**Date :**

**Practical No 5**

Aim : Create a website that shows different methods of embedding javascript.

Source code:

HTML CODE-

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8" />

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

<title>JavaScript Embedding Methods</title>

<style>

/\* Global styles \*/

body {

font-family: Arial, sans-serif;

margin: 0;

padding: 0;

background-image: url(bg.jpg); /\* Light gray background \*/

background-position: center;

background-repeat: no-repeat;

background-size: cover;

background-attachment: fixed;

}

/\* Navbar styles \*/

.navbar {

background-color: #04467c;

overflow: hidden;

}

.navbar a {

float: left;

display: block;

color: #f2f2f2;

text-align: center;

padding: 14px 20px;

text-decoration: none;

}

.navbar a:hover {

background-color: #ddd;

color: rgb(255, 253, 253);

}

/\* Content styles \*/

.content {

padding: 20px;

/\*background-color: #fff; /\* White background for content \*/

margin-top: 20px;

}

/\* Button container \*/

.button-container {

display: flex;

justify-content: space-around;

}

/\* Button styles \*/

.button {

font-size: large;

padding: 10px 20px;

background-color: #3fcddf; /\* Green \*/

color: rgb(12, 12, 12);

border: none;

border-radius: 5px;

cursor: pointer;

}

.button:hover {

background-color: #2bfff8; /\* Darker green on hover \*/

}

.text {

color: #ddd;

}

.title {

align-items: center;

color: aquamarine;

}

</style>

</head>

<body>

<!-- Navbar -->

<div class="navbar">

<a href="#">Home</a>

<a href="#">About</a>

<a href="#">Contact</a>

</div>

<!-- Content -->

<div class="content">

<h1 class="title">Embedding JavaScript Methods</h1>

<p class="text">

1.Inline JavaScript:Inline JavaScript involves directly embedding

JavaScript code within HTML elements using event attributes such as

onclick, onmouseover, onload, etc. The JavaScript code is included

directly within the HTML tags, making it convenient for simple scripts

or quick actions.<br />2.Internal JavaScript: Internal JavaScript refers

to including JavaScript code within tags and can be placed anywhere in

the HTML document.<br />3.External JavaScript: External JavaScript

involves storing JavaScript code in separate external files with a .js

extension and linking them to HTML documents using the improving code

organization, reusability, and maintainability.

</p>

<!-- Button container -->

<div class="button-container">

<!-- Button 1 - Inline JavaScript -->

<button class="button" onclick="alert('Alert using Inline JavaScript')">

Inline JavaScript

</button>

<!-- Button 2 - Internal JavaScript -->

<button class="button" id="internalBtn">Internal JavaScript</button>

<!-- Button 3 - External JavaScript -->

<button class="button" id="externalBtn">External JavaScript</button>

</div>

</div>

<!-- Internal JavaScript -->

<script>

document.getElementById("internalBtn").onclick = function () {

alert("Alert using Internal JavaScript");

};

</script>

<!-- External JavaScript -->

<script src="external.js"></script>

</body>

</html>

2nd file :external.js

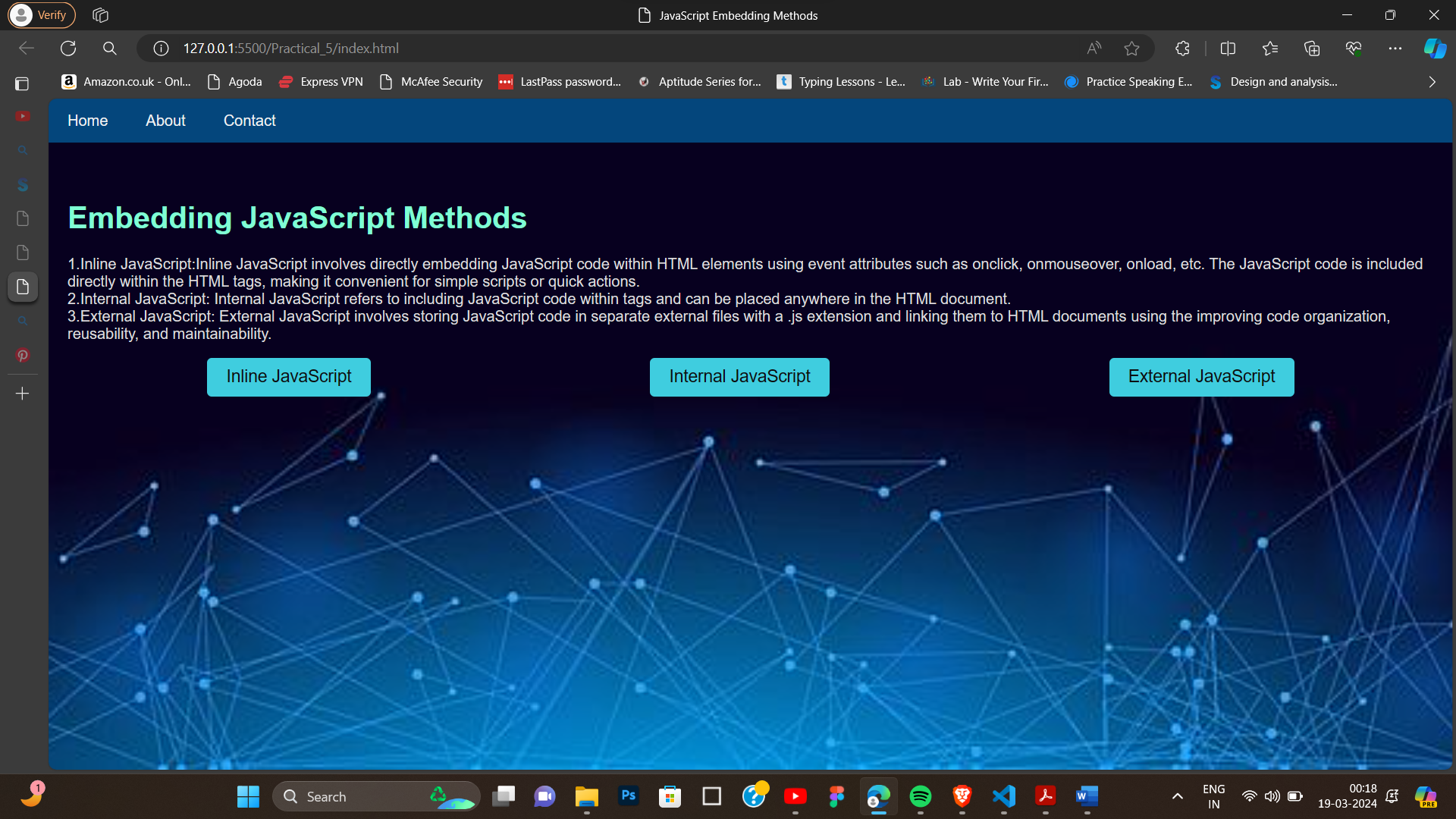
// external.js

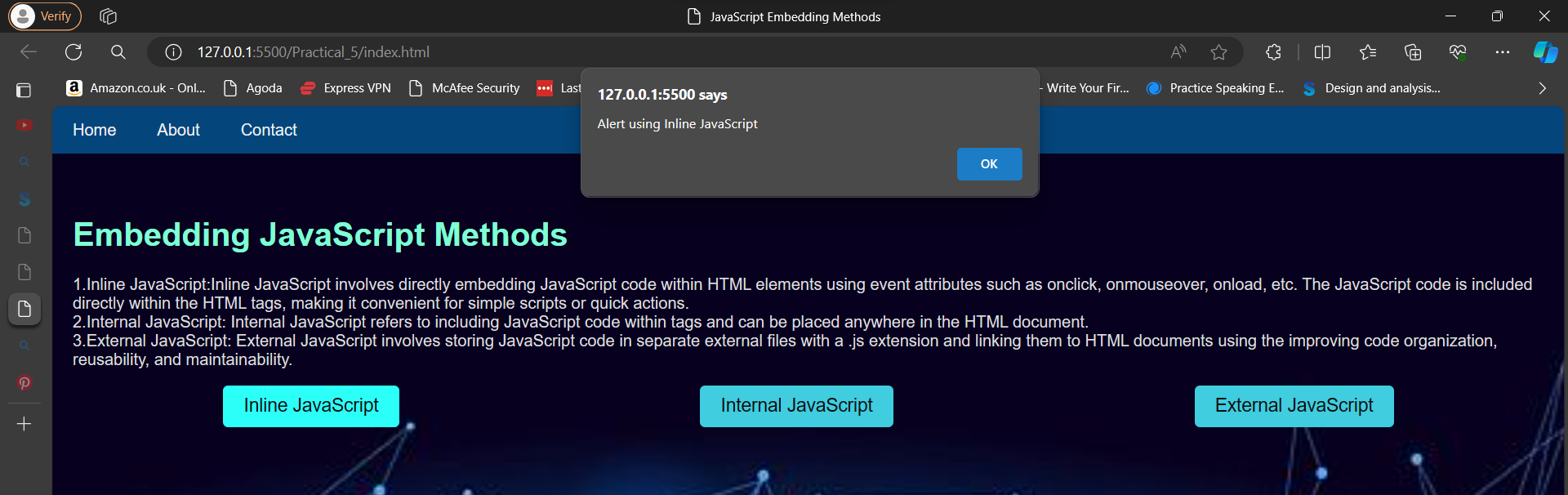
document.getElementById("externalBtn").onclick = function () {

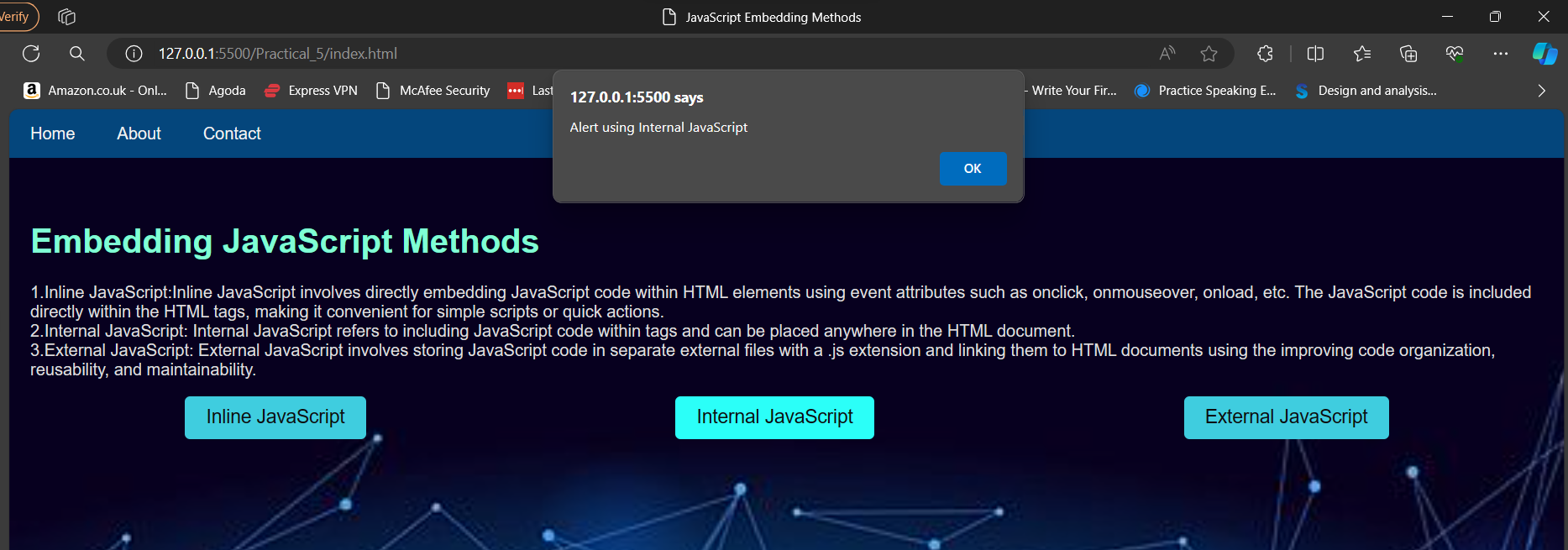
alert("Alert using External JavaScript");

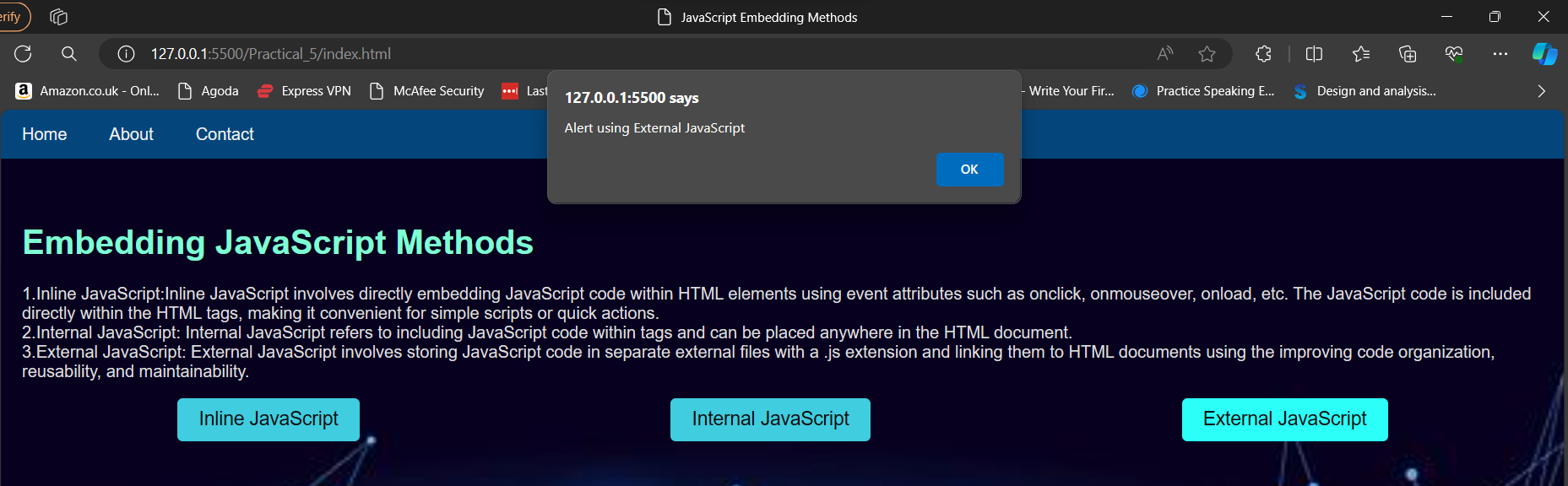
};

Output:









Conclusion :

In conclusion, the practical on embedded JavaScript provides a comprehensive understanding of three different methods of incorporating JavaScript code into HTML documents. Through the exploration of inline JavaScript, internal JavaScript, and external JavaScript, we can write javascript in different ways .