript</option>
</select>
```

6. Textarea (Multiline Input)

```
<label for="message">Message:</label><br>
<textarea id="message" name="message" rows="4" cols="30"></textarea>
```

7. Buttons

```
<input type="submit" value="Submit">
<input type="reset" value="Reset">
<button type="button">Click Me</button>
```

Registration Form

Name:

Password:

Gender:
☒ Male ☐ Female

Hobbies:
☒ Sports ☐ Music

Choose a Course:

Message:

Example :

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

  <h2>Registration Form</h2>

  <form action="#" method="post">

    <!-- 1. Text Input -->
    <label for="name">Name:</label>
    <input type="text" id="name" name="name">
    <br><br>

    <!-- 2. Password Input -->
    <label for="password">Password:</label>
    <input type="password" id="password" name="password">
    <br><br>

    <!-- 3. Radio Buttons -->
    <p>Gender:</p>
    <input type="radio" id="male" name="gender" value="male">
    <label for="male">Male</label>

    <input type="radio" id="female" name="gender" value="female">
    <label for="female">Female</label>
    <br><br>

    <!-- 4. Checkboxes -->
    <p>Hobbies:</p>
    <input type="checkbox" id="sports" name="hobbies" value="sports">
    <label for="sports">Sports</label>

    <input type="checkbox" id="music" name="hobbies" value="music">
    <label for="music">Music</label>
    <br><br>
```

```
<!-- 5. Dropdown -->
<label for="course">Choose a Course:</label>
<select id="course" name="course">
  <option value="html">HTML</option>
  <option value="css">CSS</option>
  <option value="js">JavaScript</option>
</select>
<br><br>
```

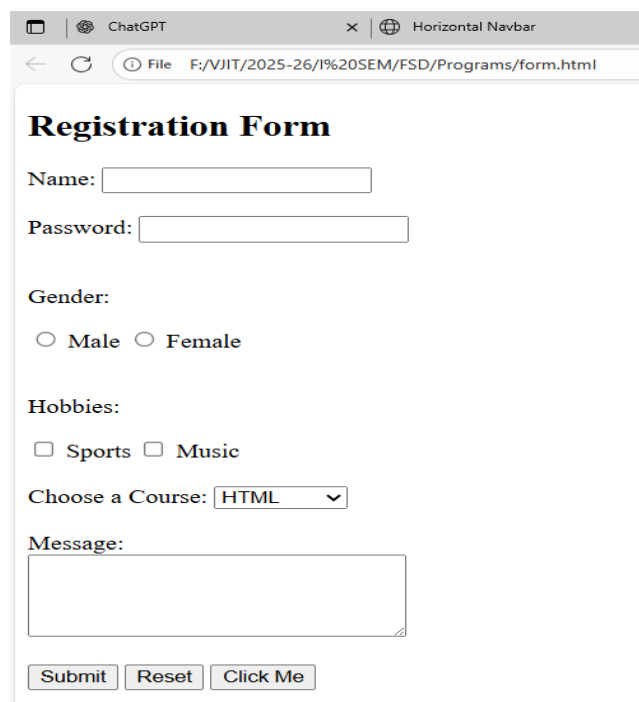
```
<!-- 6. Textarea -->
<label for="message">Message:</label><br>
<textarea id="message" name="message" rows="4" cols="30"></textarea>
<br><br>
```

```
<!-- 7. Buttons -->
<input type="submit" value="Submit">
<input type="reset" value="Reset">
<button type="button">Click Me</button>
```

```
</form>
```

```
</body>
```

```
</html>
```

Output:

Registration Form

Name:

Password:

Gender:

☐ Male ☐ Female

Hobbies:

☐ Sports ☐ Music

Choose a Course:

Message:

List Form Tags Summary Table

Tag	Description	Example
<form>	Defines the form container; contains all input elements.	<form action="submit.html" method="post"> ... </form>
<input>	Single-line input field; can be text, password, checkbox, radio, etc.	<input type="text" name="username">
<label>	Labels an input element (improves accessibility).	<label for="name">Name:</label>
<textarea>	Multi-line text input.	<textarea name="message"></textarea>
<select>	Creates a dropdown menu.	<select name="course"> ... </select>
<option>	Defines an option inside a <select>.	<option value="html">HTML</option>
<button>	Creates a clickable button (submit, reset, or button).	<button type="submit">Submit</button>
<fieldset>	Groups related inputs together in a form.	<fieldset><legend>Account Info</legend> ... </fieldset>
<legend>	Caption/title for a <fieldset>.	<legend>Personal Info</legend>
<datalist>	Provides a list of predefined options for an <input> (auto-suggestions).	<input list="browsers"><datalist id="browsers"> ... </datalist>
<optgroup>	Groups options inside a <select>.	<optgroup label="Frontend"><option>HTML</option></optgroup>
<output>	Displays the result of a calculation or script.	<output name="result">5</output>
<input type="hidden">	Stores hidden data (not visible to users, but sent with form).	<input type="hidden" name="userid" value="123">