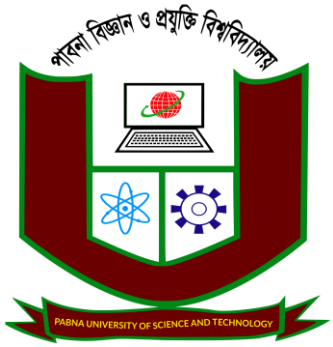
Pabna University of Science and Technology



Department of Information and Communication Engineering, Faculty of Engineering and Technology.

Lab Report

Course Code: ICE-3104

Course Title: Web Programming Sessional

Submitted by:

Name : Sabrina Ahmed
Semester : 3rd Year 1st Semester
Roll : 210610
Session : 2020-2021

Department of ICE
Pabna University of Science and Technology

Submitted to:

Md. Anwar Hossain
Professor

Department of ICE
Pabna University of Science and Technology

Date of issue: / /

Submission Date: 03 / 11 / 2024

INDEX

SL. No.	Name of the Problem	Page No.
01.	Basics of HTML a. Create a simple HTML page to demonstrate the basic structural tags of HTML. b. Create an HTML page to demonstrate the usage of basic text formatting tags. c. Prepare an HTML page to display the syntax and usage of any 5 text formatting tags.	
02.	Image in HTML a. Create an image gallery of any two popular places in Bangladesh with their short description. b. Demonstrate the use of align, vspace and hspace attribute using minimum of two image.	
03.	LISTS IN HTML a.Create the following list using html. 1. BSc a. Semester-1 ▪ ICE-3101 ▪ ICE-3102 b. Semester-2 2. MSc b.Create the following list using html. A. Books i. A brief history of time. ii. Web Programming. B. Music C. Reading D. Movies	
04.	Create an HTML page to display the registration form. (Users Name, Father's Name, Mother's Name, Date of Birth's, Gender, E-mail, Address attributes).	
05.	Create a style sheet named style.css to improve the appearance of your University. Your style sheet should do the following without any modification to your HTML code: • Change the font properties of at least two elements (such as family, size, weight, style). Here are some standard fonts you may want to use: Arial, Arial Black, Verdana, Trebuchet MS, Georgia, Tahoma, Courier New, Times New Roman.	

INDEX

06.	Create an HTML page to display the registration form that are design an internal style sheet to contain a core set of style used in all pages within this problem. At a minimum define the following CSS properties for elements on your page: background-color, color, margin-xxx(left, right, top and bottom), padding, border-style, border-color and border-width.	
07.	Create an HTML page to display the registration form that is design an external style sheet to contain a core set of the style used in all pages within this problem. At a minimum define the following CSS properties for elements on your page: padding, border-style and color, text-decoration, text-align, font-size, font-family, font-weight.	
08.	Create an HTML page to display the Bangladesh flag.	
09.	Write a JavaScript program to display the current day and time in the following format. Sample Output: Today is Tuesday. The current time is: 09 PM: 15:00	
10.	Write a JavaScript program to find the largest of three given integers.	
11.	Write a JavaScript program to calculate the multiplication and division of two numbers. 1st Number **:** **12** 2nd Number**:** **10** Multiply Divide The Result Is: 120	
12.	Write a JavaScript program to draw two intersecting rectangles, one of which has alpha transparency.	
13.	Write a program to create a chess board in PHP using a for loop.	
14.	Create a simple HTML form and accept the username and display the name through a PHP echo statement.	
15.	Write a PHP script to display string , values within a table.	
16.	Create a simple 'birthday countdown' script, the script will count the number of days between current day and birthday.	
17.	Create a PHP registration form and submit the data to the database and retrieve the data from database and show details in browser.	

Problem No: 01

Problem Title:

Basics of HTML

a. Create a simple HTML page to demonstrate the basic structural tags of HTML.
b. Create an HTML page to demonstrate the usage of basic text formatting tags.
c. Prepare an HTML page to display the syntax and usage of any 5 text formatting tags.

Part-a: Create a simple HTML page to demonstrate the basic structural tags of HTML.

Procedure:

HTML (HyperText Markup Language) is the standard markup language used to create and design web pages. It is the foundational code that browsers read to display content on the internet, including text, images, links, and multimedia etc.

HTML tags are the building blocks of an HTML document. They are used to define the structure, content, and layout of a webpage. HTML tags are enclosed in angle brackets (< >) and usually come in pairs: an opening tag (<tag>) and a closing tag (</tag>). The content between these tags is what gets displayed or structured on the webpage.

The purpose of is to demonstrate the fundamental structural components of an HTML document by creating a basic webpage that utilizes essential tags to organize content.

Key Structural Tags in HTML:

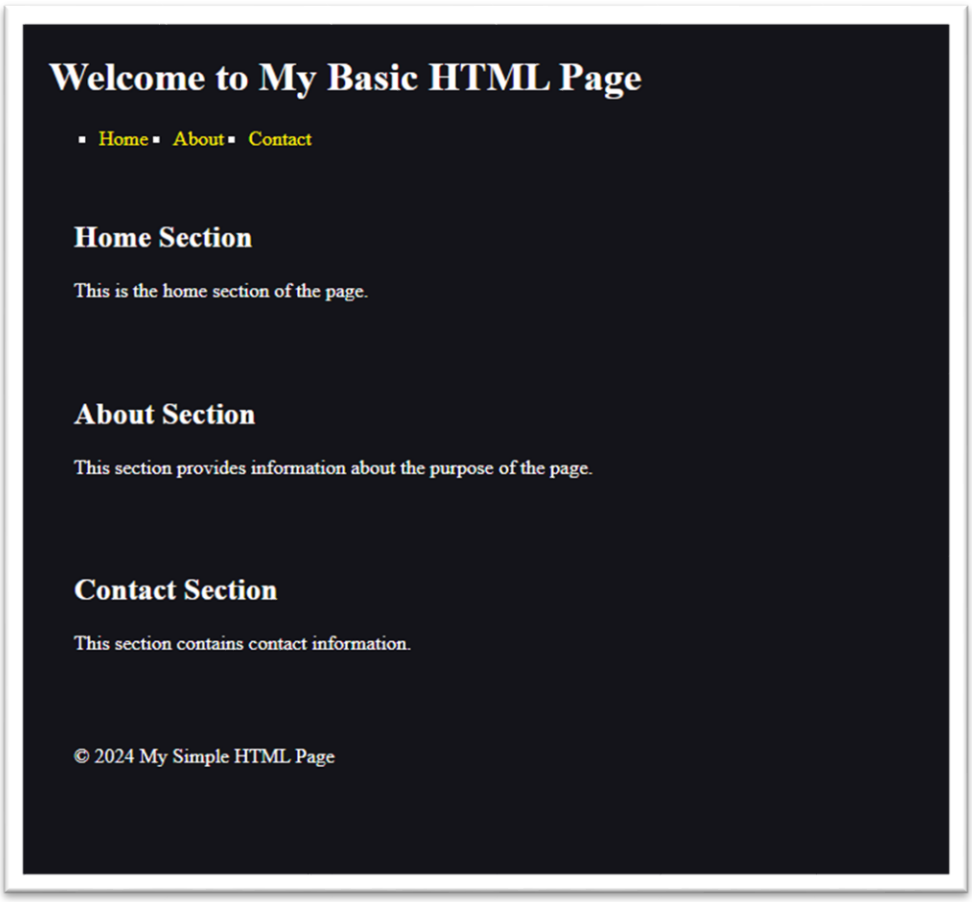
1. **<!DOCTYPE html>** :
    - **Purpose**: This tag tells the web browser that the document is written in HTML5, the latest version of HTML.
2. **<html>** :
    - **Purpose**: This is the root element of an HTML document. Everything in the webpage is enclosed within this tag.
    - **Attributes**:
        - lang="en": Specifies the language of the document, in this case, English.
3. **<head>** :
    - **Purpose**: The <head> section contains meta-information which is not displayed on the actual page but is essential for the browser and search engines.
    - **Contents**:
        - **Metadata, Title, Style.**
4. **<body>** :
    - **Purpose**: This section contains the visible content of the page. All the elements that users see, such as text, images, and buttons, are placed inside the <body> tag.
5. **<header>** :
    - **Purpose**: Used to define the header section of a webpage. This usually contains introductory information like the page title or logo and may include navigation links.
6. **<nav>** :
    - **Purpose**: The <nav> tag is used to create a navigation menu, which helps users move around the webpage. It often contains a list of links.
7. **<main>** :
    - **Purpose**: This tag contains the main content of the webpage. It groups the most important sections that are relevant to the primary topic of the page.
8. **<section>** :
    - **Purpose**: The <section> tag defines a section of the content, helping to organize the page into smaller, easily identifiable parts.
9. **<footer>** :
    - **Purpose**: The <footer> element is used to contain the closing content, such as copyright information or important links.

Source Code:

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Basic HTML Structure</title>
    <style>
        body{
            background-color: rgba(0, 1, 7, 0.947);
            color: #fffbfb;
            padding: 20px ;
        }
        ul.horizontal-list {
            list-style-type: square;
            display: flex;
        }

        ul.horizontal-list li {
            margin-right: 20px;
        }
        nav a {
            color: #fceb02;

        }

        nav a:hover {
            color: #fd04a6;
        }

        section {
            padding: 20px ;
        }
        footer {
            padding: 20px ;
        }
    </style>

</head>
<body>
<header>
    <h1>Welcome to My Basic HTML Page</h1>


    <nav class="navbar" >
```

```
        <ul class="horizontal-list">
            <li><a href="#home"style="text-decoration:
none;">Home</a></li>
            <li><a href="#about"style="text-decoration:
none;">About</a></li>
            <li><a href="#contact"style="text-decoration:
none;">Contact</a></li>
        </ul>
    </nav>
</header>
<main>
        <section id="home">
            <h2>Home Section</h2>
            <p>This is the home section of the page.</p>
        </section>

        <section id="about">
            <h2>About Section</h2>
            <p>This section provides information about the purpose of the
page.</p>
        </section>

        <section id="contact">
            <h2>Contact Section</h2>
            <p>This section contains contact information.</p>
        </section>
</main>
<footer>
    <p>&copy; 2024 My Simple HTML Page</p>
</footer>
</body>
</html>
```

Output:


Welcome to My Basic HTML Page
Home
About
Contact
Home Section
This is the home section of the page.
About Section
This section provides information about the purpose of the page.
Contact Section
This section contains contact information.
© 2024 My Simple HTML Page

Part-b: Create an HTML page to demonstrate the usage of basic text formatting tags.

Procedure:

HTML (HyperText Markup Language) provides various tags for formatting text, which can enhance readability and emphasize content.

To create an HTML page that demonstrates the use of basic text formatting tags such as bold, italic, underlined, strikethrough, subscript, superscript, highlighted, and text size variations. The main categories include:

- **Bold Text**:
  - <b>: Makes text bold but doesn't imply importance.
  - : Also makes text bold, but with added semantic importance.
- **Italic Text**:
  - <i>: Displays the text in italics, often used for style purposes.
  - : Italicizes text but with emphasis, giving it more meaning.
- **Underlined Text**:
  - <u>: Underlines the text, often used to highlight or distinguish it.
- **Strikethrough Text**:
  - <s>: Puts a line through the middle of the text, indicating something has been deleted or is incorrect.
- **Subscript and Superscript Text**:
  - <sub>: Lowers the text, useful for formulas or scientific notations (e.g., H_2O).
  - <sup>: Raises the text, commonly used for exponents (e.g., x^2).
- **Highlighted Text**:
  - <mark>: Highlights the text with a background color, usually yellow by default.
- **Small and Big Text**:
  - <small>: Decreases the size of the text compared to the surrounding text.
  - <big>: Increases the size of the text.

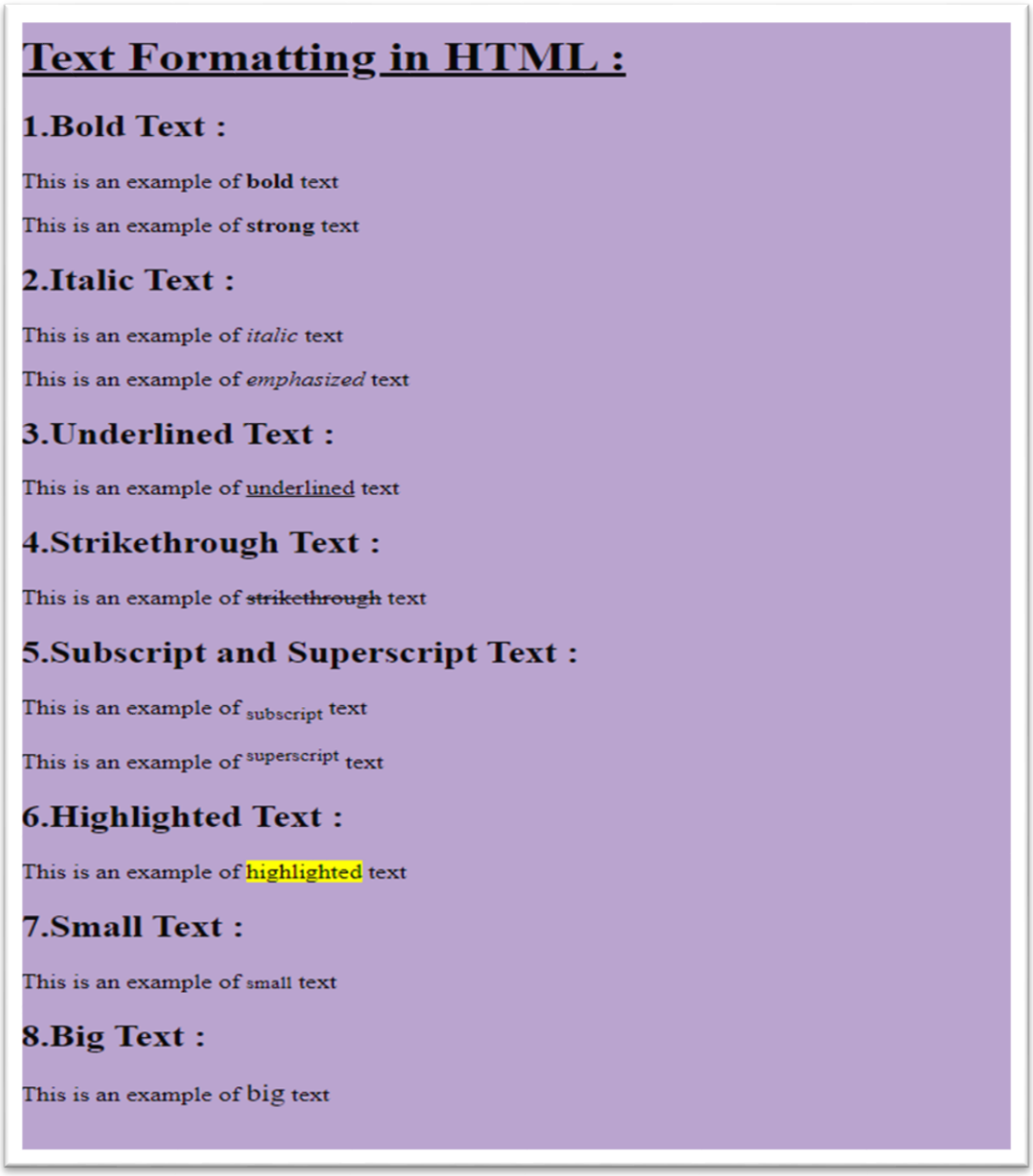
These formatting tags allow us to present text in different styles, making it easier for users to read, understand, or visually distinguish specific parts.

Source Code:

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-
scale=1.0">
    <title>Basic Text Formatting</title>
    <style>
        body
        {
            background-color: rgb(183, 161, 204);

        }
    </style>
</head>
<body>

    <h1><u>Text Formatting in HTML :</u></h1>

    <h2>1.Bold Text :</h2>
    <p>This is an example of <b>bold</b> text </p>
    <p>This is an example of <strong>strong</strong> text </p>

    <h2>2.Italic Text :</h2>
    <p>This is an example of <i>italic</i> text </p>
    <p>This is an example of <em>emphasized</em> text </p>

    <h2>3.Underlined Text :</h2>
    <p>This is an example of <u>underlined</u> text </p>

    <h2>4.Strikethrough Text :</h2>
    <p>This is an example of <s>strikethrough</s> text </p>


    <h2>5.Subscript and Superscript Text :</h2>
    <p>This is an example of <sub>subscript</sub> text </p>
    <p>This is an example of <sup>superscript</sup> text </p>


    <h2>6.Highlighted Text :</h2>
    <p>This is an example of <mark>highlighted</mark> text </p>
```

```
    <h2>7.Small Text :</h2>
    <p>This is an example of <small>small</small> text </p>


    <h2>8.Big Text :</h2>
    <p>This is an example of <big>big</big> text </p>


</body>
</html>
```

Output:



Part-c: Prepare an HTML page to display the syntax and usage of any 5 text formatting tags.

Procedure:

HTML (HyperText Markup Language) provides various tags for formatting text, which can enhance readability and emphasize content. There are many HTML tags that use in text formatting. Explaining five of them below:

Five Text Formatting Examples:

- **Bold Text ()**:
  - The syntax for making text bold is shown: text , which displays as text in the browser.
  - Example: "This is a **bold** text."
- **Italic Text ()**:
  - The syntax for italic text is: <i> text </i> , which displays as text in the browser.
  - Example: "This is an *italic* text."
- **Underlined Text (<u>) and Highlighted Text (<mark>)**:

  1. **Underlined Text (<u>):**
     - The syntax for underlining text is: <u> text </u> , which displays as <u>text</u> in the browser.
     - Example: "This is an underlined text."

  2. **Highlighted Text (<mark>)**:

     - The syntax for highlighted text is: <mark> text </mark> , which displays as <mark>text</mark> in the browser.
     - Example: "This is a highlighted text."
- **Subscript (<sub>) and Superscript (<sup>)**:
  - Subscript is often used in chemical formulas (e.g., H₂O), and superscript is used for exponents (e.g., 2³).
  - Example: H_2O and 2^3
- **Strikethrough Text (<s>)**:
  - The syntax for strikethrough is <s> text </s> , which displays as <s>text</s> in the browser.
  - Example: "This is a ~~strikethrough~~ text."

Source Code:

```
<!DOCTYPE html>
<html>
<head>

    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Text Formatting Tags</title>
    <style>
        body
        {
            background-color: rgb(133, 183, 207);

        }
    </style>
</head>
<body>
    <h1><u>Text Formatting Tags :</u></h1>

    <h2>1. Bold Text :</h2>
    <p>Syntax: &lt; b &gt; text &lt; / b &gt;</p>
    <p>Example: This is a <b>bold</b> text.</p>

    <h2>2. Italic Text :</h2>
    <p>Syntax: &lt; i &gt; text &lt; / i &gt;</p>
    <p>Example: This is an <i>italic</i> text.</p>

    <h2>3. Underlined Text and Highlighted Text:</h2>
    <p>Syntax:  &lt; u &gt; text &lt; / u &gt;</p>
    <p>Example: This is an <u>underlined</u> text.</p>
    <p>Syntax:  &lt; mark &gt; text &lt; / mark &gt;</p>
    <p>Example: This is a <mark>Highlight</mark> text.</p>

    <h2>4. Subscript Text and Superscript Text :</h2>
    <p>Syntax:  &lt; sub &gt; text &lt; / sub &gt;</p>
    <p>Example: This is H<sub>2</sub>O.</p>
    <p>Syntax:  &lt; sup &gt; text &lt; / sup &gt;</p>
    <p>Example: This is 2<sup>3</sup>.</p>

    <h2>5. Strikethrough Text :</h2>
    <p>Syntax:  &lt; s &gt; text &lt; / s &gt;</p>
    <p>Example: This is a <s>Strikethrough</s> text.</p>

</body>
</html>
```

Output:

Text Formatting Tags :

1. Bold Text :

Syntax: < b > text < / b >

Example: This is a **bold** text.

2. Italic Text :

Syntax: < i > text < / i >

Example: This is an *italic* text.

3. Underlined Text and Highlighted Text:

Syntax: < u > text < / u >

Example: This is an underlined text.

Syntax: < mark > text < / mark >

Example: This is a Highlight text.

4. Subscript Text and Superscript Text :

Syntax: < sub > text < / sub >

Example: This is H_2O.

Syntax: < sup > text < / sup >

Example: This is 2^3.

5. Strikethrough Text :

Syntax: < s > text < / s >

Example: This is a ~~Strikethrough~~ text.

Problem No: 02

Problem Title:

Image in HTML

a. Create an image gallery of any two popular places in Bangladesh with their short description.
b. Demonstrate the use of align, vspace and hspace attribute using minimum of two image.

Part-a: Create an image gallery of any two popular places in Bangladesh with their short description.

Procedure:

To develop an image gallery webpage that displays popular places in Bangladesh using HTML for structure and CSS for styling.

HTML Structure:

HTML (HyperText Markup Language) provides the framework for the webpage. Key elements include:

- **Document Setup**:
    - The webpage starts with <!DOCTYPE html> to declare the document type.
    - The <html> element includes metadata in the <head> section, such as character set and viewport settings for responsive design.
- **Body Content**:
    - A houses all content for proper alignment.
    - An <h1> heading introduces the gallery with the title "Popular Places in Bangladesh".
    - The contains individual items represented by .
- **Gallery Items**:

  Each item includes:

- An <img> tag for the image with an alt attribute for accessibility.
- An <h3> heading for the place name.
- A <p> paragraph for a brief description of the location.

CSS Styling:

CSS (Cascading Style Sheets) is used to enhance the appearance of the webpage. Key styles include:

- **General Styles**:
  - The body has a light gray background and uses a serif font for readability.
- **Container and Gallery Layout**:
  - The .container centers the content and sets a width.
  - The .gallery uses Flexbox to arrange items in a row and allows wrapping for smaller screens.
- **Gallery Item Styles**:
  - Each .gallery-item has a border, padding, and a white background to highlight images and text.
- **Text Styling**:
  - Headings and paragraphs are centered and padded for improved readability.

Source Code:

```
<!DOCTYPE html>
<html lang="en">
<head>
   <meta charset="UTF-8">
   <meta name="viewport" content="width=device-width, initial-scale=1.0">
   <title> Image Gallery</title>
   <style>
      body {
         background-color: #f4f4f4;
         font-family: Georgia, 'Times New Roman', Times, serif;
      }
      .container {
```

```
            width: 50%;
            margin: auto;
            overflow: hidden;

        }
        .gallery {
            display: flex;
            justify-content: space-around;
            flex-wrap: wrap;
            margin: 20px 0;

        }
        .gallery-item {
            flex: 1 1 45%;
            margin: 10px;
            border: 1px solid #ccc;
            border-radius: 5px;
            overflow: hidden;
            background: #fff;

        }
        .gallery-item img {
            width: 100%;
            height: auto;
        }
        .gallery-item h3 {
            text-align: center;
            padding: 10px 0;
        }
        .gallery-item p {
            padding: 0 10px 10px;
            text-align: justify;
        }
    </style>
</head>
<body>
    <div class="container">
        <h1 style="text-align: center;">Popular Places in Bangladesh</h1>
        <div class="gallery">
            <div class="gallery-item">
                <img src="final lab/1609589331085.jpeg" alt="Cox's Bazar">
                <h3>Cox's Bazar</h3>
                <p>Cox's Bazar is known for having the longest natural sea beach in the world, stretching over 120 kilometers. It's a popular tourist destination, famous for its stunning sunsets, sandy shores, and vibrant culture.</p>
            </div>
```

```
        <div class="gallery-item">
            <img src="final lab/istockphoto-1137892510-612x612.jpg" alt="Sundarbans">
            <h3>Sundarbans</h3>
            <p>The Sundarbans is the largest mangrove forest in the world and a UNESCO World Heritage Site. It's renowned for its biodiversity, including the Bengal tiger, and is a vital habitat for many unique species of flora and fauna.</p>
        </div>
    </div>
  </div>
</body>
</html>
```

Output:

Part-b: Demonstrate the use of align, vspace and hspace attribute using minimum of two image.

Procedure:

In this lab, we will explore the use of HTML image attributes, particularly focusing on the align, hspace, and vspace attributes. These attributes help control the placement and spacing of images within a webpage layout.

1. Image Alignment (align):

The align attribute determines the position of an image relative to surrounding text or other elements. Common alignment options include:

- **Left**: The image is aligned to the left, and text wraps around it on the right.
- **Right**: The image is aligned to the right, with text wrapping on the left.
- **Top**: The image is aligned to the top of the surrounding text.
- **Bottom**: The image is aligned to the bottom of the surrounding text.
- **Middle**: The image is vertically centered with the surrounding text.

2. Horizontal Space (hspace)

The hspace attribute specifies the amount of horizontal space (in pixels) to be added on the left and right sides of an image. This attribute helps create breathing space around images, enhancing visual aesthetics.

3. Vertical Space (vspace)

Similarly, the vspace attribute defines the vertical space (in pixels) above and below an image. This is useful for separating images from other content, preventing them from appearing cramped.

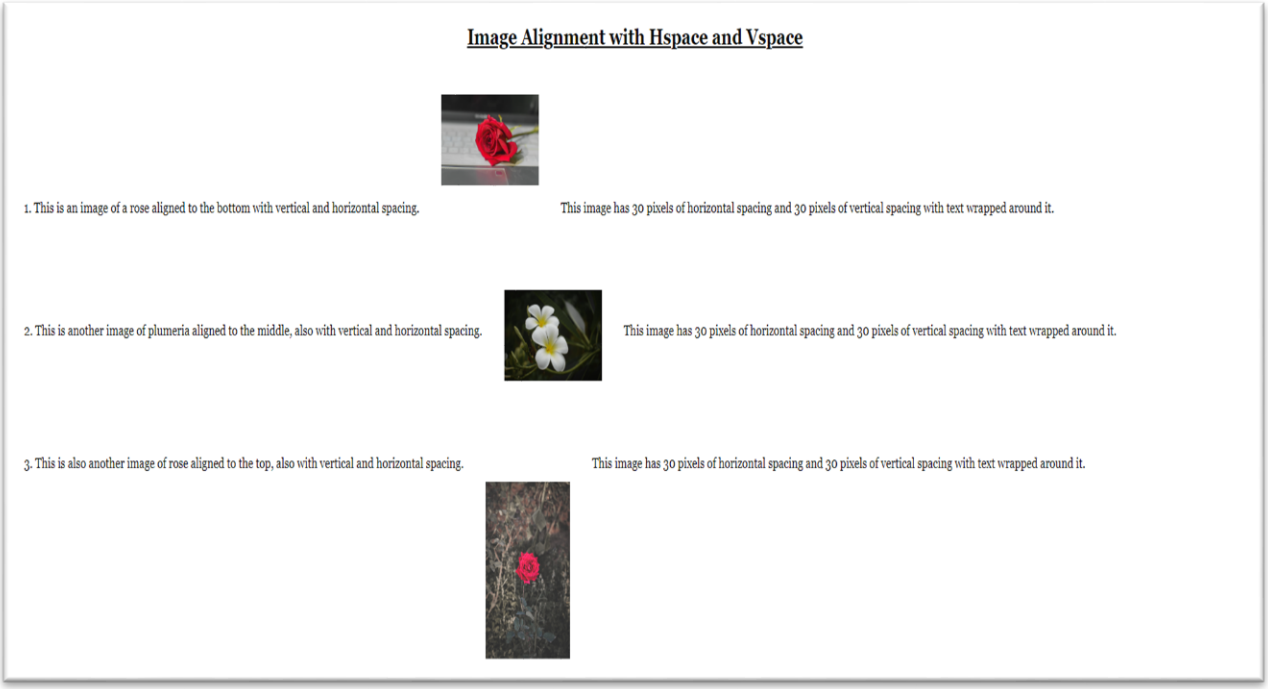
Source Code:

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

```
    <title>Image Alignment, Hspace, and Vspace Example</title>
    <style>
        body {
            font-family: Georgia, 'Times New Roman', Times, serif;
            margin: 20px;
        }
        h2{
            text-align: center;
        }
    </style>
</head>
<body>


    <h2><u>Image Alignment with Hspace and Vspace</u></h2>


    <p>
        1. This is an image of a rose aligned to the bottom with vertical
and horizontal spacing.
        <img src="DSC_0462.jpg" alt="Rose Image" align="bottom"
hspace="30" vspace="30" width="150">
        This image has 30 pixels of horizontal spacing and 30 pixels of
vertical spacing with text wrapped around it.
    </p>

    <br clear="all">


    <p>
        2. This is another image of plumeria aligned to the middle, also
with vertical and horizontal spacing.
        <img src="DSC_1785 (1).jpg" alt="Plumeria Image" align="middle"
hspace="30" vspace="30" width="150">
        This image has 30 pixels of horizontal spacing and 30 pixels of
vertical spacing with text wrapped around it.
    </p>

    <br clear="all">


    <p>
        3. This is also another image of rose aligned to the top, also
with vertical and horizontal spacing.
        <img src="DSC_0516.jpg" alt="Rose Image" align="top" hspace="30"
vspace="30" width="130">
```

```
        This image has 30 pixels of horizontal spacing and 30 pixels of
vertical spacing with text wrapped around it.
    </p>
    <br clear="all">

</body>
</html>
```

Output:



Problem No: 03

Problem Title:

LISTS IN HTML

a.Create the following list using html.

1.BSc

a. Semester-1

    i. ICE-3101
    ii. ICE-3102

b. Semester-2

2.MSc

b.Create the following list using html.

A.Books

i. A brief history of time.
ii. Web Programming.

B.Music

C.Reading

D.Movies

Procedure:

HTML lists are essential for organizing content in a structured manner on web pages. They enhance readability and provide clear formatting for information. There are three primary types of lists in HTML:

- **Ordered Lists (<ol>)**:
    - **Description**: Used for items that follow a specific sequence or rank, such as steps in a process or a list of priorities.
    - **Syntax**: Each item in an ordered list is enclosed within <li> (list item) tags.

- **Example**:

```
<ol>
  <li>Step 1</li>
  <li>Step 2</li>
  <li>Step 3</li>
</ol>
```

- **Customization**: The type attribute can change the numbering style, allowing for alphabetical or Roman numeral sequences (e.g., type="A" for uppercase letters).

- **Unordered Lists (<ul>)**:

  - **Description**: Used for items that do not require a specific order. The items are typically marked with bullets.
  - **Syntax**: Similar to ordered lists, each item is enclosed within <li> tags.
  - **Example**:

```
<ul>
  <li>Item 1</li>
  <li>Item 2</li>
  <li>Item 3</li>
</ul>
```

  - **Customization**: We can alter the bullet style using the list-style-type CSS property, allowing for different bullet shapes like circles or squares.

- **Definition Lists (<dl>)**:

  - **Description**: Used for pairing terms with their definitions, making them ideal for glossaries or dictionaries.
  - **Syntax**: Contains <dt> (definition term) tags for terms and <dd> (definition description) tags for their respective definitions.
  - **Example**:

```
<dl>
  <dt>HTML</dt>
  <dd>HyperText Markup Language</dd>
  <dt>CSS</dt>
  <dd>Cascading Style Sheets</dd>
</dl>
```

Nested Lists:

Nested lists allow for the creation of complex lists within other lists, providing a hierarchical structure. For instance, we can create a nested ordered list inside an unordered list to further categorize items. This technique is useful for organizing related sub-items clearly.

Source Code :

```
<!DOCTYPE html>
<html>
    <head>
        <title>Lists in HTML</title>
    </head>
    <body>

        <h1>    A.Create a nested list for Educational Programs : </h1>
        <ol type="1">
            <li>BSc
                <ol type="a">
                    <li>Semester-1
                        <ul type="square">
                            <li>ICE-3101</li>
                            <li>ICE-3102</li>
                        </ul>
                    </li>
                    <li>Semester-2</li>

                </ol>
            </li>
            <li>MSc</li>
        </ol>


        <h1>    B.Create another nested list for Categories :</h1>
        <ol type="A">
            <li>Books
                <ol type="i">
                    <li>A brief history of time.</li>
                    <li>Web Programming.</li>
                </ol>
            </li>
            <li>Music</li>
            <li>Reading</li>
```

```
            <li>Movies</li>
        </ol>
</body>
</html>
```

Output :

A.Create a nested list for Educational Programs :

1. BSc
    a. Semester-1
        - ICE-3101
        - ICE-3102
    b. Semester-2
2. MSc

B.Create another nested list for Categories :

A. Books
    i. A brief history of time.
    ii. Web Programming.
B. Music
C. Reading
D. Movies

Problem No: 04

Problem Title:

Create an HTML page to display the registration form. (Users Name, Father's Name, Mother's Name, Date of Birth's, Gender, E-mail, Address attributes).

Procedure:

HTML forms are essential components for collecting user input on web pages. They enable interaction between users and web applications, facilitating the gathering of data for various purposes such as registrations, feedback, and surveys. A well-structured form enhances user experience and accessibility.

Key Components of HTML Forms:

1. **Form Tag (<form>):**

- The <form> tag is the container for all form elements. It defines how the data should be sent to the server.
- **Attributes:**
  - action: Specifies the URL where the form data should be sent upon submission.
  - method: Indicates the HTTP method (GET or POST) to be used for sending the data.

2. **Input Fields (<input>):**

- The <input> tag is used to create various types of input fields for user data entry.
- **Types of Input:**
  - text: For short text entries (e.g., user's name).
  - email: For email addresses.
  - date: For date selection.
  - radio: For selecting one option from multiple choices (e.g., gender).
  - textarea: For multi-line text entries (e.g., address).
- Each input should have a name attribute to identify the data sent to the server.

3. **Labels (<label>):**

- The <label> tag provides a user-friendly name for input fields, improving accessibility.
    - It is linked to its corresponding input using the for attribute, which matches the input's id.

4. **Select Dropdowns (<select>):**

- The <select> tag creates a dropdown list for selecting one option from a list of choices.
- Each choice is defined using the <option> tag.

5. **Buttons (<button>):**

- The <button> tag is used for submission and reset actions within forms.
- A button of type submit will send the form data, while a button of type reset will clear the form fields.

Source Code:

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Registration Form</title>
    <style>
        body {
            font-family: Arial, sans-serif;
            background-color: #571e7f;
            display: flex;
            justify-content: center;
            align-items:center;
            height: 100vh;
            margin: 0;
        }
        .form-container {
            background-color: #e5c9dc;
            padding: 30px;
            border-radius: 10px;
```

```
            box-shadow: 0 0 10px rgba(0, 0, 0, 0.896);
            width: 450px;
        }
        .form-container h1 {
            margin-bottom: 30px;
            text-align: center;

        }
        .form-container label {
            font-size: large;
            display: block;
            margin-bottom: 8px;
        }
        .form-container input, .form-container select, .form-container
textarea {
            width: 100%;
            padding: 8px;
            margin-bottom: 15px;

            border-radius: 4px;
            box-shadow: 0 0 5px black;
        }
        .form-container button {
            font-size: large;
            width: 100%;
            padding: 15px;
            background-color: #657f9a;
            color: #fff;
            box-shadow: #657f9a;
            border-radius: 4px;
            cursor: pointer;
        }
        .form-container button:hover {
            background-color: #bf0b1a67;
        }
    </style>
</head>
<body>
    <div class="form-container">
        <h1>Registration Form</h1>
        <form action="#" method="POST">
            <label for="name">Name:</label>
            <input type="text" id="name" name="name" required>

            <label for="father-name">Father's Name:</label>
```

```
            <input type="text" id="father-name" name="father-name"
required>

            <label for="mother-name">Mother's Name:</label>
            <input type="text" id="mother-name" name="mother-name"
required>

            <label for="dob">Date of Birth:</label>
            <input type="date" id="dob" name="dob" required>

            <label for="gender">Gender:</label>
            <select id="gender" name="gender" required>
                <option ><h1>Select Gender</h1></option>
                <option >Male</option>
                <option >Female</option>
                <option >Other</option>
            </select>

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

            <label for="address">Address:</label>
            <textarea id="address" name="address" rows="4"
required></textarea>

            <button type="submit">Register</button>
        </form>
    </div>
</body>
</html>
```

Output:


Registration Form
Name:
Father's Name:
Mother's Name:
Date of Birth:
mm/dd/yyyy
Gender:
Select Gender
Email:
Address:
Register

Problem No: 05

Problem Title:

Create a style sheet named style.css to improve the appearance of your University. Your style sheet should do the following without any modification to your HTML code:

- Change the font properties of at least two elements (such as family, size, weight, style). Here are some standard fonts you may want to use: Arial, Arial Black, Verdana, Trebuchet MS, Georgia, Tahoma, Courier New, Times New Roman.

Procedure:

To create a visually appealing and informative university webpage using HTML and CSS, showcasing the university's mission and information about its community.

HTML Structure

HTML (HyperText Markup Language) provides the structural foundation for the webpage. The main components of the HTML code are:

1. Document Declaration and Metadata

- The document begins with <!DOCTYPE html> to declare the document type.
- The <html> element wraps the entire content, with a lang attribute set to "en" for English.
- The <head> section contains:
  - <meta charset="UTF-8">: Specifies the character encoding for the document.
  - <meta name="viewport" content="width=device-width, initial-scale=1.0">: Ensures the page is responsive on various devices.
  - <title>: Sets the webpage title displayed in the browser tab.
  - <link>: Connects the external CSS file (style.css) for styling.

2. Body Content

The <body> section contains the main content of the webpage:

- **Header**:
  - Uses the <header> tag to contain a welcome message and a motivational quote, which is visually emphasized through heading <h1> and paragraph <p> tags.
- **Section**:
  - The <section> tag introduces the "About Us" section, providing information about the university's dedication to education and innovation.
- **Footer**:
  - The <footer> tag includes copyright information, ensuring the webpage adheres to content rights and ownership.

CSS Styling

CSS (Cascading Style Sheets) is used to enhance the visual presentation of the HTML elements. Key styling components are:

1. General Styles

- The body is styled with:
  - Font family set to 'Verdana' for readability.
  - Line height and margin adjustments for spacing.
  - Background color set to a light gray (#f4f4f4) to contrast with the text.

2. Header Styles

- The header is centered and has:
  - A green background color (#4CAF50) for visual prominence.
  - White text color for contrast.
  - Padding for spacing around the content.

3. Section Styles

- The section element features:
  - A white background for a clean appearance.
  - Box shadows for a subtle three-dimensional effect.
  - Text styling to enhance readability and aesthetic appeal, including the use of different font families.

4. Footer Styles

- The footer has a dark background color (#333) with white text, ensuring it stands out from the rest of the page.

Source Code:

index.html:

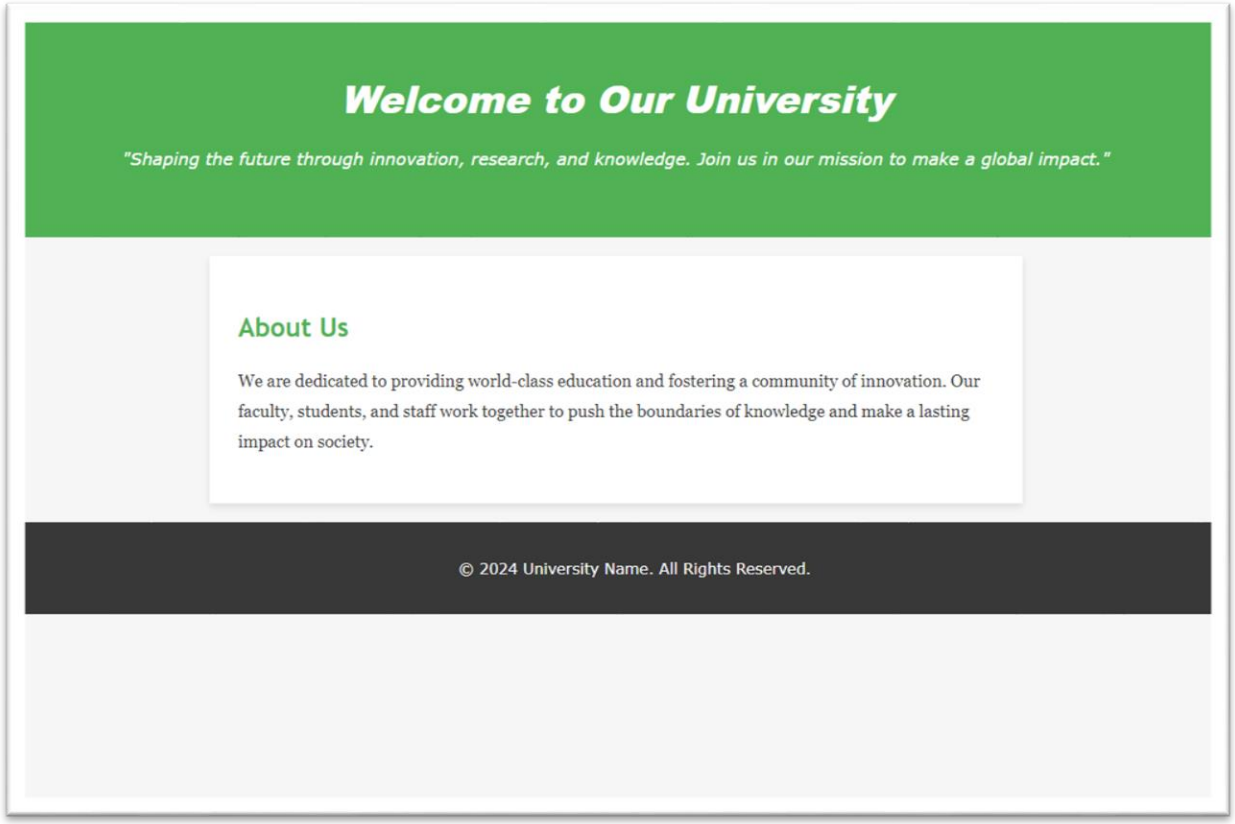
```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>University Page</title>
   <link rel="stylesheet" href="style.css">
</head>
<body>


    <header>
        <h1>Welcome to Our University</h1>
        <p>"Shaping the future through innovation, research, and
knowledge. Join us in our mission to make a global impact."</p>
    </header>


    <section>
        <h2>About Us</h2>
        <p>We are dedicated to providing world-class education and
fostering a community of innovation. Our faculty, students, and staff work
together to push the boundaries of knowledge and make a lasting impact on
society.</p>
    </section>

    <footer>
        <p>&copy; 2024 University Name. All Rights Reserved.</p>
    </footer>

</body>
</html>
```

style.css:

```css
body {
    font-family: 'Verdana', sans-serif;
    line-height: 1.6;
    margin: 0;
    padding: 0;
    background-color: #f4f4f4;
    color: #333;
}


header {
    text-align: center;
    padding: 50px;
    background-color: #4CAF50;
    color: white;
}

header h1 {
    font-family: 'Arial Black', Arial, sans-serif;
    font-size: 40px;
    font-weight: bold;
    font-style: italic;
    margin-bottom: 10px;
}

header p {
    font-size: 18px;
    font-style: italic;
}


section {
    padding: 30px;
    background-color: white;
    margin: 20px auto;
    max-width: 800px;
    box-shadow: 0 4px 8px rgba(0, 0, 0, 0.1);
}

section h2 {
    font-family: 'Trebuchet MS', sans-serif;
    font-size: 28px;
    color: #4CAF50;
    margin-bottom: 15px;
}
```

```css
section p {
    font-family: 'Georgia', Times, serif;
    font-size: 18px;
    line-height: 1.8;
}


footer {
    text-align: center;
    padding: 20px;
    background-color: #333;
    color: white;
    position: relative;
    bottom: 0;
    width: 100%;
}
```

Output:



Problem No: 06

Problem Title:

Create an HTML page to display the registration form that are design an internal style sheet to contain a core set of style used in all pages within this problem. At a minimum define the following CSS properties for elements on your page:

background-color, color, margin-xxx(left, right, top and bottom), padding, border-style, border-color and border-width.

Procedure:

In this lab, we will create a user registration form using HTML and CSS. But here we use CSS as a internal style sheet. We follows these steps:

1. **Set Up the HTML Document:**

- Start with a new HTML file, declaring <!DOCTYPE html> and setting up the <html> element with <head> and <body> sections.
- Include <meta> tags for character set and viewport settings, and set the title of the page.

2. **Create the Form:**

- In the <body>, add an <h1> heading for the form title (e.g., "Registration Form").
- Create a <form> element with the action attribute and method="POST" to collect user data securely.

3. **Input Fields:**

- Add input fields for full name, email, and password within elements labeled with <label> tags for accessibility.

4. **Style the Form:**

- Use internal CSS within a <style> tag in the <head> to enhance the form's appearance. Set background colors, borders, padding, and box shadows.

5. **Submit Button:**

- Include an <input> of type "submit" for form submission and apply styles to improve its appearance and interactivity.

6. **Testing:**

- Open the HTML file in a web browser to verify the form displays correctly and input fields function as intended.

7. **Conclusion:**

- Summarize the lab's outcome, noting the form's usability and potential improvements.

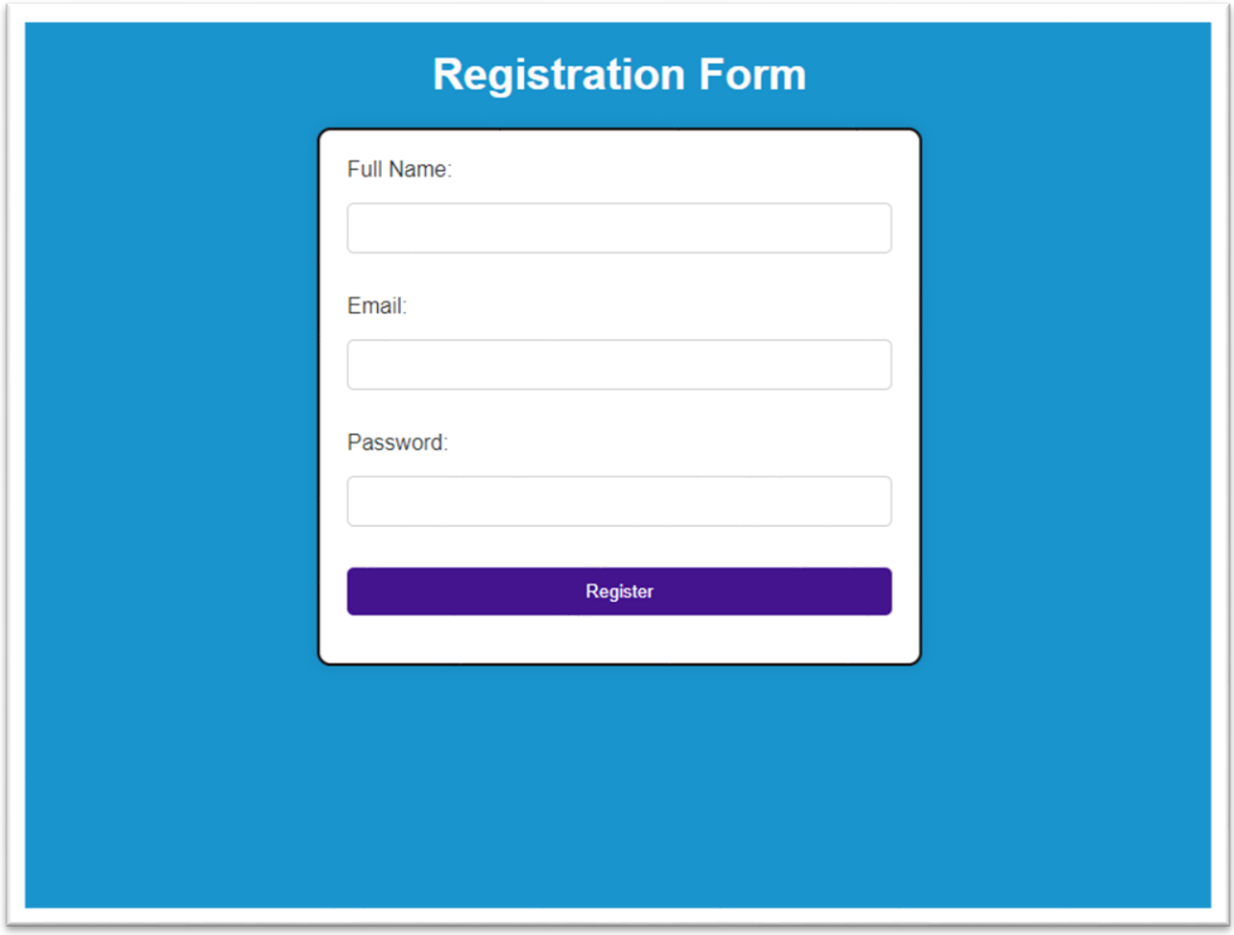
Source Code:

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

        body {
            background-color: #1390ca;
            color: #333;
            font-family: Arial, sans-serif;
            margin: 0;
            padding: 20px;
        }

        h1 {
            text-align: center;
            margin-bottom: 20px;
            color: #fcf9f9;
        }

        form {
            background-color: #ffffff;
            max-width: 400px;
            margin: 0 auto;
            padding: 20px;
            border-style: solid;
            border-color: #0f0101;
            border-width: 2px;
            border-radius: 10px;
            box-shadow: 0px 0px 10px rgba(0, 0, 0, 0.1);
        }
```

```
        label {
            display: block;
            margin-bottom: 10px;
            color: #333;
        }

        input[type="text"],
        input[type="email"],
        input[type="password"] {
            width: 100%;
            padding: 10px;
            margin: 5px 0 15px 0;
            border: 1px solid #ccc;
            border-radius: 5px;
            box-sizing: border-box;
        }

        input[type="submit"] {
            background-color: #3f0d8a;
            color: white;
            padding: 10px;
            border: none;
            border-radius: 5px;
            cursor: pointer;
            width: 100%;
        }

        input[type="submit"]:hover {
            background-color: #45a049;
        }

        .form-group {
            margin-bottom: 15px;
        }
    </style>
</head>
<body>

    <h1>Registration Form</h1>
    <form action="#" method="POST">
        <div class="form-group">
            <label for="name">Full Name:</label>
            <input type="text" id="name" name="name" required>
        </div>

        <div class="form-group">
```

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

        <div class="form-group">
            <label for="password">Password:</label>
            <input type="password" id="password" name="password" required>
        </div>

        <div class="form-group">
            <input type="submit" value="Register">
        </div>
    </form>

</body>
</html>
```

Output:



Problem No: 07

Problem Title:

Create an HTML page to display the registration form that is design an external style sheet to contain a core set of the style used in all pages within this problem. At a minimum define the following CSS properties for elements on your page: padding, border-style and color, text-decoration, text-align, font-size, font-family, font-weight.

Procedure:

In this lab, we will create a user registration form using HTML and CSS by following these steps:

- **HTML Form Creation:**
  - Create a form with input fields for full name, email, and password. Use <label> for each field and make all fields required.
- **Form Layout Styling:**
  - Style the form with a max width of 400px, center it using margin: 20px auto, and apply padding of 20px. Add a light pink background (#fad6eeb5), black border, and rounded corners.
- **Input Field Styling:**
  - Set input fields to 100% width, with padding: 10px, solid black border, and rounded corners.
- **Submit Button Styling:**
  - Style the submit button with a blue background (#3a52a0), white text, and a hover effect that changes the background to green (#45a049).
- **Label and Group Styling:**
  - Style labels with font-size: 16px and add spacing between form sections using the .form-group class.

Source Code:

index.html:

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

    <h1>Registration Form</h1>
    <form action="#" method="POST">
        <div class="form-group">
            <label for="name">Full Name:</label>
            <input type="text" id="name" name="name" required>
        </div>

        <div class="form-group">
            <label for="email">Email:</label>
            <input type="email" id="email" name="email" required>
        </div>

        <div class="form-group">
            <label for="password">Password:</label>
            <input type="password" id="password" name="password" required>
        </div>

        <div class="form-group">
            <input type="submit" value="Register">
        </div>
    </form>

</body>
</html>
```

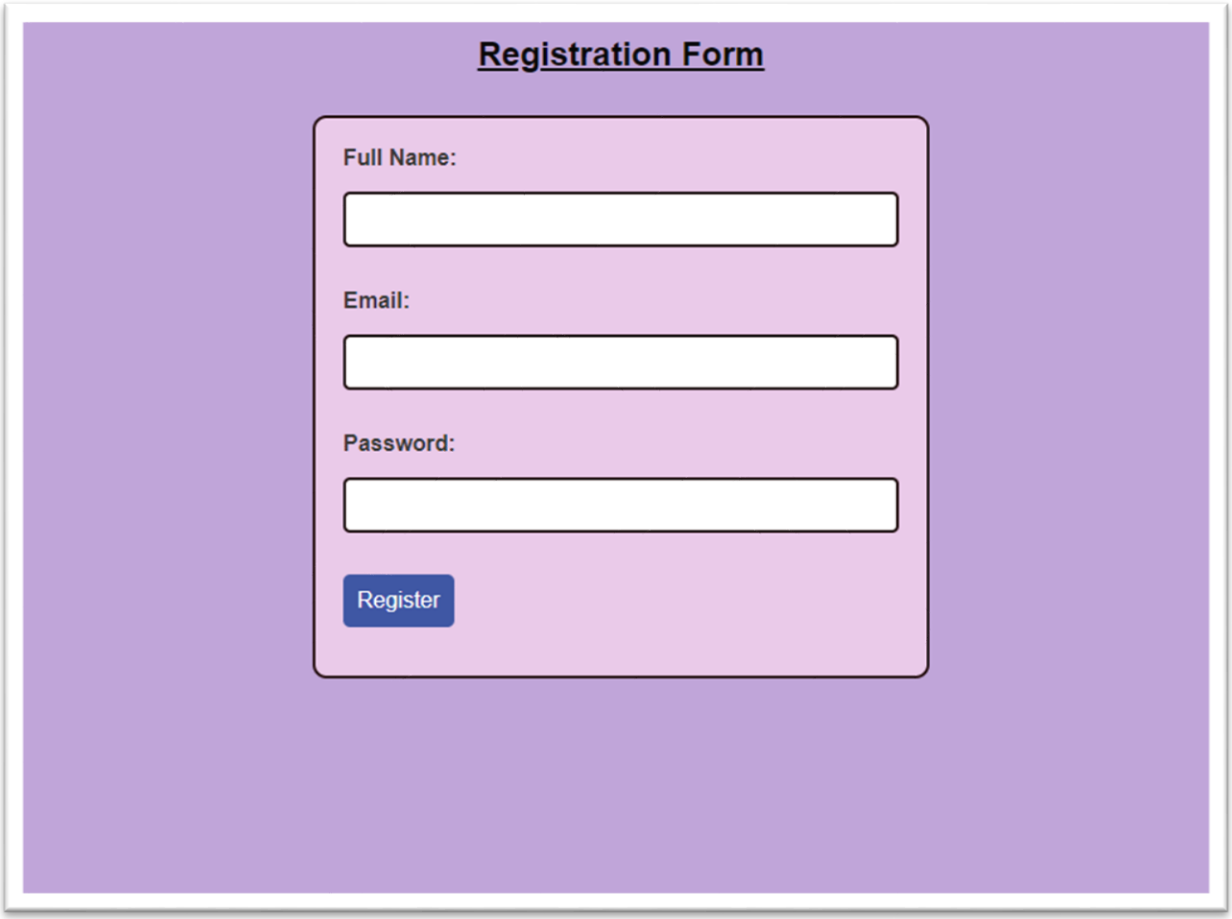
style.css:

```
body {
    font-family: Arial, sans-serif;
    text-align: center;
    background-color: rgb(189, 162, 214);
}
```

```css
h1 {
    font-size: 24px;
    font-weight: bold;
    padding: 10px;
    text-decoration: underline;
}

form {
    background-color: #fad6eeb5;
    max-width: 400px;
    margin: 20px auto;
    padding: 20px;
    border: 2px solid #1f0606;
    border-radius: 10px;
    text-align: left;
}

label {
    display: block;
    margin-bottom: 10px;
    font-size: 16px;
    font-weight: 600;
    color: #333;
}

input[type="text"],
input[type="email"],
input[type="password"] {
    width: 100%;
    padding: 10px;
    margin: 5px 0 15px 0;
    border: 2px solid  #160101;
    border-radius: 5px;
    box-sizing: border-box;
    font-size: 14px;
}

input[type="submit"] {
    background-color: #3a52a0;
    color: white;
    padding: 10px;
    border: 2px rgb(1, 8, 20);
    border-radius: 5px;
    cursor: pointer;
    font-size: 16px;
```

```
}

input[type="submit"]:hover {
    background-color: #45a049;
}

.form-group {
    margin-bottom: 15px;
}
```

Output:



Problem No: 08

Problem Title:

Create an HTML page to display the Bangladesh flag.

Procedure:

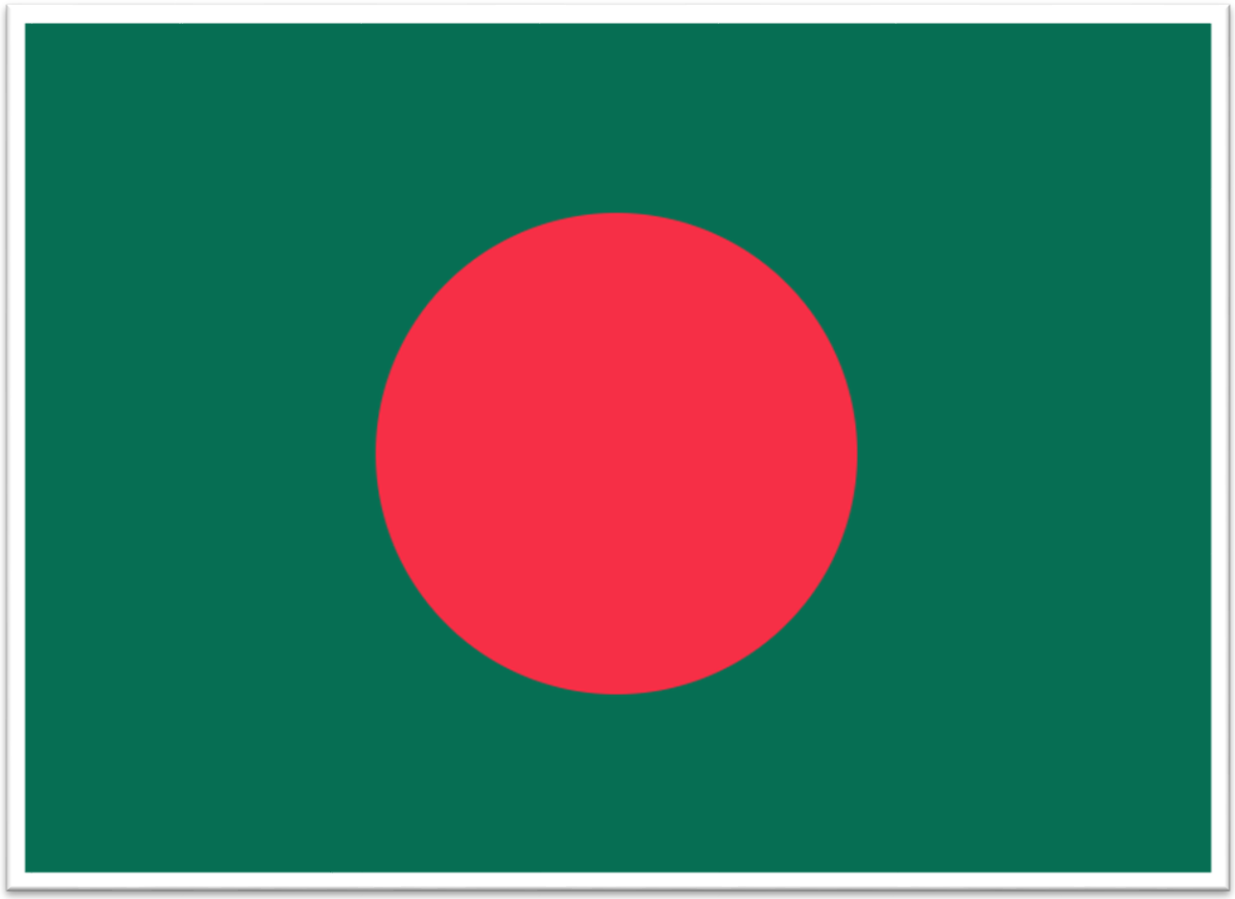
We can build a HTML page to display the Bangladesh flag by following these steps:

- **Define Page Layout:**
  - In the CSS, use display: flex for the body to enable flexbox layout. Apply justify-content: center and align-items: center to vertically and horizontally center the content on the page.
    Set the background color of the body to #006a4e, representing the green field of the Bangladesh flag. Define the height as 100vh to ensure the layout covers the full viewport height.
- **Create the Red Circle:**
  - Add a div element to the HTML document with the class name circle.
  - In the CSS, set the width and height of the .circle to 200 pixels to form a square shape.
  - Set the background color of the .circle to #f42a41 to match the red color of the Bangladesh flag's circle.
  - Use border-radius: 50% to transform the square into a perfect circle, simulating the red disc in the center of the flag.
- **Finalize and View the Output:**
  - Once the styles are applied, the flag representation will consist of a green background with a red circle centered in the middle.

Source Code:

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-
scale=1.0">
    <title>Bangladesh Flag</title>
    <style>
        body {
            display: flex;
            justify-content: center;
            align-items: center;
            height: 100vh;
            margin: 0;
            background-color: #006a4e;
        }

        .circle {
            width: 300px;
            height: 300px;
            background-color: #f42a41;
            border-radius: 50%;
        }
    </style>
</head>
<body>

    <div class="circle"></div>

</body>
</html>
```

Output:



Problem No: 09

Problem Title:

Write a JavaScript program to display the current day and time in the following format. Sample Output:

Today is Tuesday.

The current time is: 09 PM: 15:00

Procedure:

Here's a simplified overview of how the JavaScript program works to display the current day and time:

JavaScript Basics:

JavaScript is used to add interactive elements to web pages. In this case, it used to fetch and display the current day and time on the webpage.

Date Object:

The Date object is a built-in JavaScript object that gives information about the current date and time. When we create a new Date() object, it automatically gives us the current date and time.

- **new Date()**: Creates a new date object with the current date and time.
- **getDay()**: Returns the day of the week as a number (0 for Sunday, 1 for Monday, etc.).
- **getHours()**: Returns the hour (0-23).
- **getMinutes() and getSeconds()**: Return the current minutes and seconds.

Time Formatting:

- **12-Hour Format**: Since the time is returned in 24-hour format, we adjust it to 12-hour format for better readability (e.g., 14:00 becomes 2:00 PM).
- **AM/PM**: To determine whether it's AM or PM, we check if the hour is greater than or equal to 12. If so, it's PM; otherwise, it's AM.

Updating the Time:

The program uses the setInterval() function to update the displayed time every second, ensuring that the time on the webpage stays accurate without needing to refresh the page.

HTML and JavaScript Interaction:

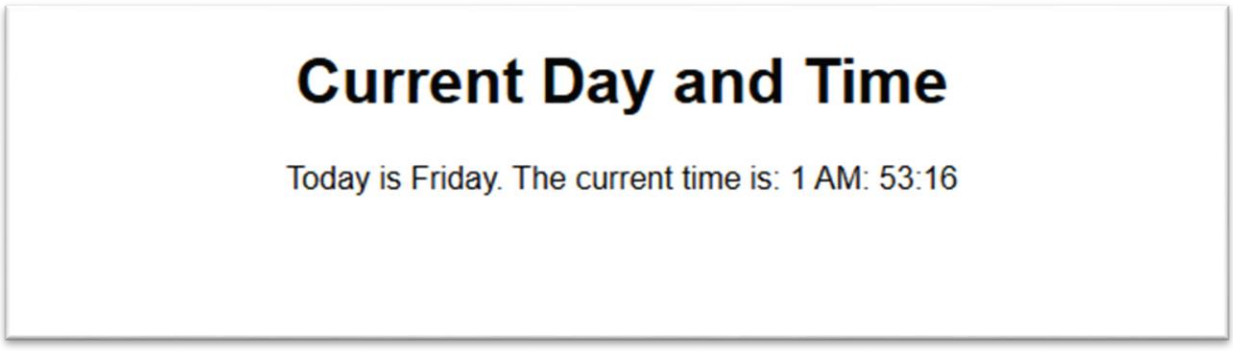
JavaScript interacts with the HTML using document.getElementById() to display the day and time in a specific area of the webpage. The time gets updated every second in the HTML element with the ID time.

Source Code:

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-
scale=1.0">
    <title>Current Day and Time</title>
    <style>
        body {
            font-family: Arial, sans-serif;
            text-align: center;
            margin-top: 20px;
        }
    </style>
</head>
<body>

    <h1>Current Day and Time</h1>
    <p id="time"></p>

    <script>
        function getCurrentDayAndTime() {
            const days = ["Sunday", "Monday", "Tuesday", "Wednesday",
"Thursday", "Friday", "Saturday"];
            const today = new Date();

            // Get the current day
            const day = days[today.getDay()];
```

```
            // Get the current hour
            let hour = today.getHours();
            const minutes = today.getMinutes();
            const seconds = today.getSeconds();
            const ampm = hour >= 12 ? 'PM' : 'AM';

            // Convert 24-hour format to 12-hour format
            hour = hour % 12;
            hour = hour ? hour : 12; // the hour '0' should be '12'

            // Add leading zero to minutes and seconds if needed
            const formattedMinutes = minutes < 10 ? `0${minutes}` :
minutes;
            const formattedSeconds = seconds < 10 ? `0${seconds}` :
seconds;

            // Display the result
            const currentTime = `Today is ${day}. The current time is:
${hour} ${ampm}: ${formattedMinutes}:${formattedSeconds}`;

            document.getElementById('time').innerText = currentTime;
        }

        // Call the function to display the current day and time
        getCurrentDayAndTime();

        // Update the time every second
        setInterval(getCurrentDayAndTime, 1000);
    </script>

</body>
</html>
```

Output:



Problem No: 10

Problem Title:

Write a JavaScript program to find the largest of three given integers.

Procedure:

JavaScript Overview:

JavaScript is widely used to create dynamic and interactive web content. In this program, it is used to compare three integers and determine the largest.

Problem Objective:

The program compares three integers to find the largest among them and displays the result in a readable format.

Comparison Logic:

Using basic comparison operators (>=), the program compares the integers. It checks:

If the first number is greater than or equal to the other two.

If not, it checks the second number.

If neither of the above, the third number is the largest.

Functionality:

The findLargest() function takes three integers and applies conditional statements to identify and return the largest number.

Output:

The result is displayed in the console using console.log() in the format:

"The largest number among num1, num2, and num3 is: largest."

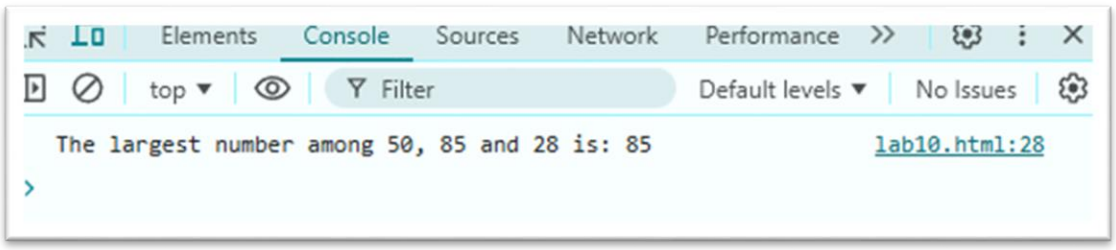
Applications:

This logic can be used in various scenarios, such as finding maximum values in datasets or scores in games.

Source Code:

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

    <h1>Check Console for the Largest Number</h1>

    <script>
        function findLargest(a, b, c) {
            if (a >= b && a >= c) {
                return a;
            } else if (b >= a && b >= c) {
                return b;
            } else {
                return c;
            }
        }

        let num1 = 50;
        let num2 = 85;
        let num3 = 28;

        let largest = findLargest(num1, num2, num3);
        console.log("The largest number among " + num1 + ", " + num2 +
" and " + num3 + " is: " + largest);
    </script>

</body>
</html>
```

Output:



Problem No: 11

Problem Title:

Write a JavaScript program to calculate the multiplication and division of two numbers.

1st Number **:** 12

2nd Number**:** 10

Multiply Divide

The Result Is:

120

Procedure:

The purpose of this program is to take two numbers as input from the user, perform multiplication or division based on the user's choice, and display the result. The user can click on either the **Multiply** or **Divide** button to see the corresponding result.

Key Components:

- **HTML Input Fields**: Two fields allow the user to enter numbers.
- **Buttons**: There are two buttons—one for multiplying the numbers and one for dividing them.
- **Result Display**: The result of the operation is shown in a paragraph element when the user clicks a button.

JavaScript Functions

There are two main functions:

- **multiply()**:
    - Reads the values entered in the input fields.
    - Multiplies the two numbers.

- Displays the result.

- **divide()**:

  - Reads the values entered in the input fields.
  - Divides the first number by the second.
  - If the second number is zero, it shows a message "Cannot divide by zero".
  - Otherwise, it displays the result of the division.

Steps to Use the Program

1. **Enter Two Numbers**: The user enters two numbers in the input fields.
2. **Click a Button**:

   - Click the "Multiply" button to get the multiplication result.
   - Click the "Divide" button to get the division result.

3. **Result Display**: The result is shown in a paragraph below the buttons.

Error Handling

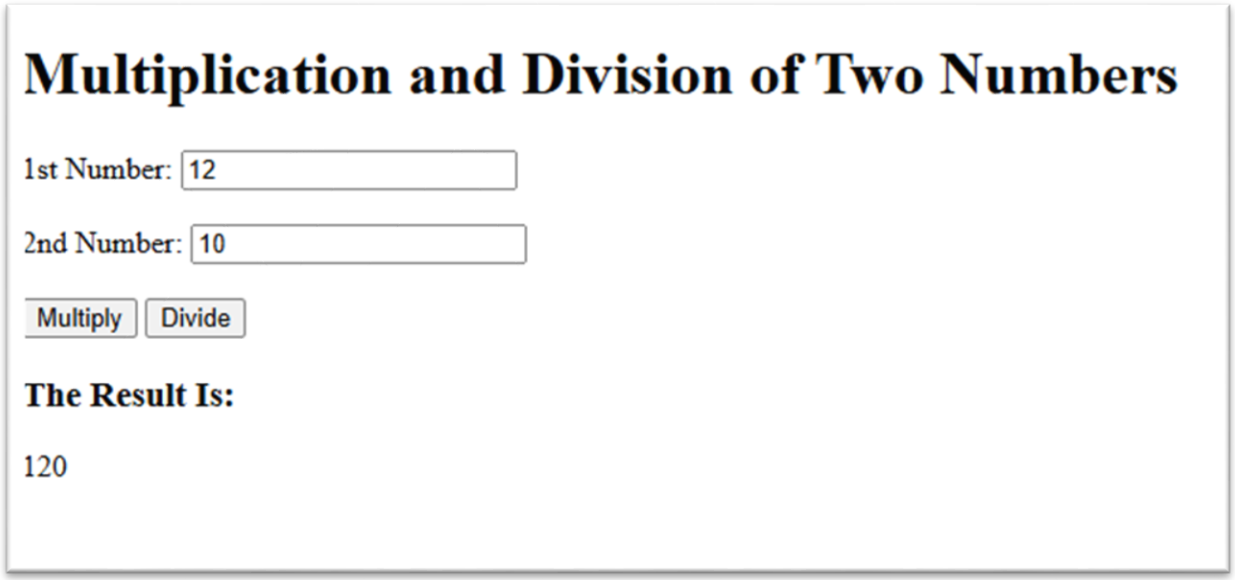
- **Division by Zero**: If the second number is zero and the "Divide" button is clicked, the program will not perform the division. Instead, it will display a message saying "Cannot divide by zero" to avoid an error.

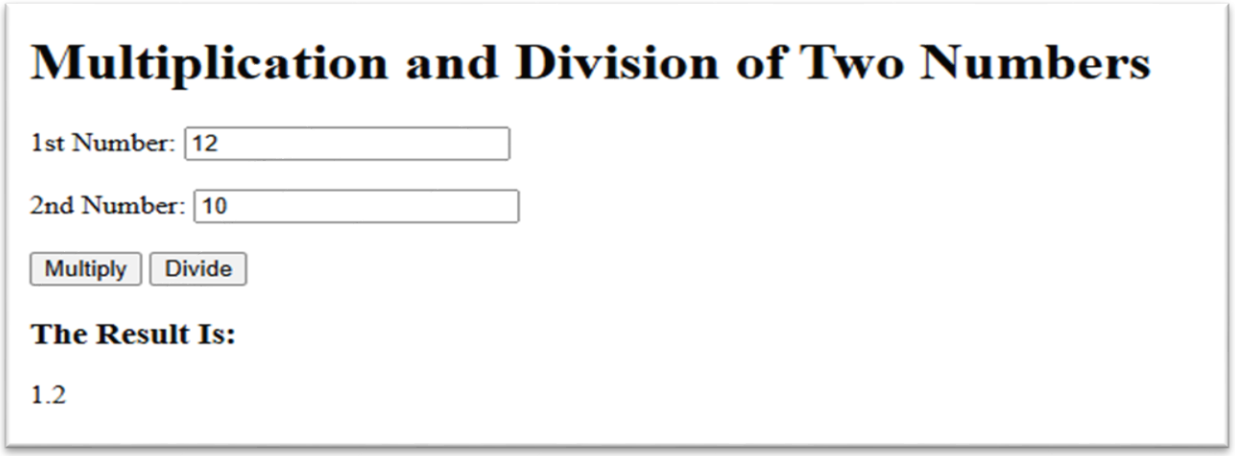
Source Code:

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

    <h1>Multiplication and Division of Two Numbers</h1>


    1st Number: <input type="number" id="num1" value="12"><br><br>
    2nd Number: <input type="number" id="num2" value="10"><br><br>
```

```
    <button onclick="multiply()">Multiply</button>
    <button onclick="divide()">Divide</button>

    <h3>The Result Is:</h3>
    <p id="result"></p>

    <script>

        function multiply() {
            let number1 = document.getElementById('num1').value;
            let number2 = document.getElementById('num2').value;
            let result = number1 * number2;
            document.getElementById('result').innerHTML = result;
        }


        function divide() {
            let number1 = document.getElementById('num1').value;
            let number2 = document.getElementById('num2').value;
            if (number2 == 0) {
                document.getElementById('result').innerHTML = "Cannot
divide by zero!";
            } else {
                let result = number1 / number2;
                document.getElementById('result').innerHTML = result;
            }
        }
    </script>

</body>
</html>
```

Output:


Multiplication and Division of Two Numbers
1st Number: 12
2nd Number: 10
Multiply
Divide
The Result Is:
120


Multiplication and Division of Two Numbers
1st Number: 12
2nd Number: 10
Multiply
Divide
The Result Is:
1.2

Problem No: 12

Problem Title:

Write a JavaScript program to draw two intersecting rectangles, one of which has alpha transparency.

Procedure:

In web development, drawing shapes like rectangles can be easily achieved using HTML5's <canvas> element and JavaScript. The <canvas> element provides a surface where 2D graphics can be rendered, including shapes, images, and even animations. In this example, we will demonstrate how to draw two intersecting rectangles—one with a solid color and the other with transparency (alpha transparency).

The <canvas> Element:

The <canvas> element is used to define an area on the webpage where graphics can be drawn using JavaScript. It's a powerful tool for rendering 2D graphics like shapes, colors, and transparency effects.

Key Attributes:

- **Width and Height**: These attributes define the size of the canvas on which you will draw.
- **Context**: The getContext('2d') method is used to retrieve the 2D drawing context, which provides methods for rendering shapes.

JavaScript Functions for Drawing Rectangles:

1. **getContext('2d')**: This method gives access to the canvas's 2D drawing context, allowing you to draw shapes, set colors, and apply styles.
2. **fillStyle**: This property is used to specify the fill color or style for shapes. It can be a solid color (e.g., hex or RGB) or a color with transparency using RGBA values.
3. **fillRect(x, y, width, height)**: This function draws a filled rectangle on the canvas at the specified position (x, y) and with the given width and height.

Steps to Draw Intersecting Rectangles:

1. Setting Up the Canvas

We define the canvas in the HTML document using the <canvas> tag and specify its width and height. Then, using JavaScript, we retrieve the 2D context for drawing.

2. Drawing the First Rectangle (Solid Navy Blue)

The first rectangle is filled with a solid navy blue color (#000080). We use the fillStyle property to set the color and fillRect() to draw the rectangle.

3. Drawing the Second Rectangle (Blue with Transparency)

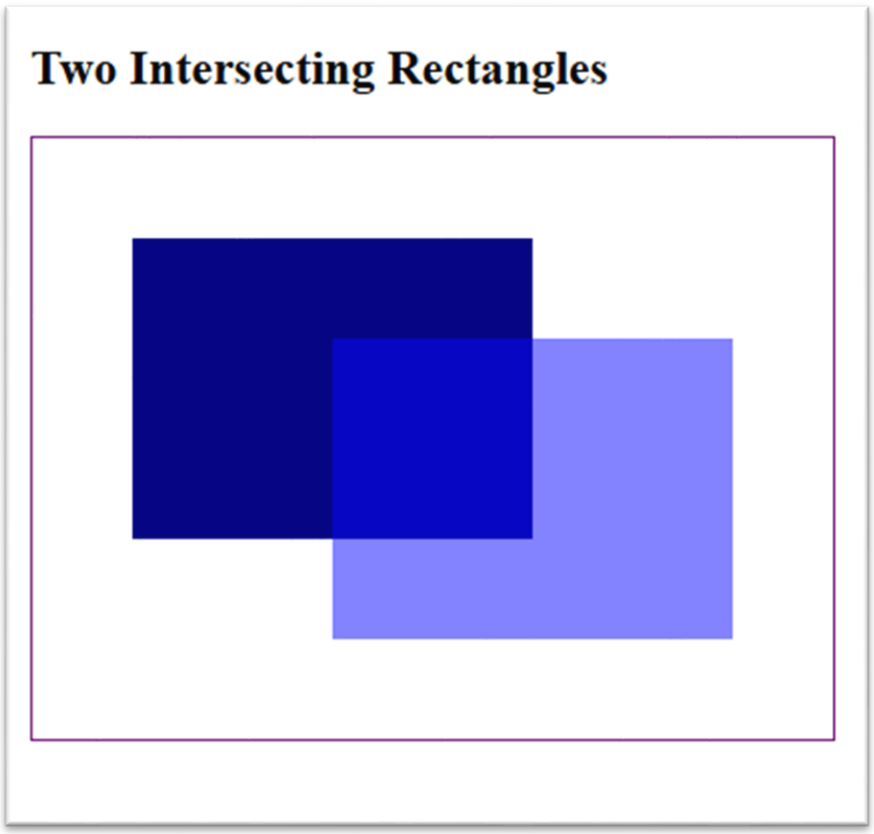
The second rectangle is drawn with a blue color and 50% transparency (rgba(0, 0, 255, 0.5)). This rectangle will overlap the first one, allowing part of the navy blue rectangle to be visible through the transparent blue.

Source Code:

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

    <h2>Two Intersecting Rectangles</h2>

    <canvas id="myCanvas" width="400" height="300" style="border:1px
solid #68066b;"></canvas>

    <script>
        const canvas = document.getElementById('myCanvas');
        const context = canvas.getContext('2d');


        context.fillStyle = "#000080";
        context.fillRect(50, 50, 200, 150);


        context.fillStyle = "rgba(0, 0, 255, 0.5)";
        context.fillRect(150, 100, 200, 150);
```

```
    </script>

</body>
</html>
```

Output:



Problem No: 13

Problem Title:

Write a program to create a chess board in PHP using a for loop.

Procedure:

PHP is a server-side scripting language that is embedded into HTML to generate dynamic web content. In this program, PHP is used to dynamically create an 8x8 chessboard by generating the HTML table structure.

- **HTML Table Structure**: An HTML table is used to represent the chessboard with rows (<tr>) and cells (<td>). Each row contains 8 cells, and there are 8 rows, forming the grid.
- **PHP for Dynamic Table Generation**: Two nested loops in PHP:

  Outer Loop: Creates each row.

  Inner Loop: Creates each cell in the row.

- **Alternating Cell Colors**: The sum of row and column indices ($row + $col) determines whether a cell is black or white:

  Even sum assigns the 'white' class.

  Odd sum assigns the 'black' class.

- **CSS for Chessboard Styling**:

  black and .white classes define the colors.

  Table: The chessboard has a fixed size (400x300px) with border-collapse for seamless cell borders.

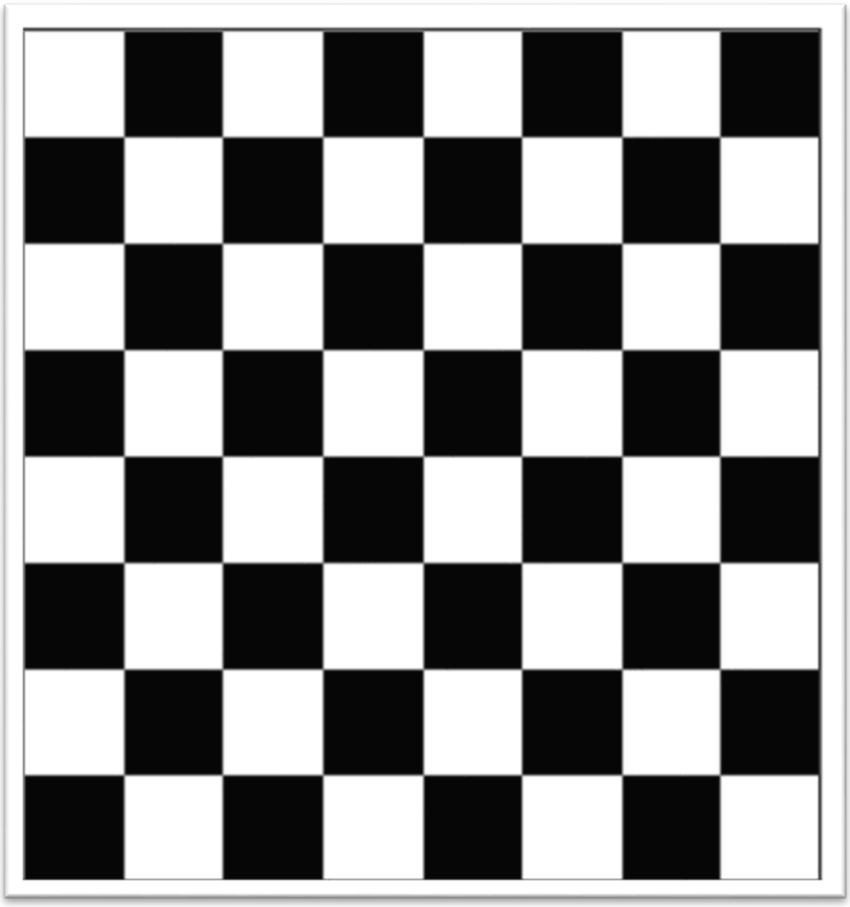
  Cells (<td>): Each cell is 50x50px for uniformity.

- **Flow of Execution**: PHP generates the HTML structure on the server, which is then rendered by the browser with the appropriate CSS styles for black and white squares.
- **Final Output**: An 8x8 chessboard displayed in the browser, with alternating black and white squares generated dynamically using PHP.

Source Code:

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-
scale=1.0">
    <title>Chessboard in PHP</title>
    <style>
        table {
            border-collapse: collapse;
            width: 400px;
            height: 300px;
        }
        td {
            width: 50px;
            height: 50px;
        }
        .black {
            background-color: black;
        }
        .white {
            background-color: white;
        }
    </style>
</head>
<body>

<table border="2">
    <?php
    for ($row = 0; $row < 8; $row++) {
        echo "<tr>";
        for ($col = 0; $col < 8; $col++) {

            $class = ($row + $col) % 2 == 0 ? 'white' : 'black';
            echo "<td class='$class'></td>";
        }
        echo "</tr>";
    }
    ?>
</table>

</body>
</html>
```

Output:



Problem No: 14

Problem Title:

Create a simple HTML form and accept the username and display the name through a PHP echo statement.

Procedure:

This application demonstrates how to accept user input through an HTML form and process it using PHP to provide a personalized response.

1. **HTML Form**:

- **Form Structure**: Utilizes <form> to encapsulate input elements, including a text field for username and a submit button.
- **Input Elements**:
    - <label> for accessibility.
    - <input type="text"> for username input.
    - <input type="submit"> for form submission.

2. **PHP Processing**:

- **Request Handling**: Checks if the form is submitted via POST using $_SERVER["REQUEST_METHOD"].
- **Data Retrieval and Sanitization**: Retrieves the username with $_POST, sanitizes it using htmlspecialchars() to prevent XSS.
- **Dynamic Output**: Displays a greeting using echo.

Data Flow:

1. **User Interaction**: The user enters their username in the form and submits it.
2. **Form Submission**: A POST request is sent to the server.
3. **PHP Execution**: The server processes the input and returns a greeting.
4. **Dynamic Response**: The greeting is displayed to the user.

Security Considerations:

- **Input Validation**: Always sanitize user inputs.
- **Use HTTPS**: Ensure data is transmitted securely.
- **Error Handling**: Implement measures to manage errors.

Source Code:

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-
scale=1.0">
    <title>Username Form</title>
</head>
<body>
    <h1>Enter Username :</h1>

    <?php
    if ($_SERVER["REQUEST_METHOD"] == "POST") {

        $username = htmlspecialchars($_POST['username']);
        echo "<h2>Hello, $username!</h2>";
    } else {
        ?>
        <form action="" method="POST">
            <label for="username">Username:</label>
            <input type="text" id="username" name="username" required>
            <input type="submit" value="Submit">
        </form>
        <?php
    }
    ?>
</body>
</html>
```

Output:


Enter Username :
Username: SABRINA AHMED
Submit


Enter Username :
Hello, SABRINA AHMED!

Problem No: 15

Problem Title:

Write a PHP script to display string , values within a table.

Procedure:

PHP (Hypertext Preprocessor) is a server-side scripting language commonly used for web development. It allows dynamic content generation, meaning web pages can be modified in real-time based on user input or database queries.

HTML (Hypertext Markup Language) is the standard language for creating web pages, providing the structure and layout of the content displayed in browsers. This script combines PHP and HTML to create a dynamic webpage displaying a list of books in a structured format.

The primary goal of the script is to present a list of books along with their details (title, author, price, and quantity) in a visually appealing HTML table. This enhances user readability and interaction with the data.

Components:

- **HTML Structure:**
  - The script begins with a <!DOCTYPE html> declaration, indicating HTML5. The <head> section contains metadata and a <title>, along with a <style> block for CSS styling.
- **CSS Styles:**
  - Styles are applied to format the body, table, headers, and rows:
    - The body features a simple font and background.
    - The table is styled for width, borders, and spacing.
    - Alternating row colors enhance readability, and hover effects provide user interaction feedback.

PHP Functionality:

- **Data Definition:**
  - A string variable ($titleString) holds the title for the book list.
  - An associative array ($books) contains details of each book.
- **Dynamic Content Generation:**
  - The script uses echo to output HTML content. It generates the main title and constructs a table with a header row.

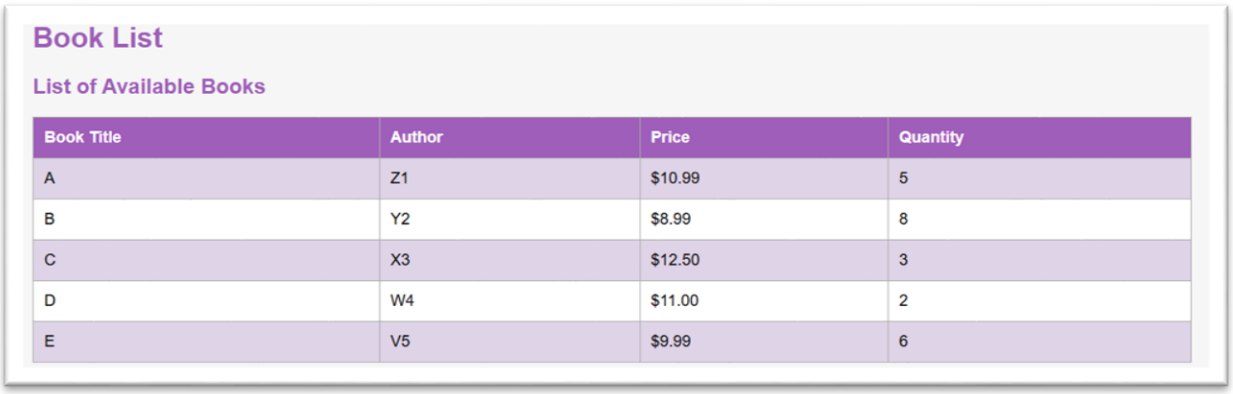
- A foreach loop iterates over the $books array, creating a new table row for each book's details.

Source Code:

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-
scale=1.0">
    <title>Display Books in a Table</title>
    <style>
        body {
            font-family: Arial, sans-serif;
            background-color: #f4f4f4;
            padding: 20px;
        }
        table {
            width: 70%;
            border-collapse: collapse;
            margin: 20px 0;
            font-size: 18px;
            text-align: left;
        }
        th, td {
            padding: 12px;
            border: 1px solid #aaa;
        }
        th {
            background-color: #9b59b6;
            color: white;
        }
        tr:nth-child(even) {
            background-color: #dcd0e5;
        }
        tr:nth-child(odd) {
            background-color: #ffffff;
        }
        tr:hover {
            background-color: #b19cd9;
        }
    </style>
</head>
<body>
```

```
<h1 style="color: #9b59b6;">Book List</h1>

<?php
$titleString = "List of Available Books";

$books = [
    ["title" => "A", "author" => "Z1", "price" => "$10.99", "quantity"
=> 5],
    ["title" => "B", "author" => "Y2", "price" => "$8.99", "quantity"
=> 8],
    ["title" => "C", "author" => "X3", "price" => "$12.50", "quantity"
=> 3],
    ["title" => "D", "author" => "W4", "price" => "$11.00", "quantity"
=> 2],
    ["title" => "E", "author" => "V5", "price" => "$9.99", "quantity"
=> 6]
];

echo "<h2 style='color: #9b59b6;'>$titleString</h2>";

echo "<table>";
echo "<tr><th>Book
Title</th><th>Author</th><th>Price</th><th>Quantity</th></tr>";

foreach ($books as $book) {
    echo
"<tr><td>{$book['title']}</td><td>{$book['author']}</td><td>{$book['pri
ce']}</td><td>{$book['quantity']}</td></tr>";
}

echo "</table>";
?>

</body>
</html>
```

Output:



Problem No: 16

Problem Title:

Create a simple 'birthday countdown' script, the script will count the number of days between current day and birthday.

Procedure:

This PHP script provides an interactive way for users to input their birthday and receive a countdown of the days remaining until that date.

To allow users to enter their birthday and display the number of days remaining until that birthday, including handling cases where the birthday is today or has already passed.

Components:

- **HTML Structure:**
    - The script uses standard HTML to create the layout, including a <head> section for metadata and styling, and a <body> section for content display.
- **CSS Styles:**
    - Basic CSS styling enhances the appearance of the page, including font choices, colors, and layout adjustments for better user experience.

User Input:

- An HTML form is included with:
    - An input field of type date that allows users to select their birthday in the YYYY-MM-DD format.
    - A submit button that sends the input to the server for processing.

PHP Functionality:

- When the form is submitted, the script checks if the request method is POST to determine if the data has been sent.
- The birthday is retrieved from the submitted form.

- The current date is obtained using date("Y-m-d").
- The difference in days between the current date and the birthday is calculated using strtotime() to convert the dates into Unix timestamps, followed by arithmetic to find the difference in days.
- The script uses conditional statements to check if the birthday is today, has passed, or is upcoming, and then displays the appropriate message.

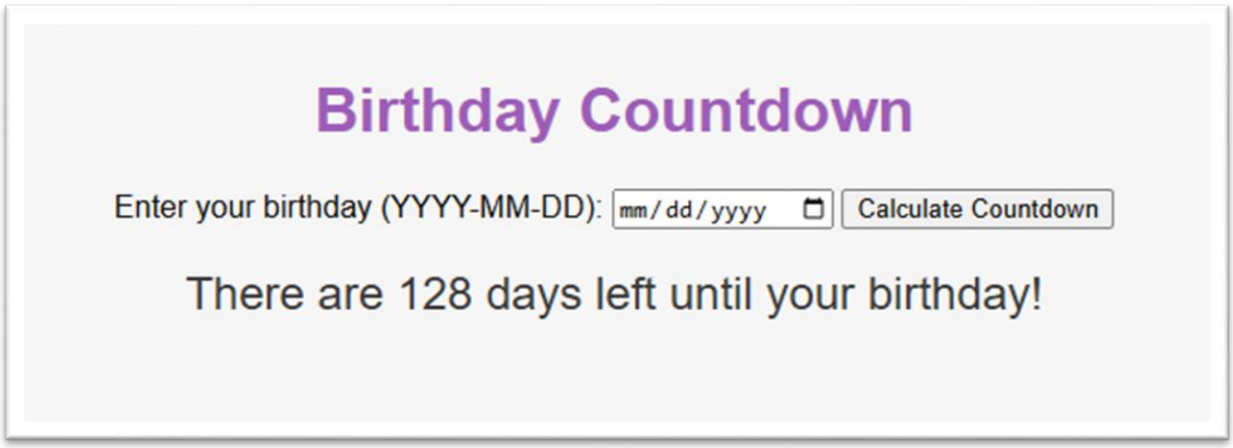
Source Code:

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-
scale=1.0">
    <title>Birthday Countdown</title>
    <style>
        body {
            font-family: Arial, sans-serif;
            background-color: #f4f4f4;
            padding: 20px;
            text-align: center;
        }
        h1 {
            color: #9b59b6;
        }
        .countdown {
            font-size: 24px;
            color: #333;
        }
        form {
            margin: 20px 0;
        }
    </style>
</head>
<body>

<h1>Birthday Countdown</h1>

<form method="POST">
    <label for="birthday">Enter your birthday (YYYY-MM-DD): </label>
    <input type="date" id="birthday" name="birthday" required>
    <input type="submit" value="Calculate Countdown">
```

```
</form>

<?php
if ($_SERVER["REQUEST_METHOD"] == "POST") {
    $birthday = $_POST['birthday'];
    $currentDate = date("Y-m-d");
    $diff = (strtotime($birthday) - strtotime($currentDate)) / (60 * 60
* 24);
    $diff = ceil($diff);

    if ($diff < 0) {
        echo "<div class='countdown'>Your birthday has already passed
this year!</div>";
    } elseif ($diff == 0) {
        echo "<div class='countdown'>Happy Birthday! Today is your
birthday!</div>";
    } else {
        echo "<div class='countdown'>There are $diff days left until
your birthday!</div>";
    }
}
?>

</body>
</html>
```

Output:


Birthday Countdown
Enter your birthday (YYYY-MM-DD):
mm/dd/yyyy
Calculate Countdown


Birthday Countdown
Enter your birthday (YYYY-MM-DD):
mm/dd/yyyy
Calculate Countdown
There are 128 days left until your birthday!

Problem No: 17

Problem Title:

Create a PHP registration form and submit the data to the database and retrieve the data from database and show details in browser.

Procedure:

This project is a simple user registration system with an HTML form and PHP code for data processing, which enables users to register and view registered user information stored in a MySQL database.

The system is implemented in a single PHP-embedded HTML file, organized into three main functional sections:

- **HTML Registration Form**

  Purpose: Provides an interface where users can input their information (name, email, and password) for registration.

  Features:

  - Uses an HTML <form> element to collect user inputs.
  - Fields include name, email, and password, each required and validated on the client side with HTML5 attributes.
  - The form submits data using the POST method to ensure data is sent securely to the server.
  - An input button labeled "Register" triggers the data submission process.

- **PHP Data Processing**

  Purpose: Handles the data submission and storage of user information in a MySQL database.

  Mechanics:

  - The script connects to a MySQL database (registration_db) using mysqli.
  - Upon form submission, PHP retrieves and sanitizes user inputs (name, email, and password).
  - Passwords are hashed using password_hash to enhance security by storing encrypted passwords.

- An SQL INSERT query adds the sanitized data into a users table in the database, containing columns for name, email, and password.
- If the registration is successful, a confirmation message displays; otherwise, an error message is shown, indicating issues with the SQL query.

- **User Data Retrieval and Display**

  Purpose: Retrieves and displays all registered users for easy viewing.

  Mechanics:

  - After registration, an SQL SELECT query fetches all records from the users table.
  - Each user's ID, name, and email are displayed in an HTML list format, providing a simple overview of all registered users.
  - If no records are found, a message notifies that no registered users exist.

Source Code:

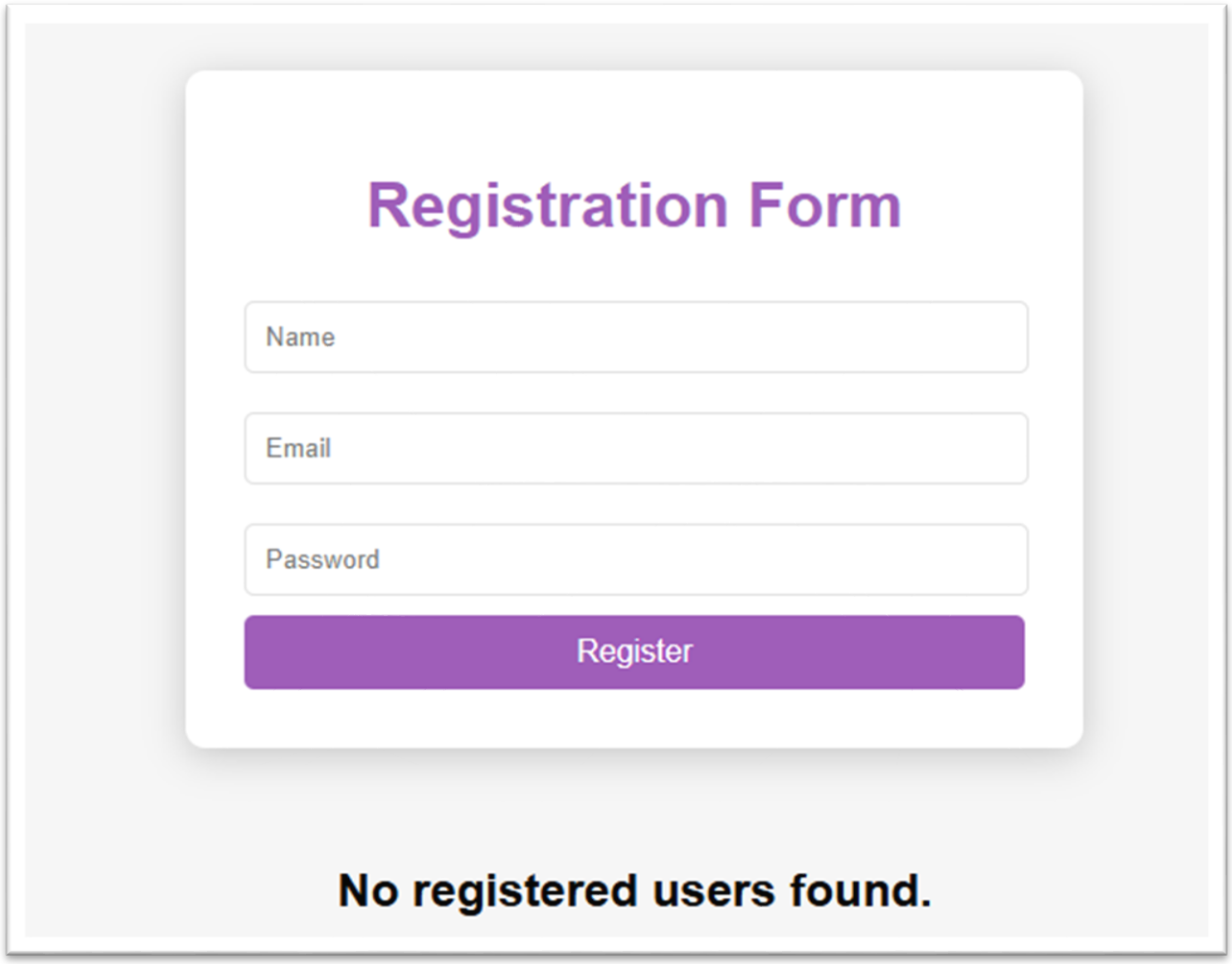
```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-
scale=1.0">
    <title>Registration Form</title>
    <style>
        body {
            font-family: Arial, sans-serif;
            background-color: #f4f4f4;
            padding: 0;
            margin: 0;
            display: flex;
            justify-content: center;
            align-items: center;
            height: 100vh;
            flex-direction: column;
        }
        .container {
            max-width: 400px;
```

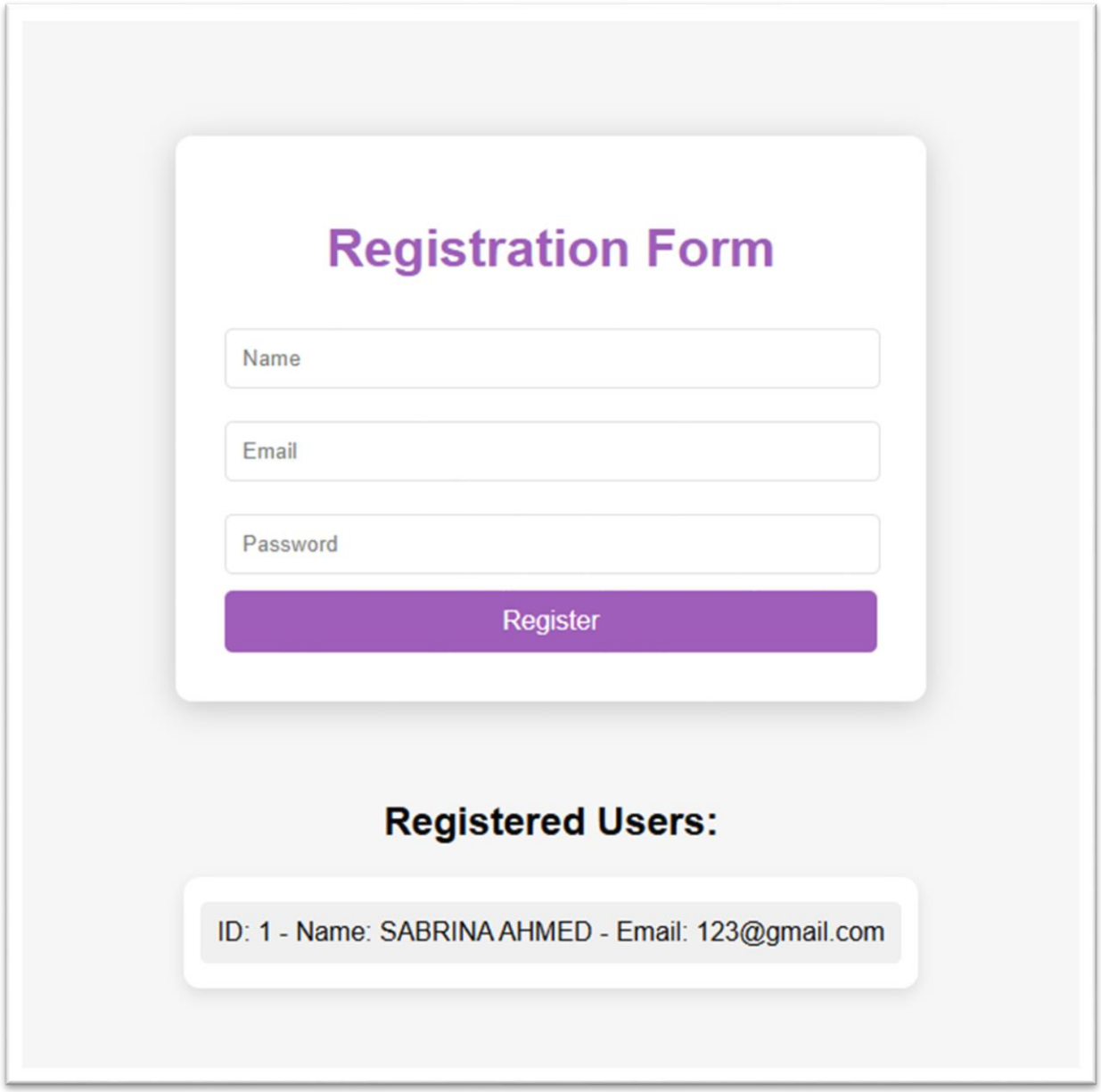
```
            width: 100%;
            background: white;
            border-radius: 10px;
            box-shadow: 0 4px 20px rgba(0, 0, 0, 0.2);
            padding: 30px;
            text-align: center;
        }
        h1 {
            color: #9b59b6;
            margin-bottom: 20px;
        }
        input[type="text"], input[type="email"], input[type="password"]
{
            width: calc(100% - 20px);
            padding: 10px;
            margin: 10px 0;
            border: 1px solid #ddd;
            border-radius: 5px;
        }
        input[type="submit"] {
            background-color: #9b59b6;
            color: white;
            border: none;
            padding: 10px;
            border-radius: 5px;
            cursor: pointer;
            width: 100%;
            font-size: 16px;
        }
        input[type="submit"]:hover {
            background-color: #8e44ad;
        }
        .user-list {
            margin-top: 40px; /* Increased distance */
            text-align: center;
        }
        ul {
            list-style-type: none;
            padding: 0;
            margin: 10px 0 0;
            background: white;
            border-radius: 10px;
            box-shadow: 0 4px 10px rgba(0, 0, 0, 0.1);
            padding: 10px;
        }
        li {
```

```
            background-color: #eee;
            margin: 5px 0;
            padding: 10px;
            border-radius: 5px;
        }
    </style>
</head>
<body>

<div class="container">
    <h1>Registration Form</h1>

    <form method="POST" action="">
        <input type="text" name="name" placeholder="Name" required>
        <input type="email" name="email" placeholder="Email" required>
        <input type="password" name="password" placeholder="Password"
required>
        <input type="submit" name="submit" value="Register">
    </form>
</div>

<div class="user-list">
    <?php
    $servername = "localhost";
    $username = "root";
    $password = "";
    $dbname = "registration_db";

    $conn = new mysqli($servername, $username, $password, $dbname);

    if ($conn->connect_error) {
        die("Connection failed: " . $conn->connect_error);
    }

    if (isset($_POST['submit'])) {
        $name = $_POST['name'];
        $email = $_POST['email'];
        $password = password_hash($_POST['password'],
PASSWORD_DEFAULT);

        $sql = "INSERT INTO users (name, email, password) VALUES
('$name', '$email', '$password')";

        if ($conn->query($sql) === TRUE) {
            echo "<h2>Registration successful!</h2>";
        } else {
```

```
            echo "Error: " . $sql . "<br>" . $conn->error;
        }
    }

    $sql = "SELECT id, name, email FROM users";
    $result = $conn->query($sql);

    if ($result->num_rows > 0) {
        echo "<h2>Registered Users:</h2><ul>";
        while ($row = $result->fetch_assoc()) {
            echo "<li>ID: " . $row["id"] . " - Name: " . $row["name"] .
" - Email: " . $row["email"] . "</li>";
        }
        echo "</ul>";
    } else {
        echo "<h2>No registered users found.</h2>";
    }

    $conn->close();
    ?>
</div>

</body>
</html>
```

Output:


Registration Form
Name
Email
Password
Register
No registered users found.


Registration Form
Name
Email
Password
Register
Registered Users:
ID: 1 - Name: SABRINA AHMED - Email: 123@gmail.com