

Module 6 – HTML, CSS and JS in PHP

➤ HTML Basics

1. What is HTML? Explain its structure.

HTML stands for **Hyper Text Markup Language**. HTML is the standard markup language for **creating Web pages(WEB SITE)**. HTML describes the structure of a Web page. HTML consists of a series of elements `<div></div><p></p>`. HTML elements/tag tell the browser how to display the content. HTML elements label pieces of content such as "this is a heading", "this is a paragraph", "this is a link, images", etc.

Structure:

```
<!DOCTYPE html>

<html>

<head>

<title>Page Title</title>

</head>

<body>

<h1>This is a Heading</h1>

<p>This is a paragraph.</p>

</body>

</html>
```

2. Describe the purpose of HTML tags and provide examples of commonly used tags.

HTML tags are the building blocks of an HTML document. Their main purpose is to:

- Structure the content of a web page.
- Define different types of content (like headings, paragraphs, images, links).
- Help the browser understand how to display content.
- Provide semantic meaning, making web pages more accessible and SEO-friendly.

Example of commonly used tags:

`<h1>-<h6>, <p></p>, <a>, , and , and , <div></div>,
, <hr>.`

3. What are the differences between block-level and inline elements? Give examples of each.

- **Block-level elements**

Definition: These elements always start on a new line and take up the full width available (by default, stretching left to right).

Behavior:

- **Occupy the entire width of their parent container.**
- **Can contain other block-level elements and inline elements.**
- **Respect top and bottom margins.**

Examples:

- <div>
- <p>
- <h1> to <h6>
- , ,
- <table>
- <form>
- <section>, <article>, <header>, <footer>

- **Inline elements**

Definition: These elements do not start on a new line and only take up as much width as their content requires.

Behavior:

- **Flow inside a line of text.**
- **Do not respect top and bottom margins (only left and right).**
- **Usually cannot contain block-level elements (only text or other inline elements).**

Examples:

-
- <a>
- ,
- <i>,
-
- <label>
- <input>, <textarea>, <select>

4. Explain the concept of semantic HTML and why it is important.

Semantic HTML means writing HTML in a way where the tags themselves describe the meaning of the content. Instead of only using generic elements like <div> or , semantic tags (like <header>, <nav>, <article>) tell both the browser and the

- **Importance of Semantic HTML :-**

1. **Accessibility**

- Screen readers and assistive technologies understand the structure better.

2. **SEO (Search Engine Optimization)**

- Search engines can better interpret and rank content.

3. **Readability for Developers**

- Code is cleaner and easier to understand.

4. **Consistency Across Browsers**

- Semantic tags are standardized and future-proof.

5. **Separation of Concerns**

- Content meaning (HTML) stays separate from styling (CSS).

➤ CSS Fundamentals

1. What is CSS? How does it differ from HTML?

CSS stands for **Cascading Style Sheets**.

It is used to **style and design webpages built with HTML**.

With CSS, you can control:

Colors, Fonts, Layouts (grid, flexbox), Spacing (margins, padding), Animations & effects, Responsive design (mobile-friendly).

Feature	HTML (Content & Structure)	CSS (Style & Design)
Purpose	Defines content & structure	Defines styling & layout
Focus	“What is on the page?”	“How should it look?”
File Extension	.html	.css
Examples	<h1>, <p>, , <table>	color: red; margin: 10px; font-size: 18px;
Dependency	Can exist alone (basic webpage works)	Needs HTML to apply styles
Analogy	Skeleton of a body	Clothes & makeup for appearance

2. Explain the three ways to apply CSS to a web page.

➤ **Inline CSS**

- CSS is applied directly to an element using the style attribute inside the HTML tag.
- It affects only that specific element.
- Best for quick styling but not recommended for larger projects (hard to maintain).

Example:

```
<p style="color: blue; font-size: 18px;">This is inline CSS</p>
```

➤ **Internal CSS (Embedded CSS)**

- CSS rules are written inside the <style> tag within the <head> section of the HTML document.
- Used when styling is specific to a single page.

Example:

```
<!DOCTYPE html>

<html>
  <head>
    <style>
      p {
        color: green;
        font-size: 20px;
      }
    </style>
  </head>
  <body>
    <p>This is internal CSS</p>
  </body>
</html>
```

➤ **External CSS**

- CSS is written in a separate .css file.
- The file is linked to the HTML document using the <link> tag inside the <head>.
- Best practice for large websites (easy to maintain and reuse).

Example:

style.css

```
p {  
color: red;  
font-size: 22px;  
}
```

index.html

```
<!DOCTYPE html>  
  
<html>  
  
<head>  
  
<link rel="stylesheet" href="style.css">  
  
</head>  
  
<body>  
  
<p>This is external CSS</p>  
  
</body>  
  
</html>
```

3. What are CSS selectors? List and describe the different types of selectors.

CSS selectors are patterns used to “select” HTML elements so that styles can be applied to them.

They tell the browser which elements to style.

Types of Selectors :

Universal Selector (*) :- Targets all elements

Example:`* { margin: 0; }`

Element Selector :- Targets elements by tag name

Example:`p { color: blue; }`

Class Selector (.) :- Targets elements with a specific class

Example:`.highlight { background: yellow; }`

ID Selector (#) :- Targets a single element with a specific ID

Example:

```
#header {  
    font-size: 24px;  
}
```

Group Selector :- Targets multiple selectors at once

Example:`h1, h2, p { color: green; }`

Descendant Selector :- Targets elements inside another element

Example:`div p { font-style: italic; }`

Child Selector (>) :- Targets direct children

Example:`ul > li { list-style: none; }`

Pseudo-class Selector (:):- Targets elements in a specific state

Example:`a:hover { color: red; }`

4. What is the box model in CSS? Explain its components

The CSS Box Model is the fundamental concept that describes how every HTML element is treated as a rectangular box in a webpage. It defines how space is taken up around elements and how their size is calculated.

Components of the CSS Box Model

Each element's box consists of four layers (from inside out):

1. Content

- The actual text, image, or data inside the element.
- Controlled by properties like width, height, font-size, etc.

2. Padding

- The space between the content and the border.
- Makes space inside the box, around the content.
- Background color applies to both content + padding.

3. Border

- The line surrounding the padding (and content).
- Controlled by border-width, border-style, border-color.

4. Margin

- The space outside the border (between this element and others).
- Creates distance between elements.

➤ **Responsive Web Design:-**

1. What is responsive web design? Why is it important?

Responsive Web Design is an approach in which a website's layout and elements automatically adjust to fit different screen sizes and devices (desktop, laptop, tablet, mobile). It ensures the site looks good and usable everywhere, without needing separate versions.

Mobile-first world

- Most users access the internet via smartphones.
- A site must be mobile-friendly to retain visitors.

Better User Experience (UX)

- Users don't need to zoom or scroll awkwardly.
- Content adapts neatly to their screen.

Improves SEO (Search Engine Optimization)

- Google favors mobile-friendly sites in search rankings.

Cost-effective

- No need to maintain separate "desktop site" and "mobile site."
- One design works everywhere.

Consistency

- Ensures branding and content remain uniform across devices.

Future-proofing

- Works across new devices with varying screen sizes.

2. Explain the use of media queries in CSS. Provide an example.

Media queries are a CSS feature that allows you to apply styles only when certain conditions are true (like screen size, orientation, resolution, etc.).

They make websites responsive, so the layout adapts to different devices (mobile, tablet, desktop).

Example:

```
/* Default styles for desktop */
body {
    background-color: white;
    font-size: 18px;
}

/* For tablets (screens 768px or smaller) */
@media (max-width: 768px) {
    body {
        background-color: lightblue;
        font-size: 16px;
    }
}

/* For mobile phones (screens 480px or smaller) */
@media (max-width: 480px) {
    body {
        background-color: lightgreen;
        font-size: 14px;
    }
}
```

3. What are the benefits of using a mobile-first approach in web design?

A mobile-first approach in web design means starting the design process with mobile devices in mind and then progressively enhancing the design for larger screens like tablets and desktops.

Benefits of using this approach:

1. Improved User Experience (UX)

- Ensures content and features are prioritized for small screens, making navigation simple and efficient.
- Eliminates clutter by focusing on the essentials first.

2. Better Performance

- Mobile-first sites usually load faster because they are optimized for smaller devices and slower networks first.
- Reduces heavy resources (images, scripts) that could slow down performance.

3. Responsive by Default

- Since the design starts from the smallest screen, scaling up for larger devices becomes easier and more natural.
- Helps avoid issues like broken layouts or text that doesn't fit on small screens.

4. Higher Accessibility

- Ensures that users on low-bandwidth or limited devices can still access the website effectively.
- Makes content more universally available.

5. SEO Advantages

- Google uses **mobile-first indexing**, meaning it primarily crawls and ranks the mobile version of a site.
- A mobile-first design can improve search engine rankings.

6. Future-Proof Design

- With the growing number of mobile users worldwide, designing for mobile ensures long-term relevance.
- Prepares websites for wearables, tablets, and other small-screen devices that may emerge.

7. Focused Content Strategy

- Forces designers to prioritize the most important features and content.
- Prevents overwhelming users with unnecessary elements.

➤ **PHP Integration**

1. How can PHP be used to dynamically generate HTML content? Provide examples.

PHP is a **server-side scripting language that can generate HTML content dynamically before sending it to the browser**. This allows you to create web pages that change based on user input, database queries, conditions, or external data.

Example:

```
<html>
<head>
<title> Dynamic HTML </title>
</head>
<body>
<h1>Welcome </h1>
<p>
<?php
$name = "Dhruv";
echo "Hello,".$name;
?>
</p>
</body>
</html>
```

2. Explain how to include CSS files in a PHP-generated HTML page.

When you generate HTML with PHP, you include CSS the **same way as in normal HTML** — using the `<link>` tag (for external CSS) or `<style>` (for internal CSS).

```
<?php
echo '<!DOCTYPE html>
<html>
```

```
<head>
<title>PHP with CSS</title>
<link rel="stylesheet" type="text/css" href="style.css">
</head>
<body>
<h1>Hello, styled with CSS!</h1>
</body>
</html>';
?>
```

3. What are the advantages of using PHP to manage HTML forms?

Using PHP to manage HTML forms has several advantages:

1. Server-Side Processing

- PHP can handle form submissions securely on the server (unlike JavaScript which is client-side).
- Useful for sensitive data like passwords, payments, etc.

2. Data Validation & Sanitization

- PHP can check if fields are filled, validate email/phone formats, and sanitize inputs to prevent SQL Injection or XSS attacks.

3. Database Integration

- Form data can be easily stored, updated, or retrieved from databases like MySQL using PHP.

4. Dynamic Feedback

- PHP can generate custom responses (e.g., “Welcome, Riddhi!” after login).
- Can redirect users or display error/success messages dynamically.

5. File Handling

- PHP allows file uploads (images, documents) via forms and manages them on the server.

6. Session & Cookie Handling

- Form data can be linked with sessions/cookies for login systems, shopping carts, or user tracking.

7. Cross-Platform & Open Source

- Works on all major servers and operating systems.
- Free and widely supported.

➤ LAB EXERCISES

1. Creating a Simple Web Page

Objective: Create a basic web page using HTML and style it with CSS.

Instructions:

Create an HTML file (e.g., index.html) that includes a header, a navigation bar, a main content section, and a footer.

Style the page using an external CSS file (e.g., styles.css).

Use CSS properties such as color, background-color, font-size, and padding to enhance the design.

Code:-

```
<!DOCTYPE html>

<html lang="en">

<head>
    <meta charset="UTF-8">
    <title>Basic Page</title>
```

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

</head>

<body>

<div class="header">
    <h1>This Is My Page</h1>
</div>

<div class="nav">
    <a href="#">Home</a>
    <a href="#">About </a>
    <a href="#">Services</a>
    <a href="#">Contact</a>
</div>

<div class="content">
    <h2>About This Page</h2>
    <p>This page is created using simple HTML and styled with CSS.</p>
</div>

<div class="footer">
    <p>Copyright © 2025</p>
</div>

</body>

</html>
```

CSS File:

```
body {
    font-family: Verdana, sans-serif;
    margin: 0;
    background-color: rgb(110, 194, 15);
```

```
}

.header {
    background-color: #ce801a;
    color: rgb(15, 10, 10);
    padding: 15px;
    text-align: center;
}

.nav {
    background-color: #333;
    text-align: center;
    padding: 10px;
}

.nav a {
    color: white;
    text-decoration: none;
    margin: 0 15px;
}

.content {
    padding: 186px;
    text-align: center;
}

.footer {
    background-color: #0066cc;
    color: white;
    text-align: center;
    padding: 10px;
}
```

2. Form Handling with PHP

Objective: Create a simple HTML form and process it using PHP.

Instructions:

Create an HTML form that collects user information (e.g., name, email, and message).

Use PHP to process the form data and display a confirmation message with the submitted information.

Validate user inputs and provide appropriate feedback.

CODE:-

```
<!DOCTYPE html>

<html lang="en">
<head>
<meta charset="UTF-8">
<title>Contact Form</title>
</head>
<body>
<h2>Contact Us</h2>
<form action="" method="post">
<label for="name">Name:</label><br>
<input type="text" name="name" id="name" required><br><br>

<label for="email">Email:</label><br>
<input type="email" name="email" id="email" required><br><br>

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

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

```
</form>  
</body>  
</html>  
  
<?php  
if ($_SERVER["REQUEST_METHOD"] == "POST") {  
    echo "<h2>Submitted Data:</h2>";  
    echo "Name: " . $_POST['name'] . "<br>";  
    echo "Email: " . $_POST['email'] . "<br>";  
    echo "Message: " . $_POST['message'] . "<br>";  
}  
?>
```

3. Dynamic Content Generation

Objective: Use PHP to generate dynamic HTML content.

Instructions:

Create a PHP script (e.g., dynamic-content.php) that generates a list of items (e.g., products or blog posts) from an array.

Use a loop to display the items in a styled HTML list.

Style the list using CSS.

4. CSS Grid and Flexbox

Objective: Create a responsive layout using CSS Grid or Flexbox.

Instructions:

Build a grid layout for a gallery of images or a product showcase using either CSS Grid or Flexbox.

Ensure that the layout is responsive and adjusts based on the screen size.

Use media queries to change the layout for mobile devices.

CODE:-

```
<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

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

<title>Responsive Gallery</title>

<style>

body {
    font-family: Arial, sans-serif;
    background: #f4f4f9;
    margin: 0;
    padding: 20px;
}

h2 {
    text-align: center;
    color: #333;
}

.gallery {
    display: grid;
    grid-template-columns: repeat(4, 1fr);
    gap: 15px;
    margin-top: 20px;
}

.gallery img {
    width: 100%;
```

```
height: 280px;  
border-radius: 8px;  
box-shadow: 0 2px 6px rgba(0,0,0,0.2);  
}  
  
@media (max-width: 900px) {  
  
.gallery {  
grid-template-columns: repeat(2, 1fr);  
}  
}  
  
@media (max-width: 600px) {  
  
.gallery {  
grid-template-columns: 1fr;  
}  
}  
  
}  
  
</style>  
  
</head>  
  
<body>  
  
<h2>Responsive Image Gallery</h2>  
  
<div class="gallery">  
  
>

>

The image consists of a single, extremely long horizontal line of text. This text is a base64 encoded string representing an image. The content of the image is a watermark or logo, which appears as a faint, illegible pattern across the entire width of the page. The text is monospaced and contains no spaces or punctuation other than the standard characters used in base64 encoding.

dasobna28p5o2Kx80alzAbmyepxio9qh/b3H/KaRtFFSk+zoMGt6ZJEGkIZVHXcahXtHZtcl4UYAH  
AyMcfpWHMsQxtRhjfOabLPvYFGxnjgULxH0ZdM2vaHWrN7VFglRpSc4B56VIDfqJF2sd2MAk5  
xVa7bTuDZPmVqHMzeyATnwxQVRLZDPxn2Xseu29irYQzSkcEtgCvH7WXcpGxo40XA7sD2T9ao  
2sZfeeNR6Y5qL7Kd2BGc+lK6txo+LYtJdfvZpWk3qHxtDBQoHPgKCm754lw4Kt7zZGSafBYs3ON  
mPE8/hVIHoUrJuYcDx28j4ZOKSdZdmmj4be9iljCwAgtjPhUltqRs33QgZzn2hmrNOzscj+xLNL/C  
qDOfriob7QFtFUuskO794An8KC8j0gs/p8nu2vuR3PaTUroLvvJAE91UOwfQUBPqFzOd0ssjkfv  
MTUsmm49y53/BD+tNSyKE94HKevBptb0l/Cx5BxcSt98geVTR291MA21yvn0FTYWLhdRBT4cb  
j9ad9luZ/akDAHxag6r9hXix6i2eLaqg/aSoreRNKi9Mm25EO4HxGK9rtSPzB/Tz/xk6wKOpGfKi4r  
QMBtUn5VqES1YYFm7fFKIjiXGFtJQPhijmiCi/RIRp0h/8upRpAc5ePGevA5rXR22f/TyfSpRYlw5ic  
GI1Cij9DJR6XHAQyGQc9AasrURKQ7Hbxg5OM1frpgONu7A8wKjuOztvcBSe8Rgeqml1AuJRTW  
sM13wm9SOckACqvWIIRIS3CMgHOTkqa1Fz2YSSLu47mUKPMc1TsdlLwuSswwOhPBNUVRex  
HSvwZtVfkHPzNSRjnB55zyelaD+x9+Nv7aPH8VA9o4tP7P6e76nd7JJo2SJFUnecdOAaDrLoeNF2  
3Kezv7DUbn7Ha3MbXI4APQ+gPQ1ZPpUoUrIV3E9FXJz61zjsrN/3k0x9zO3flpDDAAzXOPFdW8  
ZJeJA58cc1OVZoenTizF2PZGa9hMpkkYONjKQaJfsLcHBR1Xz3Gtsl3Flcj86mBjYe4CKk6sn2Xxs/t  
Mhadi7KA95ejny3UaNH0mDhUAhoa0RhjcYESV4bBWHESVJu5WNSSM4+madleFz8DXi6RZJ  
7SWKufNxWjFh5Ko+dOFnIDwaRtIVUXbKJLAFcJbiL/JgGqrU9MkgGLE3SR28ZHzz9MGtoLdl+6u  
adtkX3QopVJrsZ1lvr8mF/s3rlwcPcSohUEGOQIB6YqNOxNwZD378Hxkn3fpW7f7X1G2hpDeBT  
mOP54oOr9WGHO0UZ2HsZalMtcDdj7g/Wp07J6aCN0ckxHi5JH8qse8u93sjkeC4p5bUOPYHxd  
6CqQGnre1+CKDs3ZjhjaAY8uDRkej2acrBAfxLmof+0HOWnijXpnRQ91cWtohkv9VVV9G5PyHJ  
qiknwjNJVO5loLeFBhViHwjpVn7bV9EuQxF5OuD99WGaVPd+iNl8xTRa8gUcS/PiiYtfj3cqw/1A1  
zGDtOQm2S1BOeuas4O0lgUDPFKCOqhVNa9K5mVaHZ0iPW4SNwkYD0VTU/9Yo+MvMB15Cf  
zrlk/bEqGFpZIngjyknA9QP51Vv2m1ZiT9tYZ+6iqoHwwKGg2dLyaa43Oyy6wkX+AHZviv8AOmw  
6xcueUfHkBXGk7Qaoowt/OvOeHqdO1OsqqhdQuPZ6e1/zmj+nYq8umuYnborhnGXjlGfEmntI  
MHa8mfSuIntPrMnD6hPj+EhfyaOlO1ms71b7a77R7rAEEeoXsvxZ9M5eZSvwddaV8n2XYHj3hX  
EP6TdfTWde+zW5/u9lmMNn3n+8efXitXF253Rr32nhnHUpKQPp/vXJr6Yz3txO3WSRn+pzU1SI  
B/Ei8q1OovgY6Fu7kVkbkcg12b+jPtNLrMUIjqLgzW6ho5G6uvTn1HHPwrikTDofdNav+jfVlbDt  
AXuWZlngdSyjJB4P6UKm8dq009+D6Hh7kAe0G+dEq46ltYZO2OiQsFF/Mc9SIRj8afc9u9HhIE  
U9xN6qgUH5msijW+Vmtuj8yNwWI6qUHxpCYj33Uf6qwb9v8ASymdl47/ALu5R+NAS9vFZwLfT  
sr4d7MSfyrtOu+EdnQXLOmrcQngzAn40hd2xBHeMceSH8OOa5xD2/0zJSW1Tvv3FmFejt3Ylj  
0tMeHJpVT8j0dIQ6Z0iOeKVA22UZPAZCD/tUEt1Mn+Dp8sjZwpYgD9Tiux9vp2IX/s4CPx9gk/nR  
Ldur0k/Z9POPEtHz+RrtGu2dnTS2Zrb/APrqajRb9xbH7zCMyN8s4H4ULDHWpoJUhTunHRhbjLf  
AsSPwrN/241gkBrOZU/8AZj/mppx7b3/INpqBH/xjr/wUJUaq5/4UhKNtrfkXPaH7PHGN0mdx  
1b7QgzVa+l9oZOY9OELk5717rex+eaGPbHV3Ib+r77nqUjxx/wVKO1moMABY618FQE//mhhWj  
1/PuUdWL7X5B27Ja9LzI8Y8MGcnFN/sVqrcPcQj0BNSz9pdTlxHpressPEsSM/RaFPaLV/DQ9TI8  
MzS/wAqZfq+tvsk5UU3+GSHsPf+N1H9D/OvajTtDrJHGH6h/wDbJxtC/me/9HZUP4mcoa500e5  
HO4H70yqfptoW61GLf/dl9iY++4Yn8qtdfhtbC3RobZQZWILbc7QKoEmToUGPhXquUlseGoR5sa  
nsuul6qTBe/sJwMgmQhXH6GrnUdM0OwiDC2vLsk42wJI2PiegrA/ahCMwOY2PjEQKms9bvbN  
w0F1MD4kyFvwNdqMbTj6OhaPomhajaCdrWe2JOO7uC6t+dW0fZPQuMJkf52P61luzPaGHWL  
yOz1KzieZvcICZB+IOcfGt7bRQxLtijjQDptXrXzv2MoR9Aa9ldFwMW+fkap10O5j1WdDoFpJp4fE  
Th9r7fPnPnIPw4rYwOSTuDY8wOKkJI4B2/Klzk6UX0YjXNLvlbiP+qNH0/7OqbpGuBliflc9BXJpmLy  
uxABLE4A6V9DSur7IZIC4x1BFcO7V6a+l65dRPF3aNIWiGMKVJ4IpHNvkppqPBWWjbZldcblYM

ARkcVpNCKhj7Y29/drFBZu5IkEbhlQMp9nj1PlrLowR935VcaVDdapcpa2CiS5f3YwwXPHPWjGz  
5Fk5Lg7LpT6DqfeGxKTd3y4SL3fwqC+1TS7OYxLpGqSEHrHp74Pz21B2H7LS9nu9uLq63XE6hW  
ERARQPDPiea2CtGy8Mp8/azUJT32NMIfDuV+mwWuoWUd1HaGFZM4juY+7cY8wRQfaqWPS  
dCvZk7iGQxMkTBed7DC456856VfO8hQ9yyccc8fjWA7Y6br+uOts1hbLbQtvR1uWJY4x0x+HN  
Fty5KyilHZHOpAgwi2yT93HSug/wBG2r2tpZS6dezQiVrnNtFvGW3ADA4x1HT1qosuwczjdO7Jif  
ex7v41rtI7D2FtLBKyySshDcDjPnxVatSONrkKFKopJlx2hudatoRoekpJK5O5pmGE8uBjNcx1e27  
ZXdw39YT3fA4Ebkl9FwK7SgbbtA4XgDAP45oK4j/AGgJU4PmRWSFbHaxvnRz7OadmY+2lpPGI  
BvhLAGO5mLKR88kVs7satHfTzQ/apxs2pAYUWHPnnrnr63Hd4bu8H4V7dTYCj2uem0Ampzr3f  
BWnRsrXMB2s7RatY26wLZ3MFwxBM8UQZQPIHGCflWNuu2Os/ZVtZryReSc4Acj1I8K6/egTAc  
kkdScVjO0+talmbcWkF/PtI3ArhfIZwea10JqWyiY/LpygruRg7TXNUhuBPZXF1vB67mYH5V0S07  
VbrCCa+urKGZ19ulmQMD48YOPrXOZNUYF0WIIB0HTaPLFBiR5CWJY856dK1yhGR5sKs4cHQN  
Z7cTQ3CppZheMD2pG9rcfhkEY9etKueO5RsJjHqaVLjEbUn7Nw1vFPE0c5z064+tBtoVm5Xvlyy  
dM5oqK6t12hHRfqlWeNfaJbB8wFB/CrWJZFceydi6EqWQ+PAIql9k4kkG5l2+HsAH86voJxnlikl  
/zkD68fIRC3LNhY1QMPuh/wAenWlxGyPNA0iy0vdII3e6JOOh9a0cdwhXruxVJHNMcSztp/iIu4  
+QxU8Zce7JI3+R2rrDKVkWv2jJwsTEjqA4/WnNcF1KtbIVPi3P5VVq97uARUY+G4gfXJoLUbt7W  
1nuLmC1HdKWOCuTj025P1oYoGY/tL2kTSYFht1QXUg9hDGQFH7xzXOry4lvZzc30zzSk+85zz6e  
Q9KbJcy6ldPdTEF5DnaOijwA9KhvZvhYKBlh4UjObbYVHbwylkQkcEY5PnRENvbqB+zwAchxwy  
n0PhVVZSMAzSOeTkYok3LA4yvXxoqSJyjJPY6P8A0fsI9PnjmuTeyCXcqT+/Gnx6nJrQ3Wo29rA8  
v2EOi8lYve+grk+j6qLG+guGbKA4dc9VPWt5c2zyTBlvIDGcFRIDux/pYUjhG5qhVliOb+kPSo8qlro  
NzlcHI+VW+ka/baxbM1qO6cNho5wVJ/Ej8aqo9Nt3QGW/Z5sHiDYoP1yfxovvtPjiCTyAknALSbQ  
foaScIW2RWnOfZZp329m2BSD91D+eKI79ud075HVdhY/XFukkujRAOURUHVhJx9MGnLqeIxby  
4VlfyD3uB8MMoSps1Rqx9l2moJEP2xJOfd2sW/DP5UxLvvWJVZwvrAR+dVH9ZW/tiNJlwONyt  
E6p8fGhzeRSMzG5gk2dcQBCPXOTU9JltRGhuBE2O/3AebK38qqdQ1fTLJUP9YxgH3QA3P60GZ  
8hXFy212ABUcfUmqrtnBobai0M6TtkEBud4I8s44poePd/EwVK7hG8UM7T9qlodNU6dKJXudyC  
VcgR4xnPkeRXOy+33WJHU5Xqa3+r9nbefQnS0EQuklHX3ml6j5jNc4SWPlwzA/DrWynGMFZH  
l15zqPKRPKwuAN21X/eUcEfWvVulhi2KuT4nOaHAifJZ/iB40xxE8e5WGfDAwapchb2RyM8rlgp  
HpmISE21QNzZ8aVJsPublTIOAkQ9B/tUyF05Tu1PouPxqrW5mPAkcfOnhvGVnJ9eKuQLYS3Lcvc  
rj+PBr2R53P8A45T6DP5CqvcTxEhLebNxTH1EwcZjPgcL0ogbLuGG/VR/fwiZ915GUfSiVi9o238  
RHkWC1kZNRMm4xuFPTCrj50MJO95M5fnkuc/SuFv9DR9pLy+sLSLublXeQ4zGdwQfzNZI3hd2  
a4iLNklj5+dGuWXBU4wODnHNMLvgBmyPPdQIF+zINLoFSezjQnZMDnrkfzoGeaFpCU7wZPU  
rVhNDFJw6rn94AD8qrLq2aI7gcrUZJo0U3Fikn3BQrMQB0wOKaGUn2mJNQ816EJ6Ck2RaWhj  
chinCnJNxjx/Wup6Xqwg0u2gikZlVAFd1Ull8Mg+lcq0jTJr2dVCHZ1Y+GK6XATHAkaOwRFwAjEA  
AfCq04ZcmepUUHZFuNWg2t3pdCerJGoJ/ChpbqylJaDUriFj1/Z4J+YqtumUtge1xwSckfOojCIV  
AQ7zyTuyD8qfTS4Aq8mWsBlecLDqPeZ8HiGPnmop7treQO4gmdTtyqKuzPiNooG1COhWTeoU  
8lVPX5DmlcRWzSkvO6eAKxnPzzihgh1WfolSxvFYxhkkyD3jMzBvl0HzxiIZFoooFaAm4De9HIZVH  
r14+VASxxiMfZbiXZghiQTk0re5CMju9xIQOnfMoU+fH5Urhfgr9MMW3kaCOeOCfbj/CjDYA+O  
AfxNTwmMtL7cn7LBlaTecnyPDFi/Lki08yKerSSu2D8P+tex5jkjKE4AzHuG4fMCjGDFIXilsVParW  
rSO3kggnxKwKsHg/DlrnO3ZyM5FdH7U28l3DIzdj9GVeN3rwKxZ0mfGSrZycjb0/GldOSO14SKg  
k16j7T6eVGyWpiYB4jxwcZqKSOD7rOp8mFLZodTTIt6HkqM0q82Y6HPwpUA7GmW48FPXzpO  
Qg3zyHHwzmopbqGDhUZ29TxVXcTSXDbpcj0zV2zMIcPe/MgKxkpH0znk1CjqRtUdPOg9645PP  
pS78+DYHwpcguATIQwUMi8fWvAgByrcjwJqBZucE5z0xS7wKBkn9K66OxYS120Y5jPH0pgu1HI

yPMeBpqMrjGC2fAdKlttPRjukJHPTNFZPgDcFyS25+0uojUnPU54WvNYhSOEFX6cdRg/rVpD3E  
aBV6joAKqtaaKSRMEhh1z0ppq0dyVOWU9imXbnyohVA55aofZU4xmpo5DuBBO3HlrMbJGv7  
NzRRxBApjkPUgD2qv3MgGVYAH96sjpCROyMXI8evNXW9sg5JGPu1tp/tPMql4wt2C5LvjPI/0r  
1ZZGjIGFiB5I6/hQrTZGCrY8TTUm29CxHkRwKewLtBa3UcWFFwxAOQpz186lF2ZYcm5IAPQORi  
hGvB4of+Gm98jHc6sP9IxQsg5ytyGRXbe6JHbH8RlrzuwZVkwVDH2uuDUQchRNGitjg54xUxcs  
VZcsHGNvBxRSQJVJcXJgqB/Z7zbnpk0VGoUjAO088mhbYxNnDZPQjOMURGwAbZ9DTE8vZBe  
Wqyj3Q58w3tD4f9aqbjSNo3RSyRufBsn9KvWIAHQAjplcjNRrN3y73hXI8iDn5UGrhUvRjr63nW  
UlyG/iKYz86rZ7Tdz3W5vQ1vLxe9U7ZEBAwUZTn8aoriBSxRFjfP3Su0/yqcoFoVmjjPEitgmQY8O  
KVXsmn+17Mc3+lgfzpVHSZpXkRK3cpBBBjoWeFicq270NSFvhTAc0NmNG8QQo4PQ04LIegosD  
NOCge8aGJTUBlgkYctip4rZc+0ST8akDDoq5qZVzgn8KZRRKVRnsKBcdKLDEcZ49BQ4xT+89nAF  
UWxmldhUe04OPCUmgdYYMoCKvHio5qUOwGNxpk/tpj/n6UJboNN4yuUvI6jNKJCzgAnr4UTK  
Nj4OPkKZFxFISMjPj5Vnsbsti80+3+z8vkeXOKsUuinukn55qoiEuAyuGHiT1NFq6sMFgG8q1R2R59  
T9xZLdPMP8NcfHrXpYtyuF9M1WMXQcHj0NR/aGzy3FNcRItnlliySpOPKmfbXkGJEAHhQNvqBj  
99iyj8Kka4jlJlyc/WuuNawUs0y+6odPQ80+K9dXO1XiJ655FVgnKH2d3yNSfa4yckHPrQuCxaNJK  
w3OFOfvDg06KaRT/iNjyqujuWHKjipUv4x76YpkxGi0F1IB1yB9ad3iOuQ5VvHFCW95A5JUgEef  
WvZJ4jwRsPmOhprgsG7J2we87zHOCK8IlycyBRjzFV7XMq8owx4YNNN3ifeUE+dG4LEkrsJGBIH  
xH+1e1H9rB94kHxFeUA2MZH7VTKi+NNVQvT515K+MVkPRe72JdvISMI+83PlQxIHnXneEjiuuH  
BhissfA8alVqDiUnljU4PAFFMnKJPu9RT1ORQu6nBj4nA8qNxHEnyPL6Us9ePDxqlvDjilvJ4NG4u  
I2XD43AU2NFUkkYHw60mKDJPUVWPXcTSMqk7FgrqzezimNN5ZFBi7E9OKnV/A8UchdOwQ  
kuePCnzINhKHOo8qGDKKaZD4mmuLhuLdtPNSpOc5VgMedDAK3VmBprKy42nloXhxTLE3SO  
MNGvxzTDsPQ7R60GWpd4RzijkDD0GhyvRuPSnhTw8aDVwR6/CpEfByrUUybgGxN3dEreFRtIG  
PWgGnO2o8hhknFNkLgWZmDchceopnfMOlApIwOEJNSM7Ac0cjsUFd7hqmTXIBd6fKIXXFxYB  
4UNcGlSrOzdDkgjaljpUqVFpcBa+7Xq+6aVKnMxHMxU8U+MkjmlSoIL/aegnJFJiQvFe0qLADtz  
XqnJpUqUoJSdxqZTkUqVFAZ4Sc0mPsmlSogGANNSjmISrgSE1Ida9pUQDXGBkUgxxSpUGcSxM  
SvNeNwa8pUwnZNH0+VTqxK817SpkTZ5mlSpUQH/9k=" alt="Image 3">>









```

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

## **5. Styling a PHP Application**

Objective: Apply CSS styles to a PHP web application.

Instructions:

Create a simple PHP application (e.g., a user registration page).

Use an external CSS file to style the form elements (e.g., inputs, buttons, labels).

Ensure that the application is visually appealing and user-friendly.

**CODE:-**

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="UTF-8">
<title>Contact Form</title>
<link rel="stylesheet" href="A-Q-5.css">
</head>
<body>
<div class="">
<h2>User Registration</h2>
<form action="" method="post">
<label for="name">Name:</label>

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


```

```

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

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

<label for="password"> Password:</label>

<input type="password" name="password" id="password" required>

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

<?php
if ($_SERVER["REQUEST_METHOD"] == "POST") {
 echo "<h2>Submitted Data:</h2>";
 echo "Name: " . $_POST['name'] . "
";
 echo "Email: " . $_POST['email'] . "
";
 echo "Password: " . $_POST['password'] . "
";
}
?>
</div>
</body>
</html>

```

CSS file:-

```

body {
 font-family: Arial, sans-serif;
 background: #f3f4f6;
 display: flex;
 justify-content: center;
 align-items: center;
 height: 100vh;
}

```

```
margin: 0;
}

.header {
background: #fff;
padding: 25px;
border-radius: 12px;
box-shadow: 0 4px 10px rgba(0,0,0,0.15);
width: 350px;
}

h2 {
text-align: center;
color: #333;
margin-bottom: 20px;
}

label {
display: block;
margin: 10px 0 5px;
color: #444;
font-weight: bold;
}

input {
width: 100%;
padding: 10px;
border: 1px solid #ccc;
```

```
border-radius: 6px;
margin-bottom: 15px;
font-size: 14px;
}
```

```
input:focus {
border-color: #007bff;
outline: none;
}
```

```
button {
width: 100%;
padding: 12px;
background: #007bff;
border: none;
border-radius: 6px;
color: white;
font-size: 16px;
cursor: pointer;
transition: 0.3s;
}
button:hover {
background: #0056b3;
}
```

## **6. Implementing a Responsive Navigation Bar**

Objective: Create a responsive navigation bar using HTML and CSS.

Instructions:

Build a navigation bar using HTML <ul> and <li> elements.

Use CSS to style the navigation bar and make it responsive (e.g., using media queries).

Implement a dropdown menu for sub-navigation items.

**CODE:-**

```
<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<title>Basic Page</title>

<link rel="stylesheet" href="A-Q-1.css">

</head>

<body>

<div class="header">

<h1>Header </h1>

</div>

<div class="menu">

Home

About

Services

Contact

</div>

</body>

</html>
```

**CSS File:**

```
body {

font-family: Verdana, sans-serif;

margin: 0;
```

```
background-color: rgb(110, 194, 15);
}

.header {
 background-color: #ce801a;
 color: rgb(15, 10, 10);
 padding: 15px;
 text-align: center;
}

.nav {
 background-color: #333;
 text-align: center;
 padding: 10px;
}

.nav a {
 color: white;
 text-decoration: none;
 margin: 0 15px;
}
```

## **7. Image Gallery with Lightbox Effect**

Objective: Create an image gallery that opens images in a lightbox effect.

Instructions:

Use HTML to create a gallery of images.

Implement CSS for styling and layout.

Use JavaScript or a CSS library to create a lightbox effect when images are clicked.

**CODE:-**

```
<!DOCTYPE html>

<html lang="en">
 <head>
 <meta charset="UTF-8">
 <title>Lightbox Gallery</title>
 <link rel="stylesheet" href="style.css">
 </head>
 <body>
 <h2>Image Gallery</h2>

 <div class="gallery">

 < https://encrypted-tbn0.gstatic.com/images?q=tbn:ANd9GcQoRh9o1uTWRPGq7wvFuS-5tzwXOywtG6X_6H7_7hk9gNQzY-hdynp0uvU&s ="img8.webp alt="Image 4">

 </div>

 <!-- Lightbox -->
 <div class="lightbox" id="img1">
 ×

 </div>
```

```
<div class="lightbox" id="img2">
×

</div>
```

```
<div class="lightbox" id="img3">
×

</div>
```

```
<div class="lightbox" id="img4">
×

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

### CSS File:-

```
body {
 font-family: Arial, sans-serif;
 background: #f9f9f9;
 text-align: center;
 padding: 20px;
}
```

```
h2 {
 margin-bottom: 20px;
}
```

```
.gallery {
 display: flex;
 justify-content: center;
 gap: 15px;
 flex-wrap: wrap;
}

.gallery img {
 width: 500px;
 border-radius: 6px;
 cursor: pointer;
 transition: transform 0.3s;
}

.gallery img:hover {
 transform: scale(1.05);
}

/* Lightbox styling */
.lightbox {
 display: none;
 position: fixed;
 top: 0; left: 0;
 width: 100%;
 height: 100%;
 background: rgba(0,0,0,0.85);
 justify-content: center;
```

```
 align-items: center;
 z-index: 999;
}

 .lightbox img {
```

```
 max-width: 90%;
 max-height: 80%;
 border-radius: 8px;
 box-shadow: 0 0 15px #000;
}
```

```
.lightbox:target {
 display: flex;
}
```

```
.close {
 position: absolute;
 top: 20px;
 right: 30px;
 font-size: 32px;
 color: white;
 text-decoration: none;
}
```

```
.close:hover {
 color: red;
}
```

