**Wall Task :**

**Functionalities:**

The webpage provides a user interface for inputting the number of walls and their heights.

It calculates and displays the walls that are visible from both the right and left sides.

The code generates a visual representation of the walls, highlighting visible walls in green (from the right) and blue (from the left).

The user can enter the number of walls and their heights, and upon clicking the "Calculate" button, the results and graphics are updated**.**

**Logic:**

JavaScript handles the logic for calculating visible walls, both from the right and left sides, based on the given input heights.

The code dynamically generates and updates the visual representation of the walls using HTML and CSS.

**Script:**

<!DOCTYPE html>

<html>

<head>

<style>

body {

font-family: Arial, sans-serif;

text-align: center;

display: flex;

flex-direction: column;

align-items: center;

justify-content: center;

height: 100vh;

margin: 0;

}

h1 {

font-size: 24px;

}

#input-section {

margin: 20px;

padding: 20px;

background-color: #f0f0f0;

border-radius: 10px;

box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);

text-align: left;

}

#output {

margin: 20px;

}

label {

font-weight: bold;

}

input[type="number"],

input[type="text"] {

width: 100%;

padding: 10px;

margin: 5px 0;

border: 1px solid #ccc;

border-radius: 4px;

}

button {

padding: 10px 20px;

background-color: #212223;

color: #fff;

border: none;

border-radius: 4px;

cursor: pointer;

font-size: 16px;

transition: background-color 0.3s ease;

}

button:hover {

background-color: #566e7e;

}

</style>

</head>

<body>

<h1>Visible Walls</h1>

<div id="input-section">

<h2>Input Parameters:</h2>

<label for="numWalls">Number of Walls:</label>

<input type="number" id="numWalls" min="1" value="">

<label for="heights">Height of Walls (Enter values separated by #):</label>

<input type="text" id="heights" value="">

<button onclick="calculateVisibleWalls()">Calculate</button>

</div>

<div id="output">

<h2>Results:</h2>

<h3>Walls Visible from Right:</h3>

<p id="visibleRight"></p>

<h3>Walls Visible from Left:</h3>

<p id="visibleLeft"></p>

<h3>Graphics:</h3>

<div id="graphics"></div>

</div>

<script>

function calculateVisibleWalls() {

const numWalls = parseInt(document.getElementById('numWalls').value);

const heights = document.getElementById('heights').value.split('#').map(Number);

// Calculate visible walls from the right

let visibleRight = [heights[0]];

let maxRight = heights[0];

for (let i = 1; i < numWalls; i++) {

if (heights[i] > maxRight) {

visibleRight.push(heights[i]);

maxRight = heights[i];

}

}

// Calculate visible walls from the left

let visibleLeft = [heights[numWalls - 1]];

let maxLeft = heights[numWalls - 1];

for (let i = numWalls - 2; i >= 0; i--) {

if (heights[i] > maxLeft) {

visibleLeft.unshift(heights[i]);

maxLeft = heights[i];

}

}

// Display the results

document.getElementById('visibleRight').textContent = visibleRight.join(', ');

document.getElementById('visibleLeft').textContent = visibleLeft.join(', ');

// Generate graphics

generateGraphics(heights, visibleRight, visibleLeft);

}

function generateGraphics(heights, visibleRight, visibleLeft) {

const graphicsContainer = document.getElementById('graphics');

graphicsContainer.innerHTML = '';

for (let i = 0; i < heights.length; i++) {

const wall = document.createElement('div');

wall.className = 'wall';

wall.style.height = heights[i] \* 20 + 'px';

if (visibleRight.includes(heights[i])) {

wall.style.backgroundColor = 'green';

}

if (visibleLeft.includes(heights[i])) {

wall.style.backgroundColor = 'blue';

}

graphicsContainer.appendChild(wall);

}

}

</script>

<style>

.wall {

width: 20px;

background-color: gray;

margin: 5px;

display: inline-block;

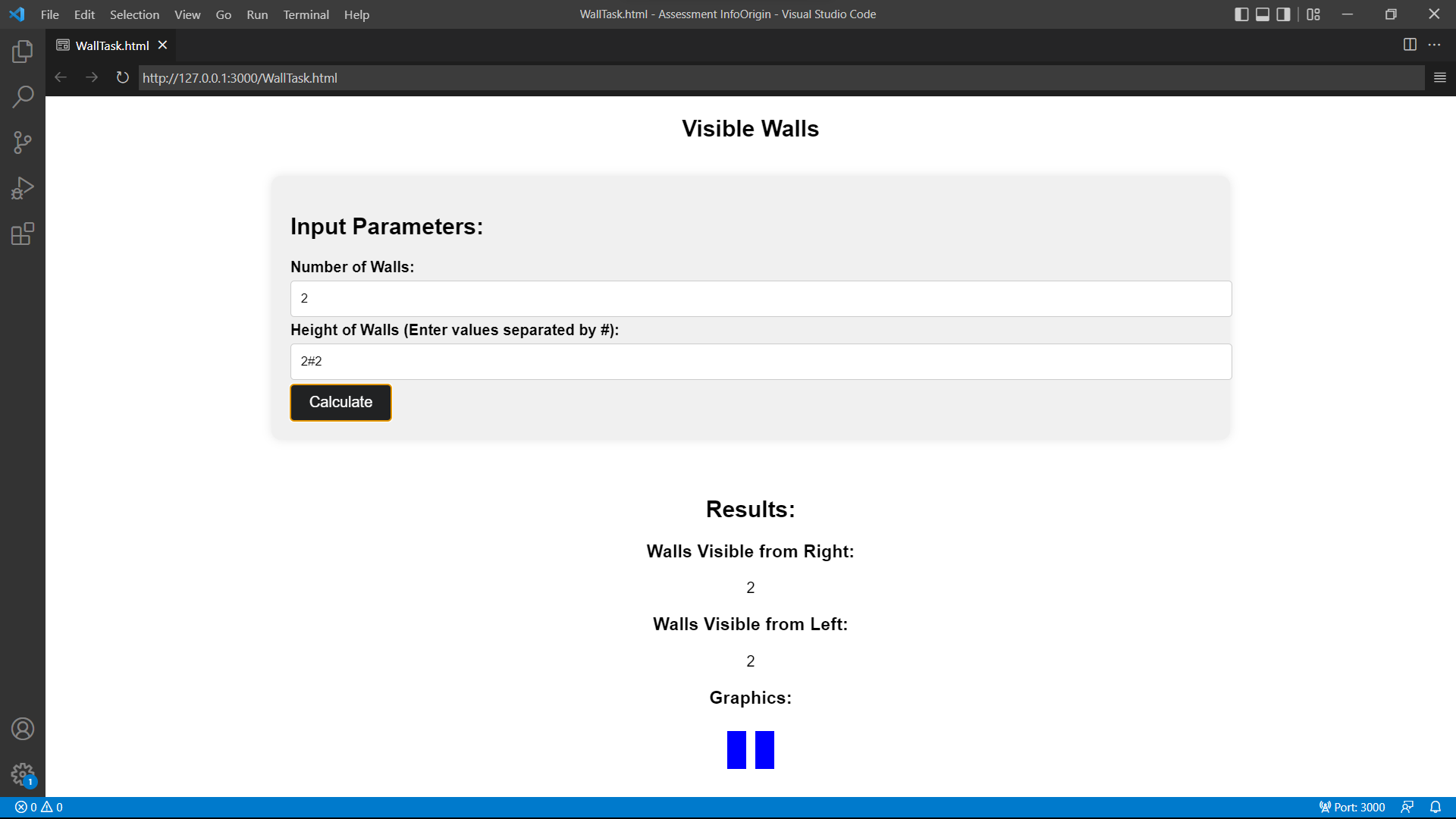
}

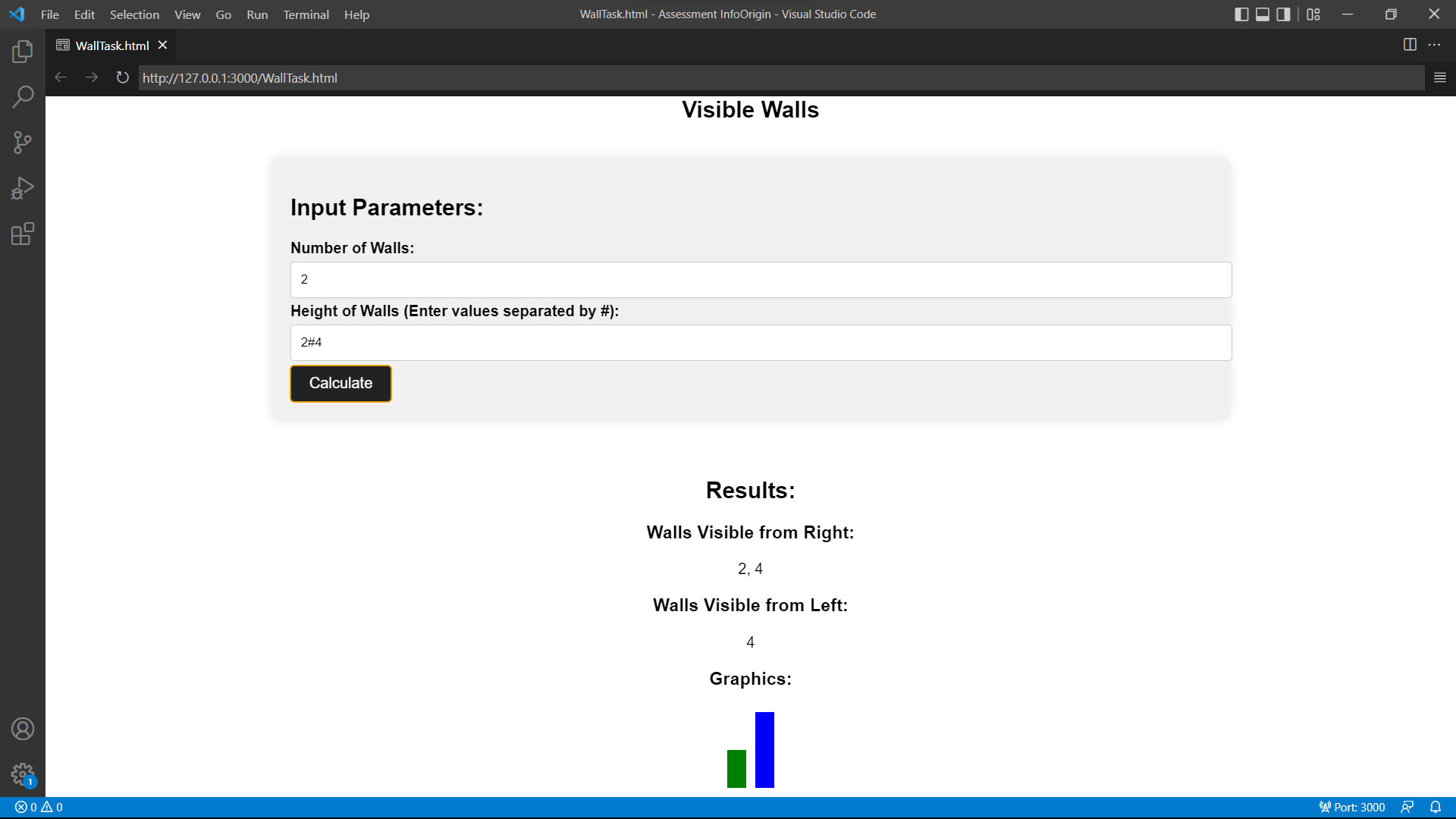
</style>

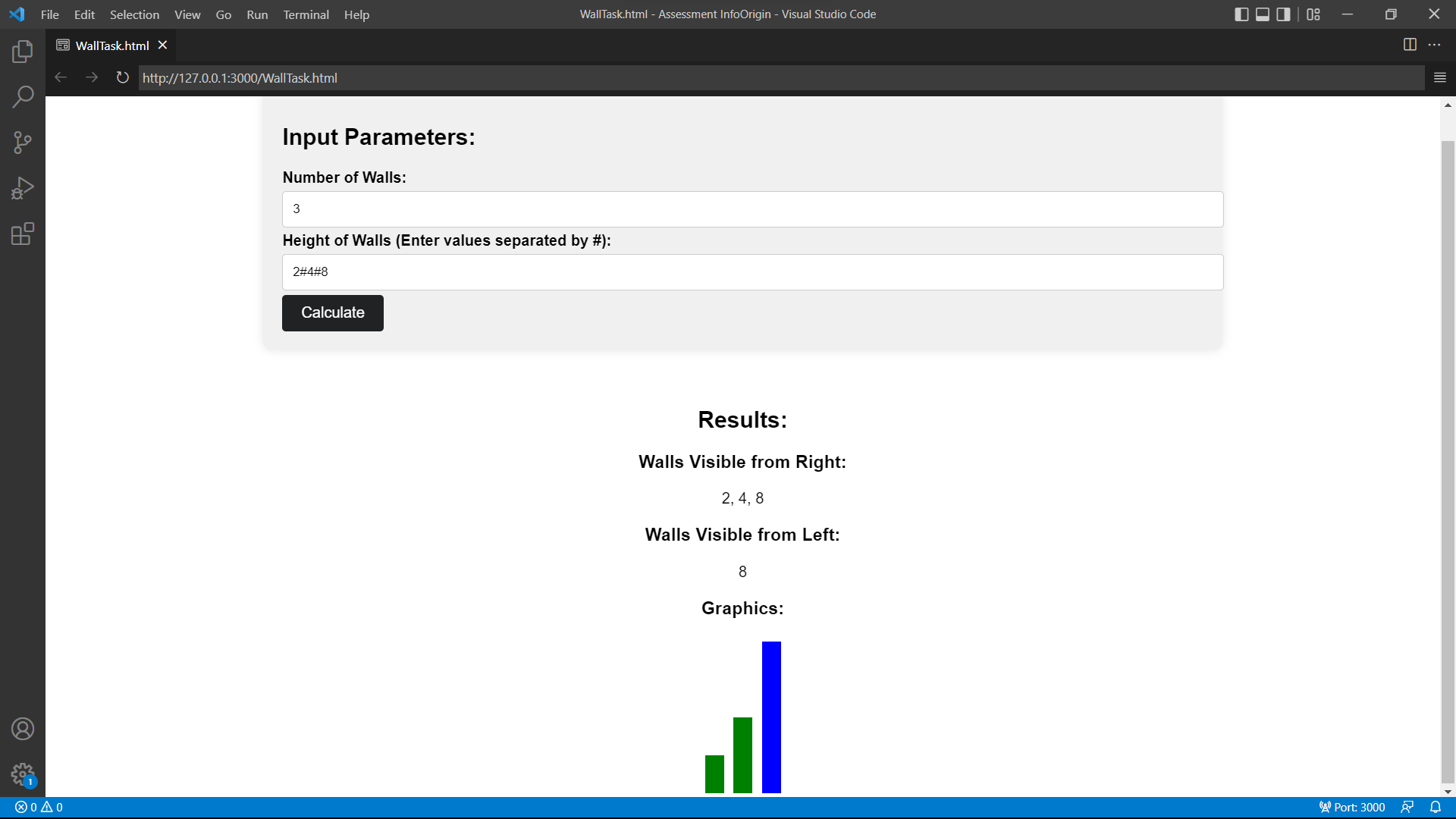
</body>

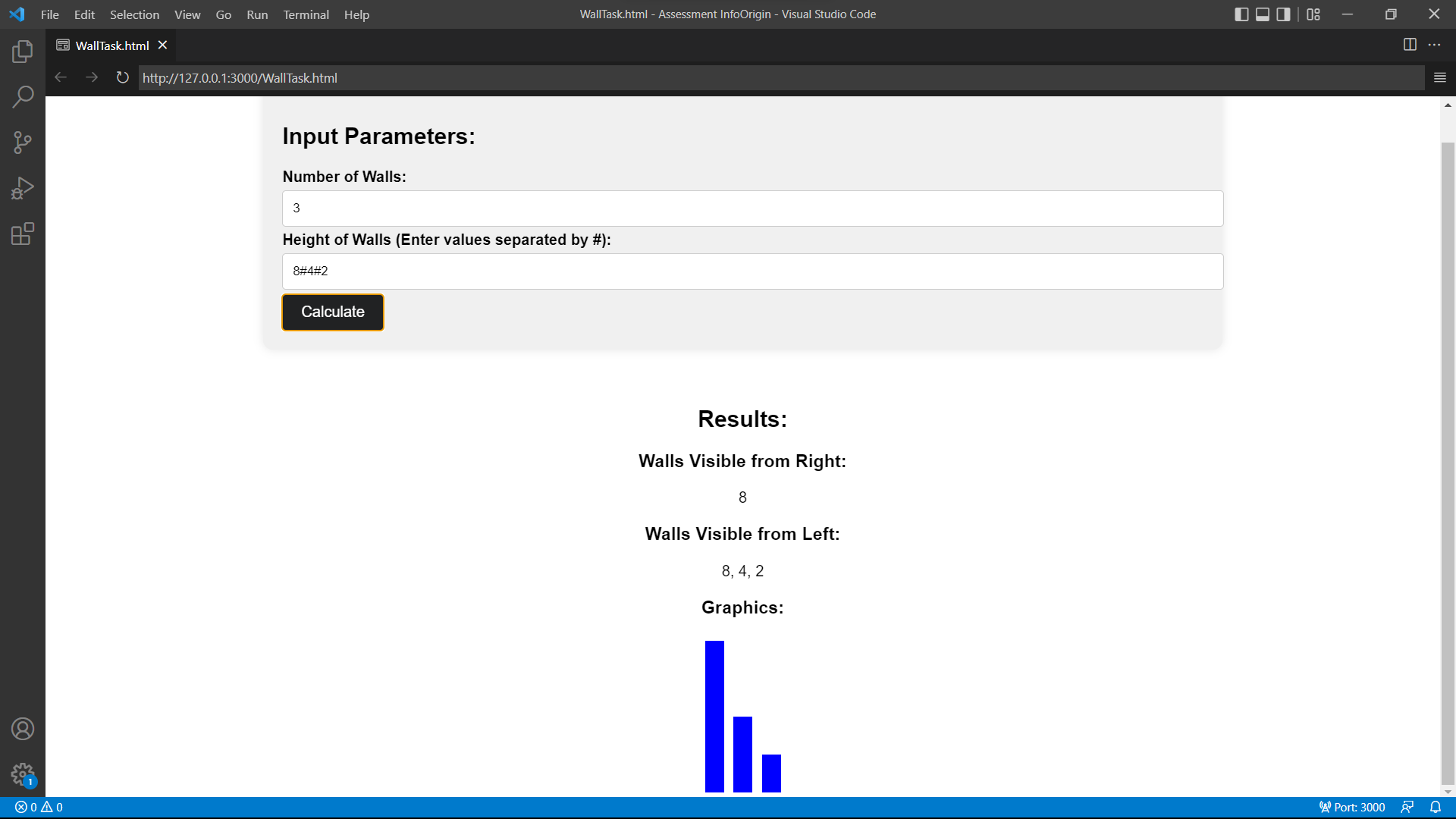
</html>

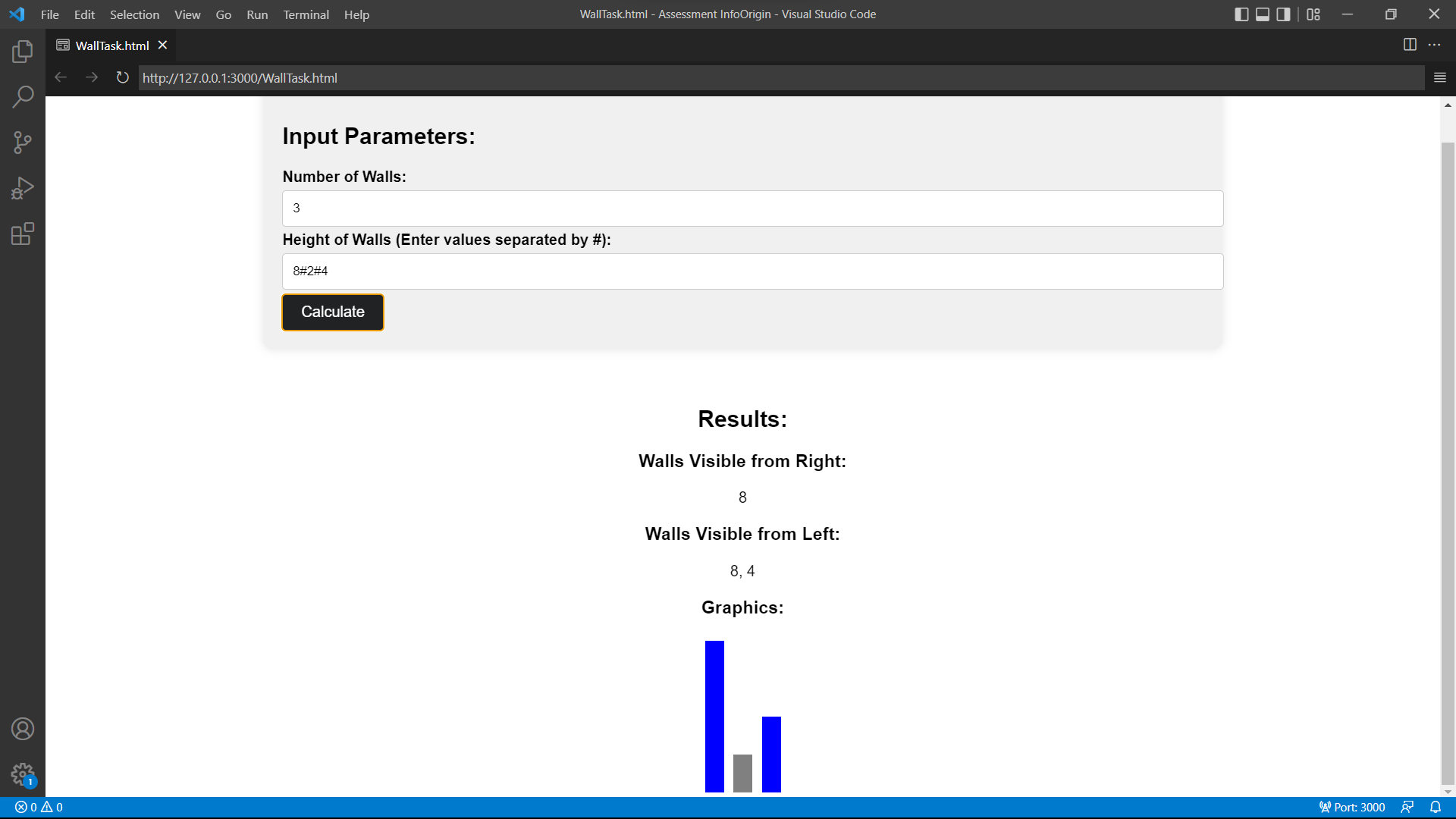
**Output:**

****

****

****

****

**s**