Nithin S
221IT085

IT254 Lab Assignment 2

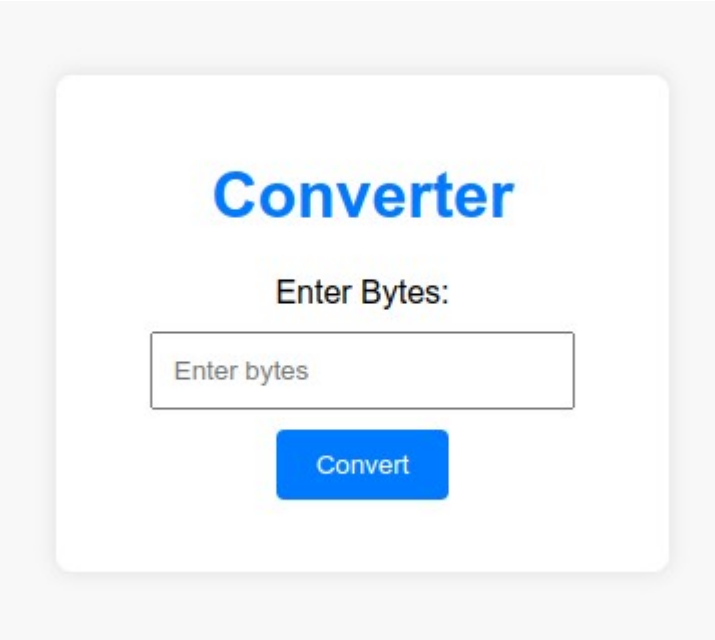
Q1) Write a JavaScript program to convert a number in bytes to a humanreadable string.

CODE

```html
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Converter</title>
    <style>
        body {
            font-family: 'Arial', sans-serif;
            background-color: #f8f8f8;
            margin: 0;
            display: flex;
            align-items: center;
            justify-content: center;
            height: 100vh;
        }

        .container {
            background-color: #fff;
            border-radius: 8px;
            box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);
            padding: 20px;
            text-align: center;
        }

        h1 {
            color: #007BFF;
        }

        input {
            width: 80%;
            padding: 10px;
            margin: 10px 0;
            box-sizing: border-box;
        }

        button {
            background-color: #007BFF;
            color: #fff;
            padding: 10px 20px;
            border: none;
            border-radius: 4px;
            cursor: pointer;
        }

        button:hover {
            background-color: #0056b3;
        }
    </style>
</head>
```

```html
<body>
    <div class="container">
        <h1>Converter</h1>
        <label for="inputBytes">Enter Bytes:</label>
        <input type="number" id="inputBytes" placeholder="Enter bytes">
        <br>
        <button type="button" onclick="convertBytes()">Convert</button>
        <p id="result"></p>
    </div>

    <script>
        const convertBytes = () => {
            const inputBytes = document.getElementById("inputBytes").value;
            const resultElement = document.getElementById("result");

            if (!inputBytes || isNaN(inputBytes)) {
                resultElement.textContent = "Please enter a valid number.";
            } else {
                const result = prettyBytes(inputBytes);
                resultElement.textContent = `Converted: ${result}`;
            }
        };

        const prettyBytes = (num, precision = 3, addSpace = true) => {
            const UNITS = ['B', 'KB', 'MB', 'GB', 'TB', 'PB', 'EB', 'ZB', 'YB'];
            if (Math.abs(num) < 1) return num + (addSpace ? ' ' : '') + UNITS[0];
            const exponent = Math.min(Math.floor(Math.log10(num < 0 ? -num : num) / 3), UNITS.length -
1);
            const n = Number(((num < 0 ? -num : num) / 1000 ** exponent).toPrecision(precision));
            return (num < 0 ? '-' : '') + n + (addSpace ? ' ' : '') + UNITS[exponent];
        };
    </script>
</body>
</html>
```

OUTPUT



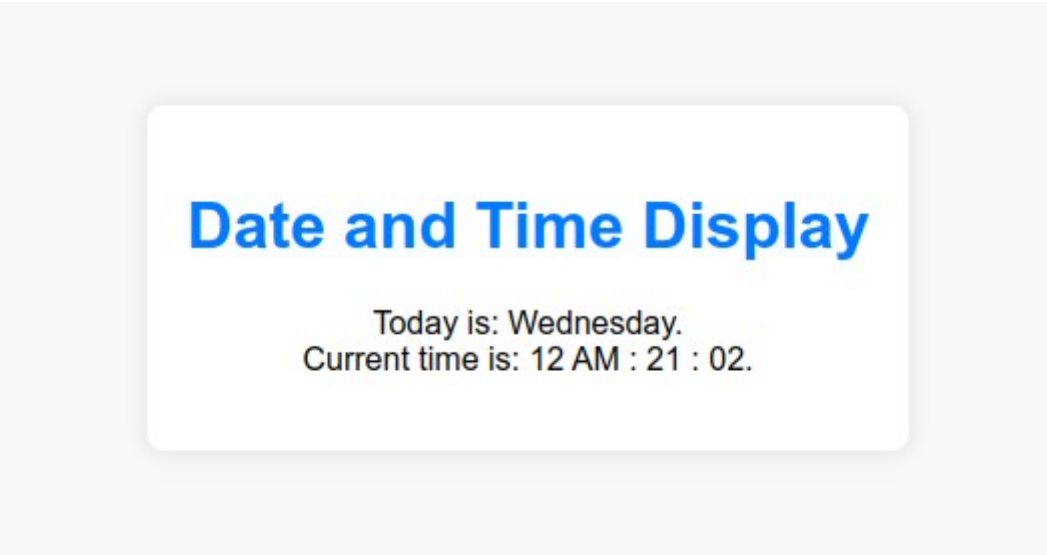
Q2) Write a JavaScript program to display the current day and time in the
following format.
Today is : Tuesday.
Current time is : 3 PM : 20 : 28

CODE

```html
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Date and Time Display</title>
    <style>
        body {
            font-family: 'Arial', sans-serif;
            background-color: #f8f8f8;
            margin: 0;
            display: flex;
            align-items: center;
            justify-content: center;
            height: 100vh;
        }

        .container {
            background-color: #fff;
            border-radius: 8px;
            box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);
            padding: 20px;
            text-align: center;
        }

        h1 {
            color: #007BFF;
        }
    </style>
</head>
<body>
    <div class="container">
        <h1>Date and Time Display</h1>
        <p id="dateTime"></p>
    </div>
```

```html
    <script>
        const displayDateTime = () => {
            const dateTimeElement = document.getElementById("dateTime");
            const daysOfWeek = ['Sunday', 'Monday', 'Tuesday', 'Wednesday', 'Thursday', 'Friday',
'Saturday'];
            const months = ['January', 'February', 'March', 'April', 'May', 'June', 'July', 'August',
'September', 'October', 'November', 'December'];

            const now = new Date();
            const dayOfWeek = daysOfWeek[now.getDay()];
            const month = months[now.getMonth()];
            const date = now.getDate();
            const year = now.getFullYear();
            let hours = now.getHours();
            const ampm = hours >= 12 ? 'PM' : 'AM';
            hours = hours % 12 || 12;
            const minutes = now.getMinutes();
            const seconds = now.getSeconds();

            const formattedTime = `${hours} ${ampm} : ${minutes < 10 ? '0' : ''}${minutes} : ${seconds
< 10 ? '0' : ''}${seconds}`;

            const formattedDateTime = `Today is: ${dayOfWeek}.<br>Current time is: ${formattedTime}.`;

            dateTimeElement.innerHTML = formattedDateTime;
        };

        // Call the function to display date and time when the page loads
        displayDateTime();

        // Update the date and time every second
        setInterval(displayDateTime, 1000);
    </script>
</body>
</html>
```

OUTPUT



Q3) Write a JavaScript program to create and display a Singly Linked List.

CODE

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

<head>
    <title>Create and display a Singly Linked List</title>
</head>
<style>
    body {
        font-family: Arial, sans-serif;
        margin: 0;
        padding: 0;
    }

    h1 {
        text-align: center;
        margin: 20px 0;
    }

    label,
    input,
    button {
        display: block;
        margin-bottom: 10px;
    }

    input,
    button {
        padding: 5px;
        font-size: 16px;
        border: 1px solid #ccc;
        border-radius: 5px;
    }

    input[type="number"] {
        width: 100px;
    }

    button {
        background-color: #08c0d4;
        color: white;
        cursor: pointer;
    }

    button:hover {
        background-color: black;
    }

    #list {
        margin: 20px 0;
        padding: 0;
        list-style-type: none;
        text-align: center;
    }
```

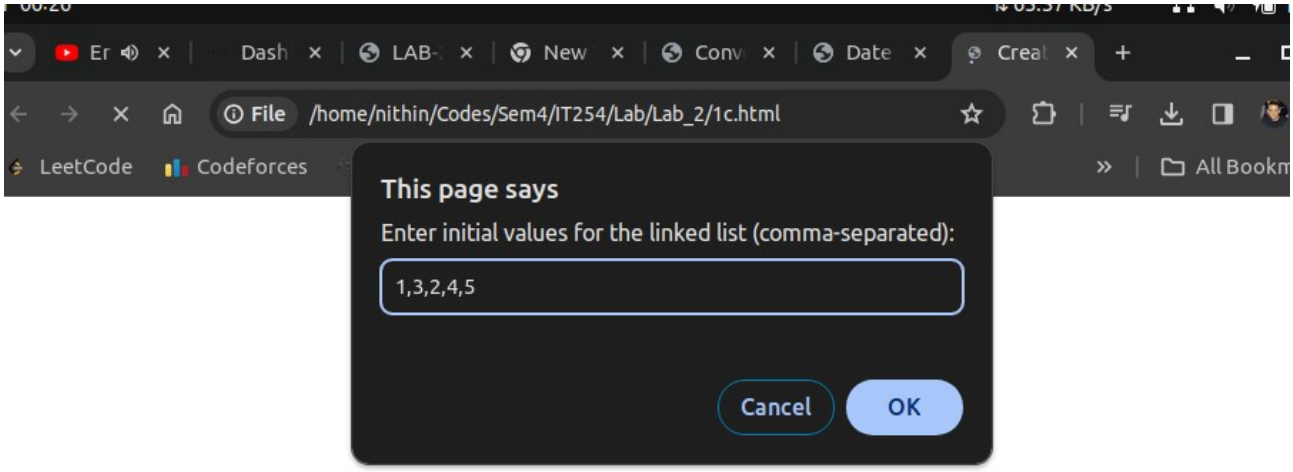
```
    #list li {
        display: inline-block;
        margin-right: 10px;
    }

    #list li .arrow {
        margin-left: 5px;
        margin-right: 5px;
        font-size: 20px;
    }

    .error {
        color: red;
        font-weight: bold;
        margin-top: 10px;
    }

    #initialValues {
        /* hide the initial values */
        display: none;
    }

    .main-body {
        border: 1px solid black;
        border-radius: 5px;
        text-align: center;
        display: flex;
        flex-direction: column;
        align-items: center;
        justify-content: center;
    }
</style>

</style>

<body onload="createLinkedList()">
    <h1>Inserting Nodes in a Singly Linked List</h1>
    <div class="main-body">
        <input type="text" id="initialValues" />
        <p>Enter position to insert node at:</p>
        <input type="number" id="position" />
        <p>Enter value for new node:</p>
        <input type="number" id="value" />
        <button onclick="insertNode()">Insert Node</button>
    </div>
    <p style="text-align: center;">Result:</p>
    <ul id="list"></ul>
```

```
<script>
    // class for each node in the linked list
    class Node {
        constructor(data) {
            this.data = data;
            this.next = null;
        }
    }

    // class for the linked list
    class LinkedList {
        constructor() {
            this.head = null;
        }

        // Method for adding a new node to the end of the list
        addNode(data) {
            var newNode = new Node(data);
            if (!this.head) {
                this.head = newNode;
            } else {
                let currentNode = this.head;
                while (currentNode.next) {
                    currentNode = currentNode.next;
                }
                currentNode.next = newNode;
            }
        }

        // Method for inserting a new node at a specified position in the list
        insertNodeAtPosition(data, position) {
            var newNode = new Node(data);
            if (position === 0) {
                newNode.next = this.head;
                this.head = newNode;
            } else {
                let currentNode = this.head;
                for (let i = 0; i < position - 1; i++) {
                    currentNode = currentNode.next;
                }
                newNode.next = currentNode.next;
                currentNode.next = newNode;
            }
        }
    }
```

```javascript
// Function for displaying the linked list on the webpage
function displayList() {
    // Clearing the list
    var listElement = document.getElementById("list");
    listElement.innerHTML = "";

    // Displaying the list
    let currentNode = linkedList.head;
    while (currentNode) {
        var liElement = document.createElement("li");
        liElement.textContent = currentNode.data;

        // Adding arrow element
        if (currentNode.next) {
            var arrowElement = document.createElement("span");
            arrowElement.innerHTML = "&#8594;";
            arrowElement.className = "arrow";
            liElement.appendChild(arrowElement);
        }

        listElement.appendChild(liElement);
        currentNode = currentNode.next;
    }
}

// Creating the linked list from user input
var linkedList = new LinkedList();
var initialValuesInput = document.getElementById("initialValues");

function createLinkedList() {
    var initialValues = prompt(
        "Enter initial values for the linked list (comma-separated):"
    );
    if (initialValues) {
        initialValuesInput.value = initialValues;
        var initialValuesArray = initialValues.split(",");
        for (let i = 0; i < initialValuesArray.length; i++) {
            linkedList.addNode(Number(initialValuesArray[i]));
        }
        displayList();
    }
}
```

```
        // Function for inserting a new node at a specified position in the list
        function insertNode() {
            // Getting the position and value entered by the user
            var positionInput = document.getElementById("position");
            var valueInput = document.getElementById("value");
            var position = Number(positionInput.value);
            var value = Number(valueInput.value);

            // Determining the length of the current list
            let currentNode = linkedList.head;
            let length = 0;
            while (currentNode) {
                length++;
                currentNode = currentNode.next;
            }

            // Ensuring that the position entered by the user is valid
            if (position < 0 || position > length) {
                alert("Error: Invalid position entered!");
            } else {
                linkedList.insertNodeAtPosition(value, position);
                positionInput.value = "";
                valueInput.value = "";
                displayList();
            }
        }
    </script>

</body>

</html>
```

OUTPUT



Inserting Nodes in a Singly Linked List

Enter position to insert node at:

Enter value for new node:

Insert Node

Result:

1 → 3 → 2 → 4 → 5

Inserting Nodes in a Singly Linked List

Enter position to insert node at:

1

Enter value for new node:

100

Insert Node

Result:

1 → 3 → 2 → 4 → 5

Inserting Nodes in a Singly Linked List

Enter position to insert node at:

Enter value for new node:

Insert Node

Result:

1 → 100 → 3 → 2 → 4 → 5

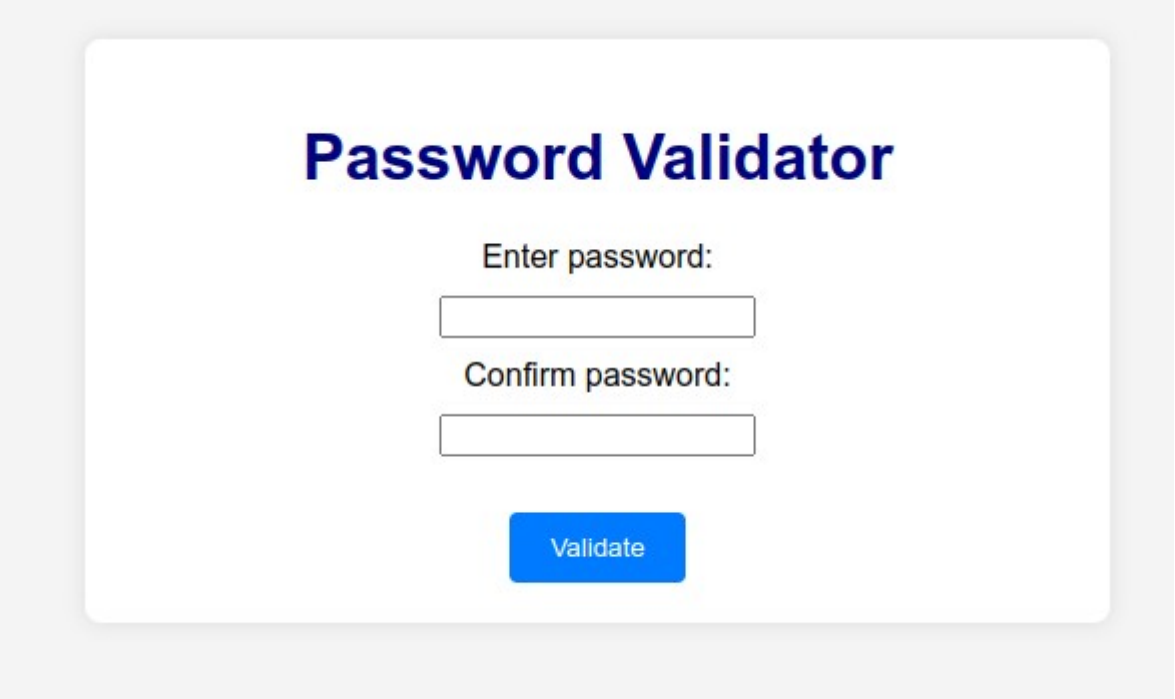
Q3) Create a webpage that allows the user to enter a password two times to
validate it.
Web page content:
• Two password fields: first to enter the password and a second one to
verify it
• A button labeled "Validate" that alerts one of the following messages -
Display an informative error message if any of the following occur:
o The passwords entered don't match
o the passwords are not at least 8 characters long

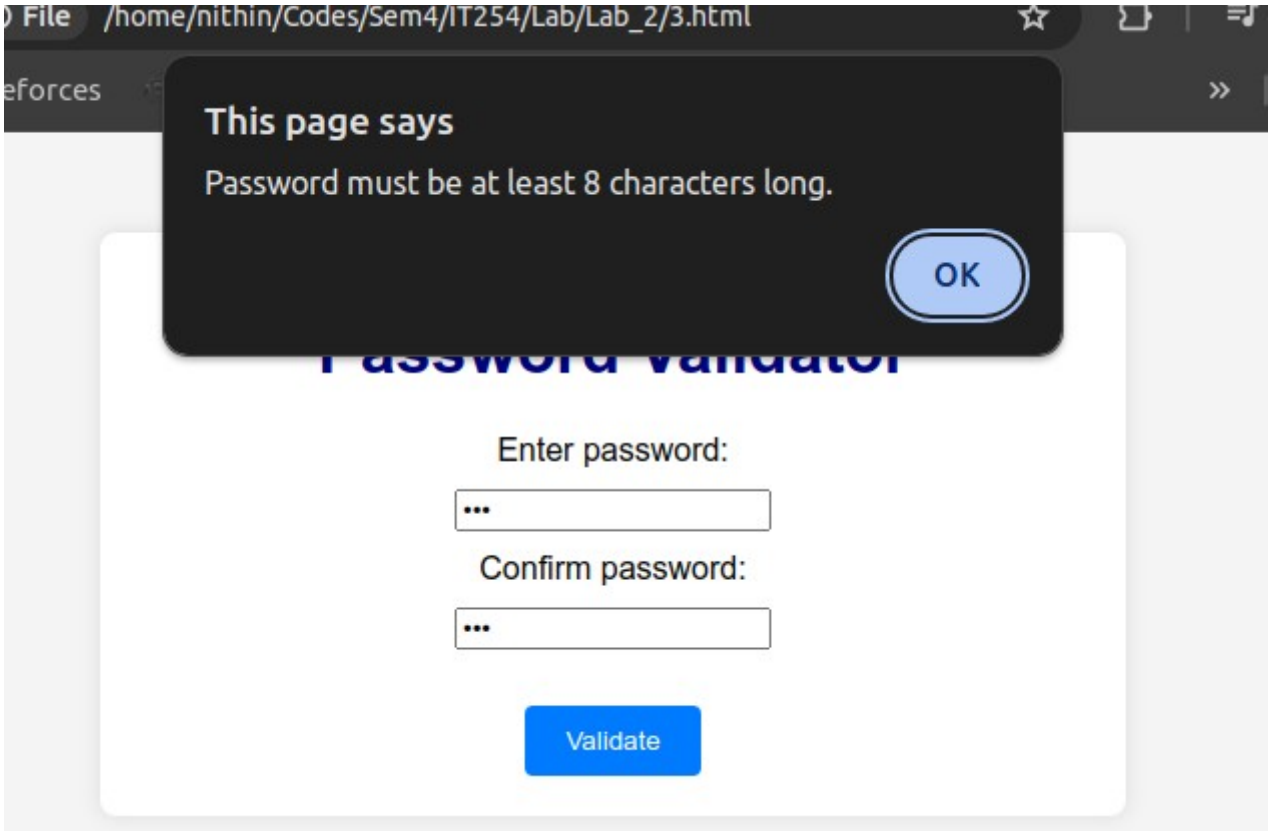
CODE

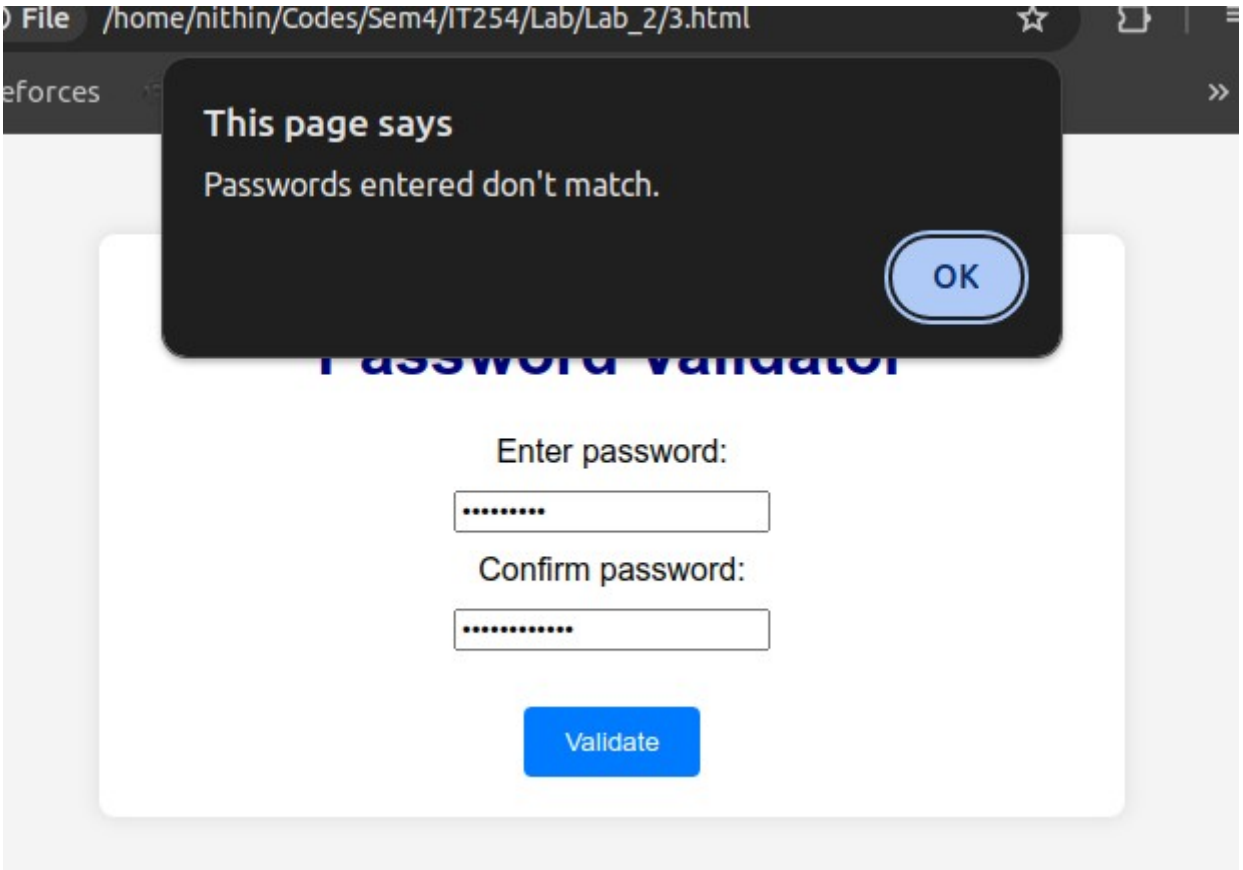
```html
<!DOCTYPE html>
<html lang="en">

<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Password Validator</title>
    <style>
        body {
            background-color: #f4f4f4;
            font-family: Arial, sans-serif;
        }

        .container {
            width: 50%;
            margin: 50px auto;
            background-color: #fff;
            padding: 20px;
            border-radius: 8px;
            box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);
            text-align: center;
        }

        h1 {
            color: rgb(0, 4, 128);
        }

        form {
            margin-top: 20px;
            display: flex;
            flex-direction: column;
            align-items: center;
        }

        label, input {
            display: block;
            margin-bottom: 10px;
        }

        button {
            background-color: #007BFF;
            color: #fff;
            padding: 10px 20px;
            border: none;
            border-radius: 4px;
            cursor: pointer;
            display: inline-block; /* Added to center the button */
        }
    </style>
</head>
```

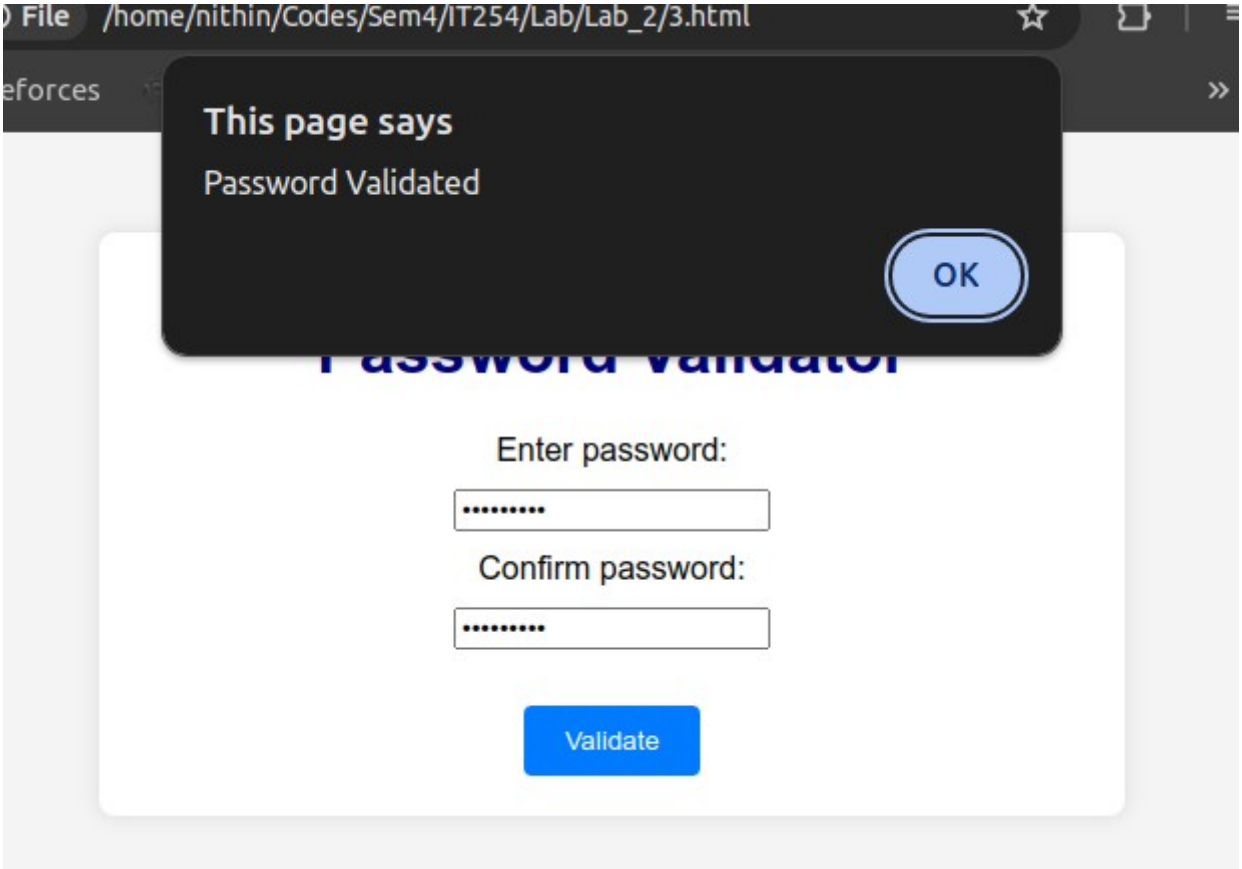
```html
<body>

    <div class="container">
        <h1>Password Validator</h1>
        <form id="passwordForm">
            <label for="password1">Enter password:</label>
            <input type="password" id="password1" name="password1">
            <label for="password2">Confirm password:</label>
            <input type="password" id="password2" name="password2">
        </form>
        <br/>
        <button type="button" onclick="validatePassword()">Validate</button>
    </div>

    <script>
        function validatePassword() {
            var password1 = document.getElementById("password1").value;
            var password2 = document.getElementById("password2").value;

            if (password1.length < 8) {
                alert("Password must be at least 8 characters long.");
            } else if (password1 !== password2) {
                alert("Passwords entered don't match.");
            } else {
                alert("Password Validated")
                document.getElementById("passwordForm").reset();
            }
        }
    </script>
</body>

</html>
```

OUTPUT




File /home/nithin/Codes/Sem4/IT254/Lab/Lab_2/3.html
This page says
Password must be at least 8 characters long.
OK
Enter password:
Confirm password:
Validate


File /home/nithin/Codes/Sem4/IT254/Lab/Lab_2/3.html
This page says
Passwords entered don't match.
OK
Enter password:
Confirm password:
Validate


File
/home/nithin/Codes/Sem4/IT254/Lab/Lab_2/3.html
This page says
Password Validated
OK
Enter password:
Confirm password:
Validate