Factorial Calculator

A simple and responsive web application to calculate the factorial of a number
using iterative or recursive methods.

Overview

The Factorial Calculator is a user-friendly web application designed to compute the factorial of a number using two different methods: iterative and recursive. The application emphasizes simplicity and functionality, ensuring users can calculate results with minimal effort. It incorporates input validation to ensure accurate calculations and provides dynamic output that clearly displays the result and the selected method.

The application is designed with a modular approach, making it suitable for further enhancements such as backend integration or deployment on hosting platforms. It serves as a practical example of web development, blending front-end technologies like HTML, CSS, and JavaScript, and optionally leveraging backend technologies like Node.js for extended functionality.

App Functionality

The web application's primary function is to compute factorial values. Users are presented with a single input field to enter a positive integer. Upon entering the number, they can choose their preferred calculation method—iterative or recursive. After selecting a method and clicking the "Calculate" button, the application instantly displays the calculated factorial and indicates the method used.

To ensure user inputs are valid, the app includes robust input validation. Non-integer values, negative numbers, or empty fields prompt immediate error messages, guiding users to correct their inputs. This validation ensures the application runs smoothly without errors, even with improper input.

- Type in a positive integer in the input box.
- Select one of the option given below:

  1. Iterative Method

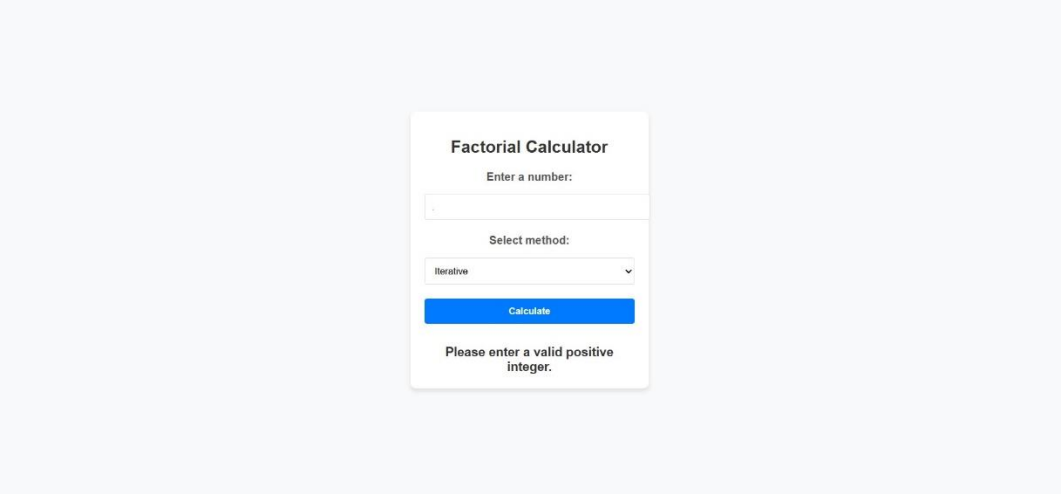
  The iterative method calculates the factorial of a number using a loop. It involves starting from 1 and repeatedly multiplying the current result by the next integer until the desired number is reached.

  2. Recursive Method

The **recursive method** calculates the factorial by repeatedly calling the same function with a smaller input until a base case is reached. This method leverages the mathematical definition of factorial: n!=n×(n−1)! with 0!=1n! = n \times (n-1)! \text{ with } 0! = 1n!=n×(n−1)! with 0!=1.

- Click the calculate button

Validation : If the user introduces an invalid input, an error message will be displayed for the user to update the input.

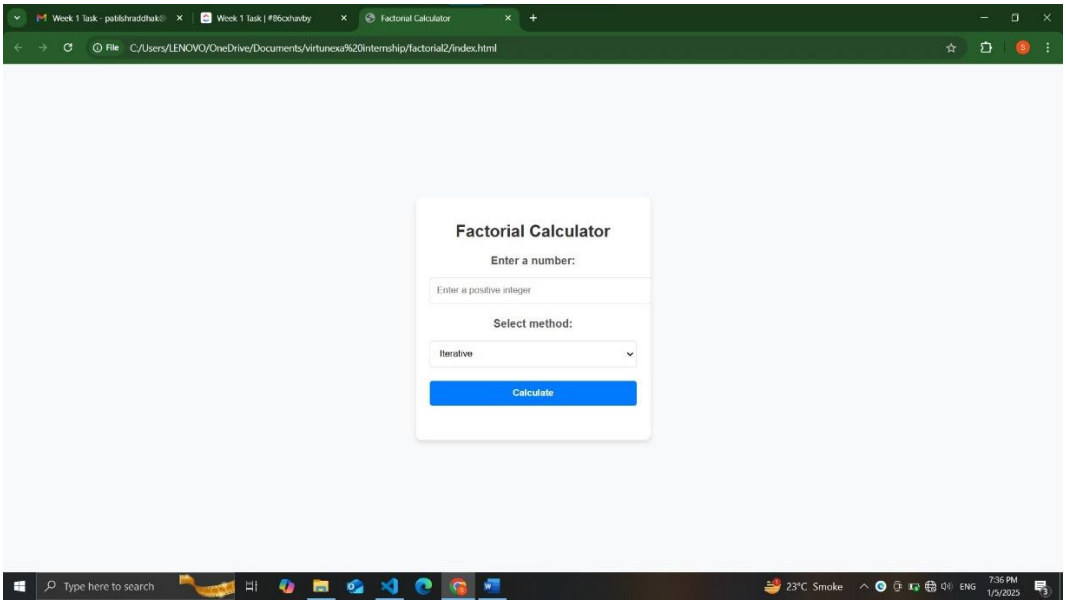


Testing

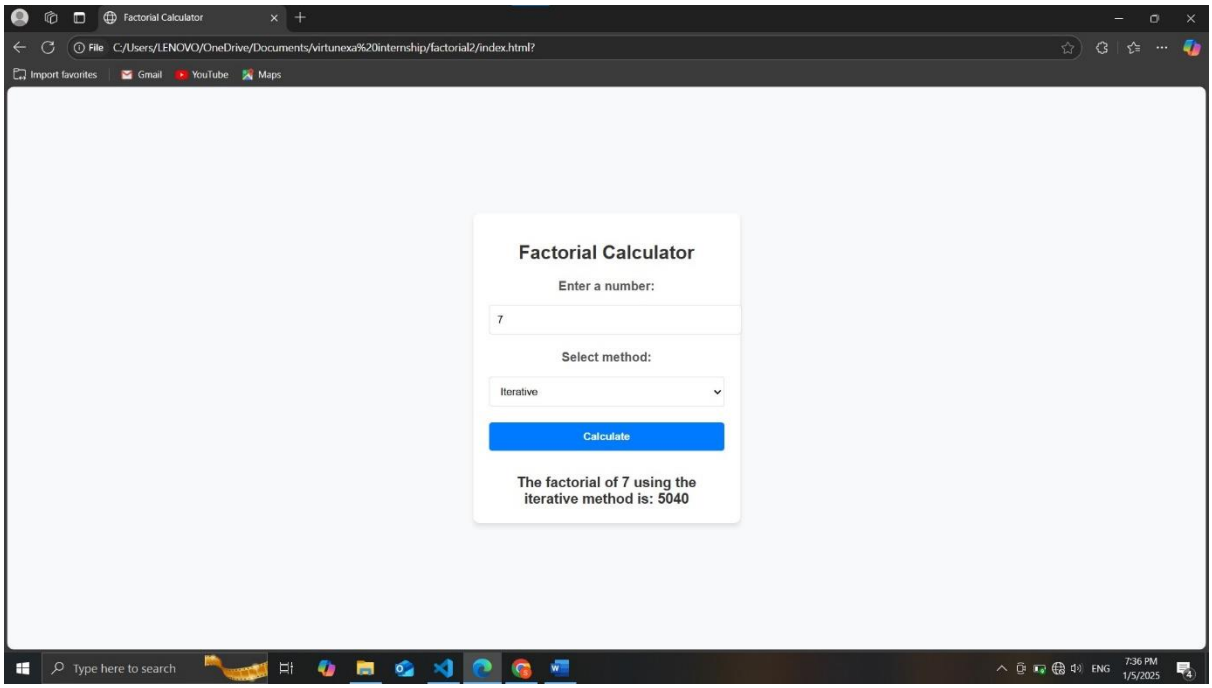
The application was run and tested on different browsers and devices to ensure responsiveness and cross-browser compatibility.

Browser used :

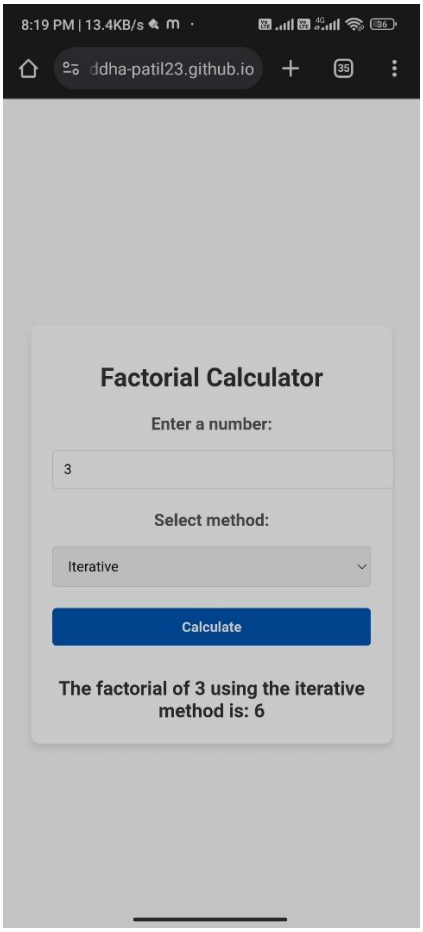
Google Chrome



Microsoft Edge



Mobile :



Html code

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Factorial Calculator</title>
    <link rel="stylesheet" href="styles.css">
</head>
<body>
    <div class="container">
        <h1>Factorial Calculator</h1>
        <form id="factorialForm">
            <label for="number">Enter a number:</label>
            <input type="number" id="number" placeholder="Enter a positive integer" required>

            <label for="method">Select method:</label>
            <select id="method">
                <option value="iterative">Iterative</option>
                <option value="recursive">Recursive</option>
            </select>

            <button type="button" id="calculateBtn">Calculate</button>
        </form>
        <div id="output"></div>
    </div>
    <script src="script.js"></script>
</body>
</html>
```

CSS code

```
body {
    font-family: Arial, sans-serif;
    background-color: #f8f9fa;
    margin: 0;
    padding: 0;
    display: flex;
    justify-content: center;
    align-items: center;
    height: 100vh;
}

.container {
    background: #fff;
    padding: 20px;
    border-radius: 8px;
    box-shadow: 0 4px 6px rgba(0, 0, 0, 0.1);
    width: 300px;
    text-align: center;
}

h1 {
    font-size: 24px;
    margin-bottom: 20px;
    color: #333;
}

label {
    font-weight: bold;
    display: block;
```

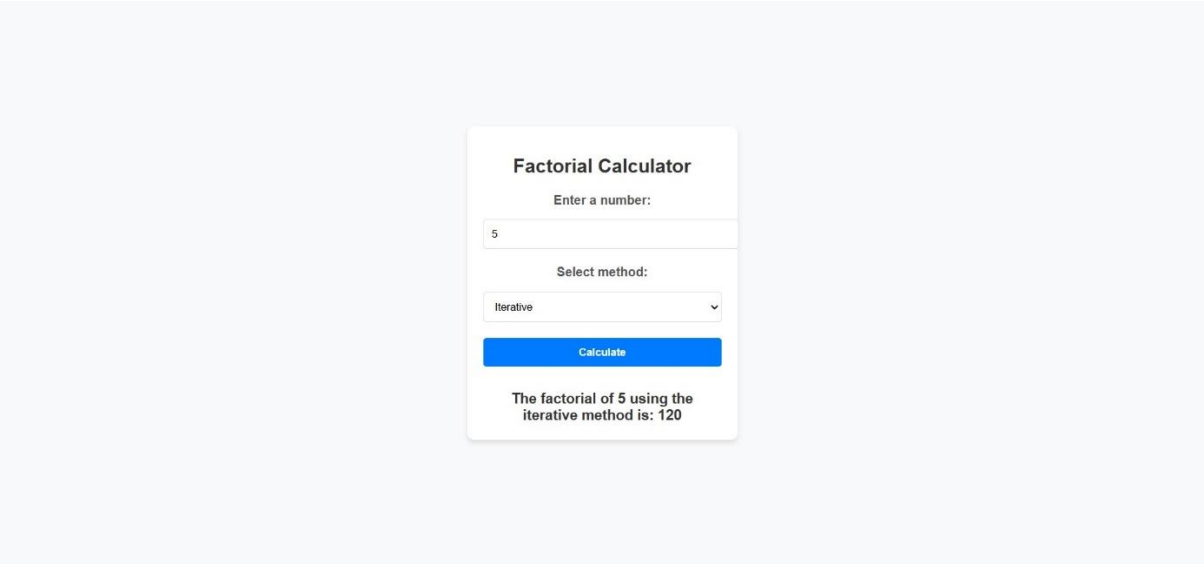
```
    margin: 10px 0 5px;
    color: #555;
}

input, select, button {
    width: 100%;
    padding: 10px;
    margin: 10px 0;
    border: 1px solid #ddd;
    border-radius: 4px;
}

button {
    background-color: #007bff;
    color: white;
    font-weight: bold;
    border: none;
    cursor: pointer;
}

button:hover {
    background-color: #0056b3;
}

#output {
    margin-top: 20px;
    font-size: 18px;
    color: #333;
    font-weight: bold;
}
```

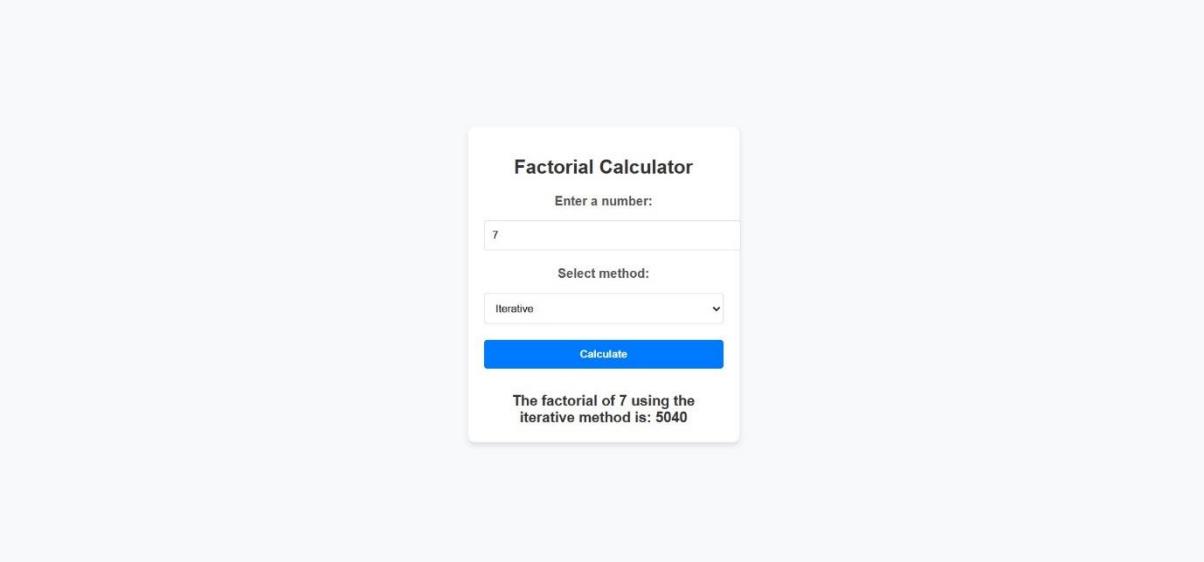
Javascript code

```
// JavaScript for factorial calculations
document.getElementById("calculateBtn").addEventListener("click", () => {
    const numberInput = document.getElementById("number").value;
    const method = document.getElementById("method").value;
    const outputDiv = document.getElementById("output");

    // Clear previous output
    outputDiv.innerHTML = "";

    // Input validation
    const number = parseInt(numberInput);
    if (isNaN(number) || number < 0) {
        outputDiv.innerHTML = "Please enter a valid positive integer.";
        return;
    }

    // Factorial calculation
    let result;
    if (method === "iterative") {
        result = factorialIterative(number);
    } else if (method === "recursive") {
        result = factorialRecursive(number);
    }

    // Display result
        outputDiv.innerHTML = `The factorial of ${number} using the ${method} method is: <strong>${result}</strong>`;
});
```

```
// Iterative factorial function
function factorialIterative(num) {
    let result = 1;
    for (let i = 1; i <= num; i++) {
        result *= i;
    }
    return result;
}

// Recursive factorial function
function factorialRecursive(num) {
    if (num === 0 || num === 1) {
        return 1;
    }
    return num * factorialRecursive(num - 1);
}
```

Output :





Link for the web app :

https://shraddha-patil23.github.io/factorial-webapp/