BCS/16/21/005/TZ - ABDILLAH TALIB ALI

ASSIGNMENT 1

**Part 1: HTML Structure and Tags**

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <title>My Web Page</title>

    <link rel="icon" href="favicon.ico" type="image/x-icon">

</head>

<body>

    <h1>Welcome to My Web Page</h1>

    <img src="me.jpg" alt="A photo of me" width="200">

    <h2>This is an H2 Heading</h2>

    <h3>This is an H3 Heading</h3>

    <h4>This is an H4 Heading</h4>

    <h5>This is an H5 Heading</h5>

    <h6>This is an H6 Heading</h6>

    <p>I am a passionate learner and aspiring tech professional who enjoys exploring books, playing football, coding, and expanding knowledge through university studies and self-improvement.</p>

    <hr>

    <p>This is a second paragraph.<br>

        And this line starts after a line break.</p>

    <h2>My Top 5 Favorite Books</h2>

    <ol>

        <li>The Alchemist</li>

        <li>Atomic Habits</li>

        <li>Sapiens</li>

        <li>1984</li>

        <li>Clean Code</li>

    </ol>

    <h2>My Top 5 Hobbies</h2>

    <ul>

        <li>Reading</li>

        <li>Photography</li>

        <li>Traveling</li>

        <li>Playing Guitar</li>

        <li>Learning New Skills</li>

    </ul>

    <h2>My Favorite Image</h2>

<img src="https://images.pexels.com/photos/4974912/pexels-photo-4974912.jpeg" alt="Person coding on a MacBook Pro at desk" width="250">

    <h2>Useful Links</h2>

    <p>

        Visit <a href="https://www.wikipedia.org" target="\_blank">Wikipedia</a><br>

        Watch videos on <a href="https://www.youtube.com" target="\_blank">YouTube</a><br>

        Learn code at <a href="https://www.freecodecamp.org" target="\_blank">freeCodeCamp</a>

    </p>

    <h2>My Weekly Schedule (Mon–Fri)</h2>

    <table border="1" cellpadding="8" cellspacing="0">

        <tr>

            <th>Day</th>

            <th>Morning</th>

            <th>Afternoon</th>

        </tr>

        <tr>

            <td>Monday</td>

            <td>University</td>

            <td>Studying, Coding</td>

        </tr>

        <tr>

            <td>Wednesday</td>

            <td>University</td>

            <td>Football, Coding</td>

        </tr>

        <tr>

            <td>Friday</td>

            <td>Studying</td>

            <td>Playing Football, Learning</td>

        </tr>

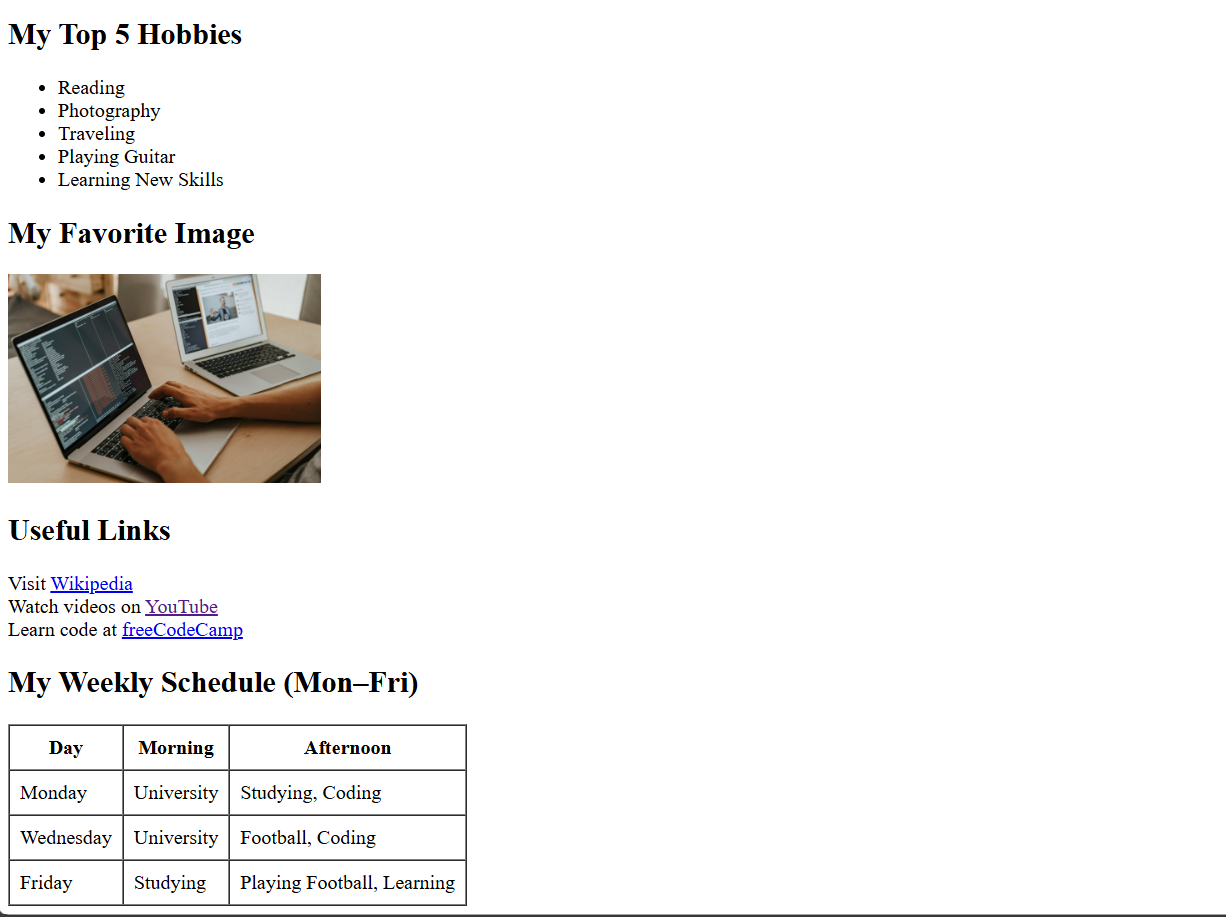
    </table>

</body>

</html>

OUTPUT

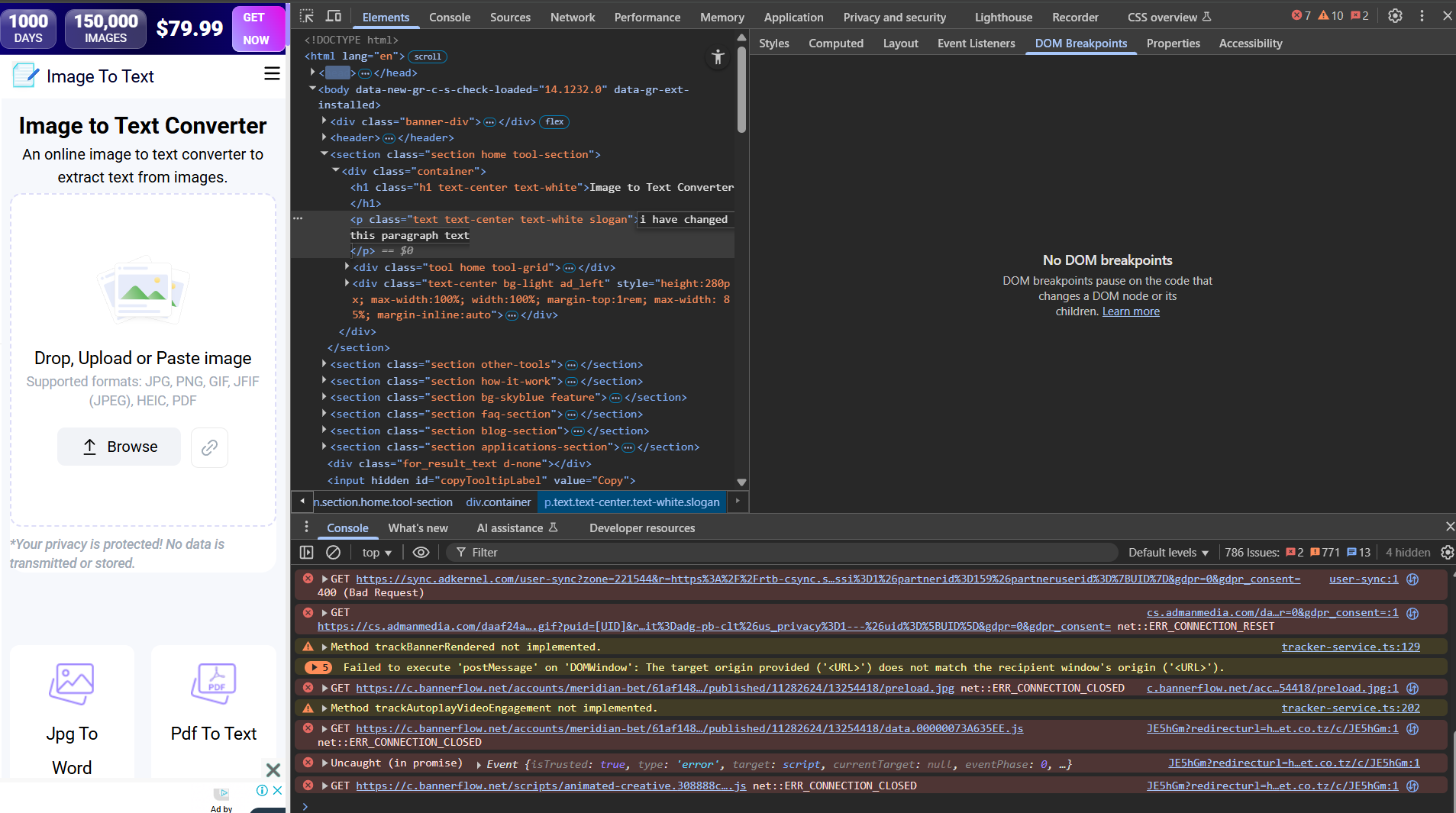




Q1: What is the purpose of the <head>section in HTML?   
A: The <head>section contains meta-information about the HTML document that is not displayed directly on the web page. It includes elements like the page title, character encoding, links to stylesheets, scripts, and favicons. This information helps browsers understand how to load and display the content correctly.

Q2: Why do we use semantic tags like <section>, <article>, and <footer>?  
A: Semantic tags give meaning to the structure of a webpage. They help both developers and browsers understand the role of different parts of the content. For example, <section> defines a thematic grouping of content, <article> represents a self-contained piece of content, and <footer> typically contains information like contact details or copyright. Semantic tags improve accessibility, SEO, and code readability.

**Part 2: Understanding and Exploring the DOM**



**Q1: How is the DOM different from the HTML file?**  
**A:**The **HTML file** is the source code written by the developer, while the **DOM (Document Object Model)** is a live, in-memory representation of that HTML structure created by the browser. The DOM can change dynamically through JavaScript or developer tools, while the original HTML file remains unchanged unless you edit and save it manually.

**Q2: What happens in the browser when you add or delete nodes in the dev tools?**  
**B:**When you add or delete nodes in the browser’s developer tools, it updates the **DOM in real-time**, and you can immediately see the changes on the web page. However, these changes are **temporary** and will disappear when the page is refreshed because they don’t affect the original HTML file saved on disk.

**Part 3: Forms in HTML**

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <title>My Form</title>

</head>

<body>

    <h1>Registration Form</h1>

    <form action="#" method="post" enctype="multipart/form-data">

        <fieldset>

            <legend>Basic Information</legend>

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

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

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

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

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

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

        </fieldset>

        <fieldset>

            <legend>Gender</legend>

            <input type="radio" id="male" name="gender" value="male">

            <label for="male">Male</label><br>

            <input type="radio" id="female" name="gender" value="female">

            <label for="female">Female</label><br>

            <input type="radio" id="other" name="gender" value="other">

            <label for="other">Other</label><br>

        </fieldset>

        <fieldset>

            <legend>Interests</legend>

            <input type="checkbox" id="coding" name="interests" value="Coding">

            <label for="coding">Coding</label><br>

            <input type="checkbox" id="sports" name="interests" value="Sports">

            <label for="sports">Sports</label><br>

            <input type="checkbox" id="music" name="interests" value="Music">

            <label for="music">Music</label><br>

        </fieldset>

        <fieldset>

            <legend>Location</legend>

            <label for="country">Country:</label>

            <select id="country" name="country">

                <option value="tanzania">Tanzania</option>

                <option value="kenya">Kenya</option>

                <option value="uganda">Uganda</option>

                <option value="rwanda">Rwanda</option>

                <option value="burundi">Burundi</option>

                <option value="south-sudan">South Sudan</option>

            </select>

        </fieldset>

        <fieldset>

            <legend>Profile Picture</legend>

            <label for="profilePic">Upload Profile Picture:</label>

            <input type="file" id="profilePic" name="profilePic" accept="image/\*">

        </fieldset>

        <br>

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

    </form>

</body>

</html>

**OUTPUT**



**Q1: What is the difference between GET and POST methods in forms?**

* **GET**: This method appends form data to the URL as query parameters. It is used when you want to retrieve or display data without making any changes on the server. GET requests have limitations on the amount of data that can be sent and should not be used for sensitive information, as the data is visible in the URL.
* **POST**: This method sends form data in the body of the HTTP request, not in the URL. It is used for submitting sensitive data or making changes to the server (e.g., creating or updating records). POST requests do not have the data size limitation like GET and are more secure for transmitting sensitive information.

**Q2: What happens when a form is submitted without a required input?**  
When a form is submitted without a required input, the browser will typically display a **warning message** and prevent the form from being submitted. The **required** attribute on input fields ensures that the user provides the necessary information before submitting the form. This behavior is built-in to most modern browsers, and the form will only submit once all required fields are completed.

**Part 4: CSS Styling**

/\* General Styles \*/

body {

    font-family: Arial, sans-serif;

    background-color: #F5EEDC;

    color: #183B4E;

    margin: 0;

    padding: 0;

}

h1, h2, h3, h4, h5, h6 {

    color: #27548A;

}

p {

    font-size: 16px;

    line-height: 1.6;

}

a {

    color: #27548A;

    text-decoration: none;

}

a:hover {

    color: #DDA853;

}

/\* Header and Page Title \*/

h1 {

    text-align: center;

    margin-top: 20px;

    font-size: 36px;

}

/\* Form Styles \*/

form {

    margin: 20px auto;

    padding: 20px;

    width: 80%;

    background-color: #FFFFFF;

    border-radius: 8px;

    box-shadow: 0px 4px 8px rgba(0, 0, 0, 0.1);

}

fieldset {

    border: 2px solid #183B4E;

    padding: 20px;

    margin-bottom: 20px;

}

legend {

    font-size: 18px;

    color: #27548A;

    padding: 0 10px;

}

label {

    display: block;

    margin-bottom: 8px;

    font-weight: bold;

}

input[type="text"],

input[type="email"],

input[type="password"],

select,

input[type="file"] {

    width: 100%;

    padding: 10px;

    margin-bottom: 10px;

    border: 1px solid #DDA853;

    border-radius: 4px;

    font-size: 14px;

}

input[type="submit"] {

    background-color: #27548A;

    color: white;

    border: none;

    padding: 10px 20px;

    cursor: pointer;

    border-radius: 5px;

    transition: background-color 0.3s ease;

}

input[type="submit"]:hover {

    background-color: #183B4E;

}

/\* Table Styles \*/

table {

    width: 80%;

    margin: 20px auto;

    border-collapse: collapse;

}

th, td {

    padding: 12px 15px;

    text-align: left;

    border: 1px solid #DDA853;

}

th {

    background-color: #27548A;

    color: white;

}

tr:nth-child(even) {

    background-color: #F5EEDC;

}

/\* Image Styles \*/

img {

    border-radius: 8px;

    margin-bottom: 20px;

    display: block;

    margin-left: auto;

    margin-right: auto;

}

/\* Useful Links Section \*/

#useful-links {

    text-align: center;

    margin: 30px 0;

}

#useful-links a {

    font-size: 18px;

    margin: 0 15px;

    text-transform: uppercase;

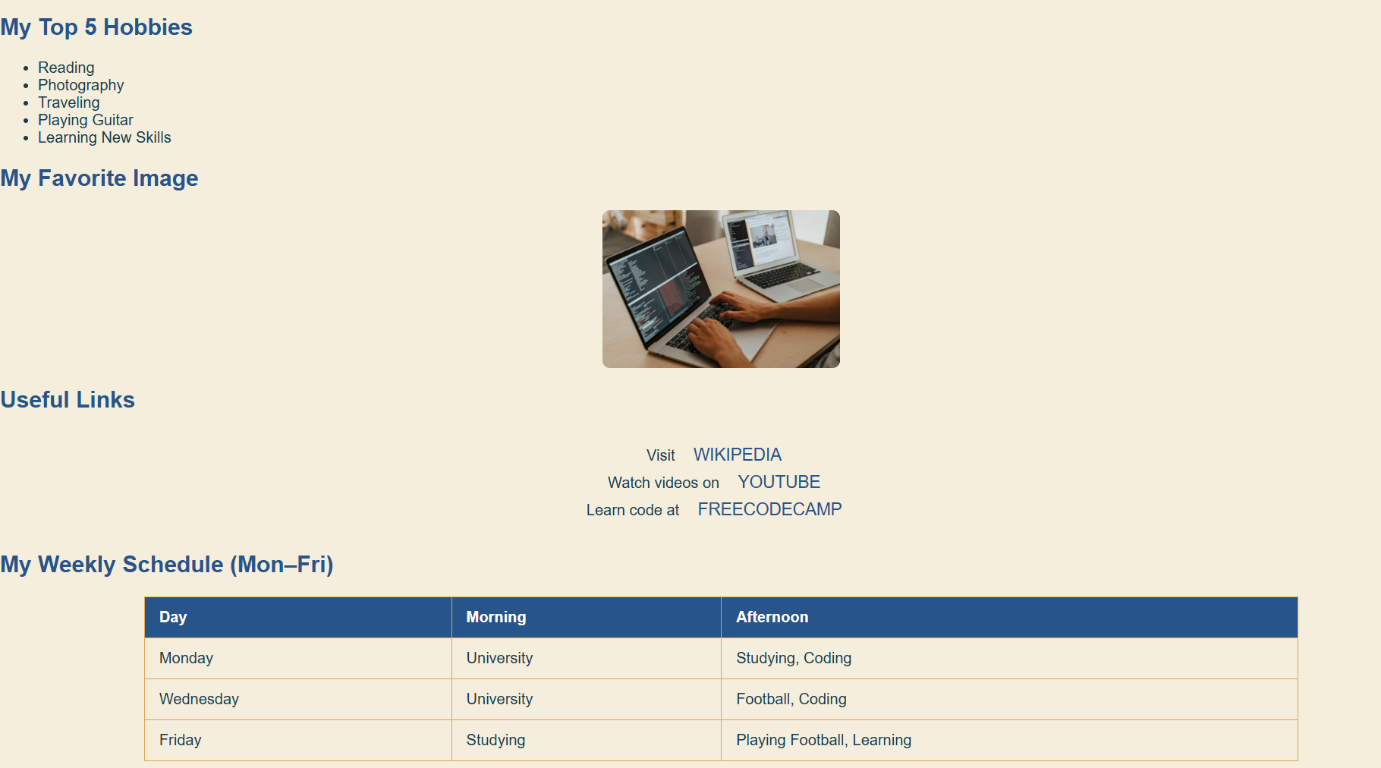
}

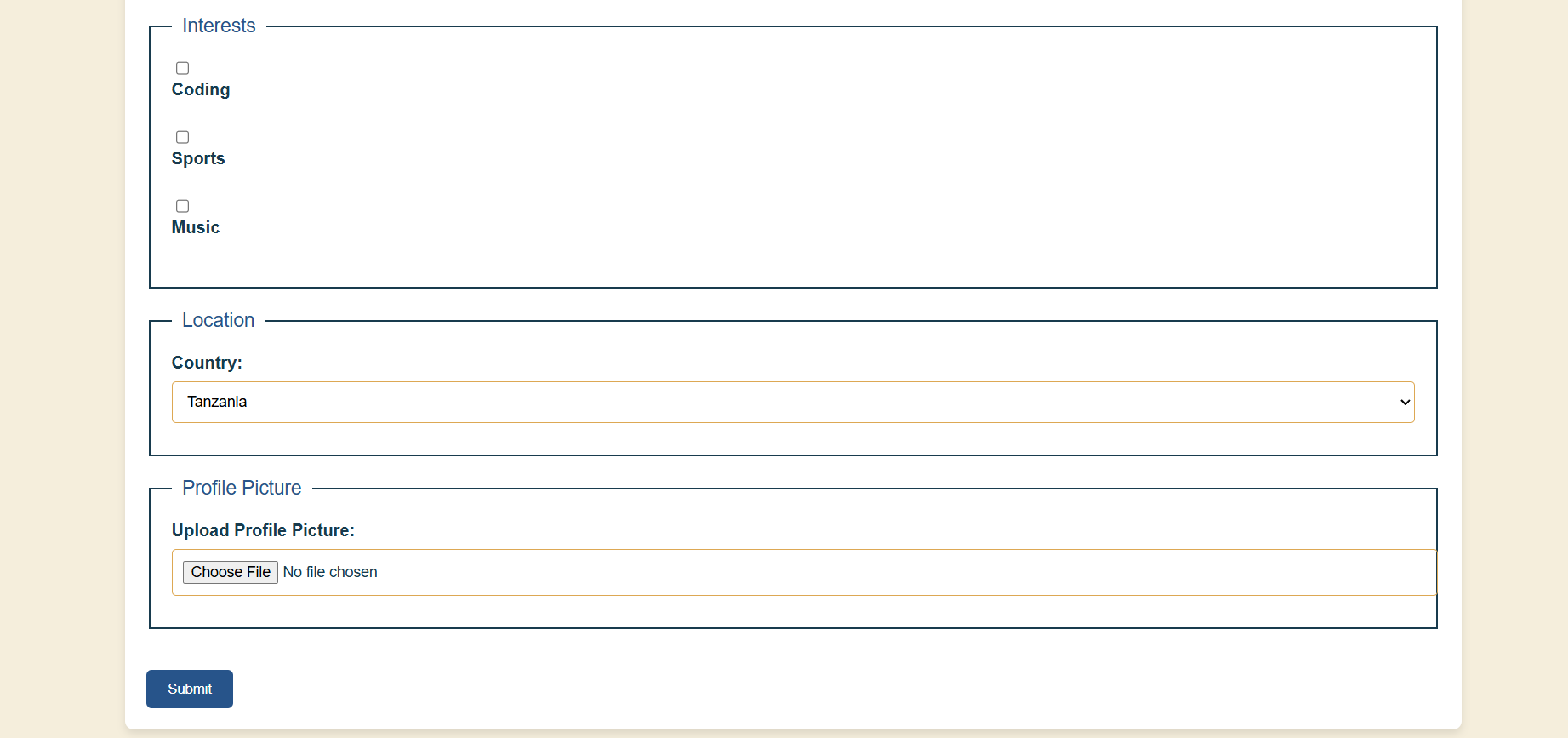
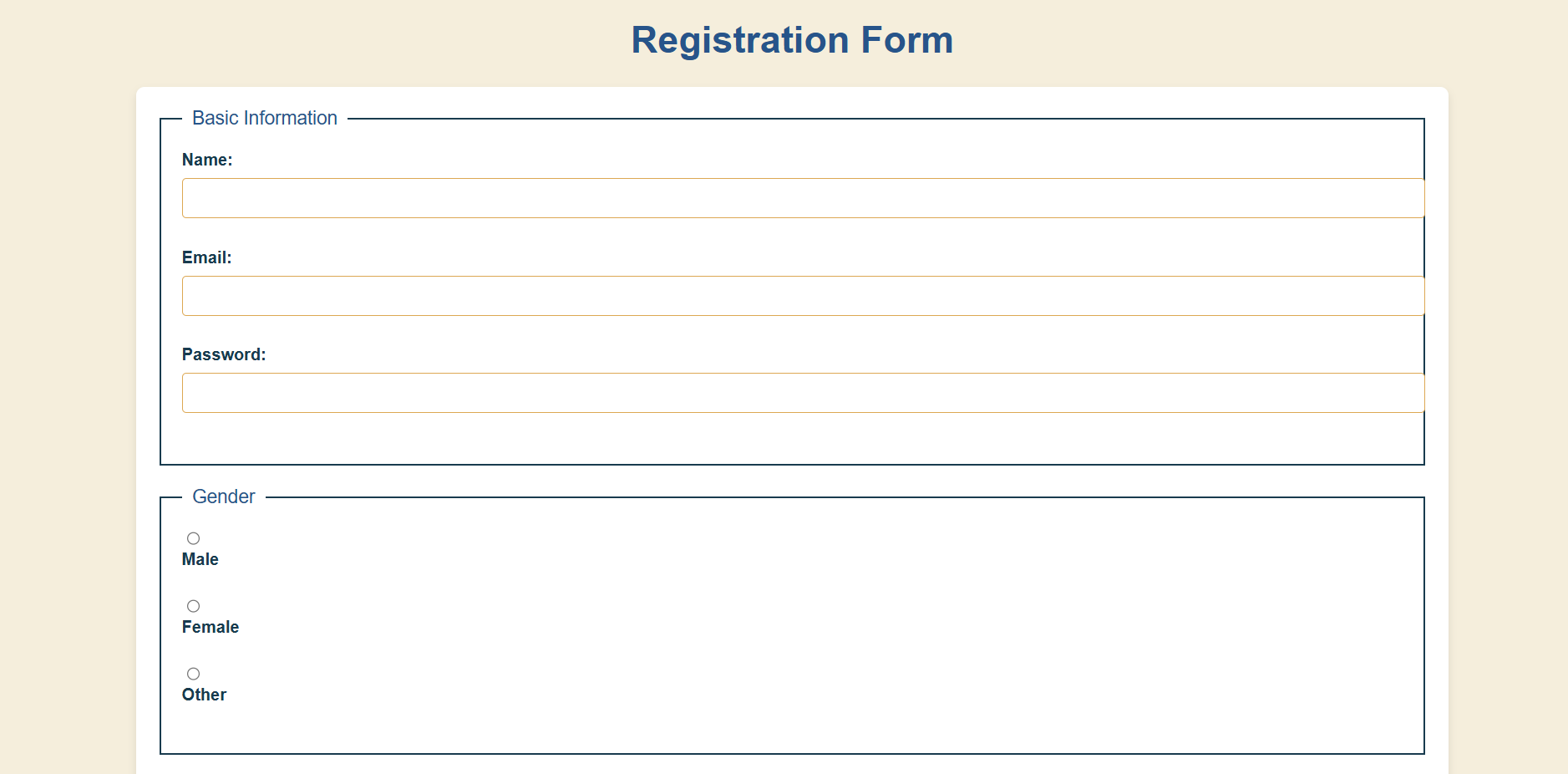
#useful-links a:hover {

    color: #183B4E;

}

**OUTPUT**





Here are the brief differences:

**Questions:**  
What’s the difference between inline, internal, and external CSS?

* **Inline CSS**: Styles are applied directly within an HTML element using the style attribute. It only affects that specific element.
* **Internal CSS**: Styles are defined inside the <style> tag in the <head> section of the HTML document. It affects the entire document but is not reusable across multiple pages.
* **External CSS**: Styles are written in a separate .css file and linked to the HTML document. It allows for styling multiple pages consistently and is the most efficient for larger projects.

How do you resolve conflicting styles between selectors?

* **Specificity**: More specific selectors (like IDs) override less specific ones (like classes).
* **Order of Appearance**: When selectors have the same specificity, the last one in the stylesheet is applied.
* **!important**: Forces a style to override others, but should be used sparingly to avoid issues with maintainability.