Web-Based Interface Design

(CSF 3133)

LAB REPORT 6

NAME: WONG CAI YI

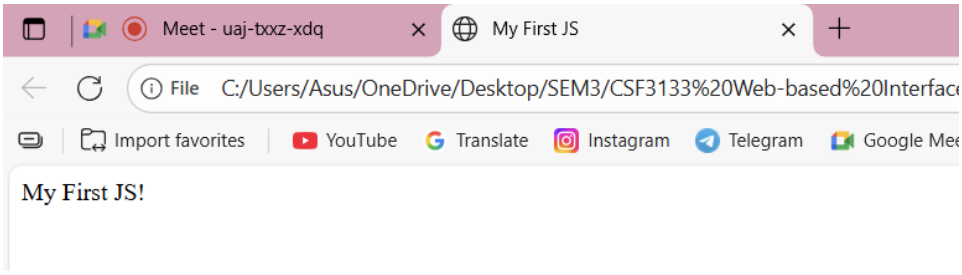
MATRIC NO: S76084

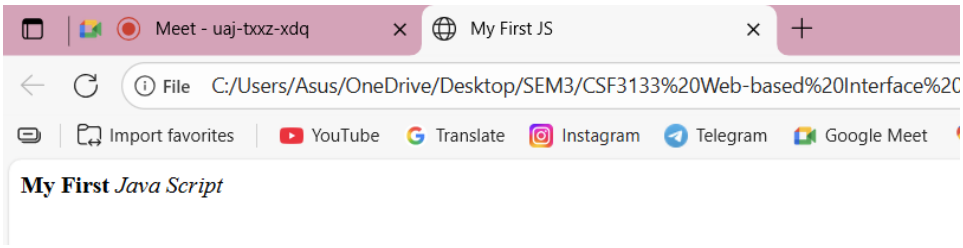
DATE: 03 December 2025

GROUP: K4

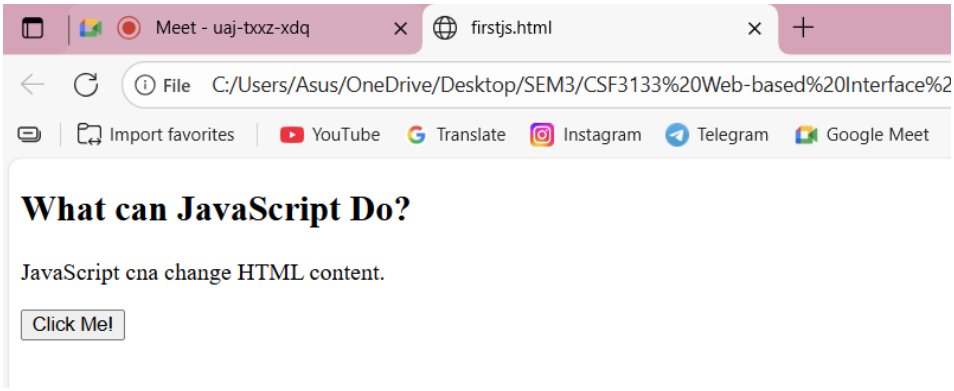
PROGRAMME: COMPUTER SCIENCE (SOFTWARE ENGINEERING)

6.3 Basic JavaScript

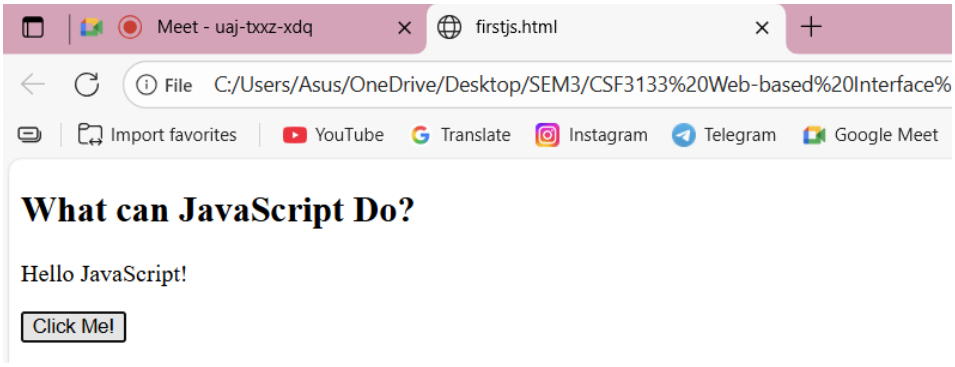




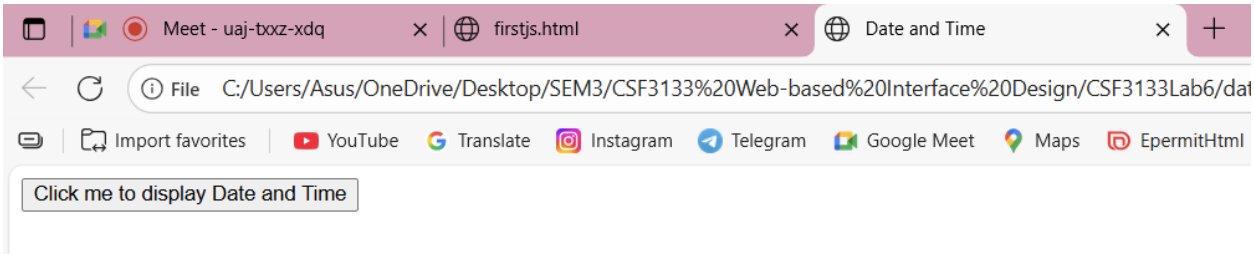
6.4 getElementById()



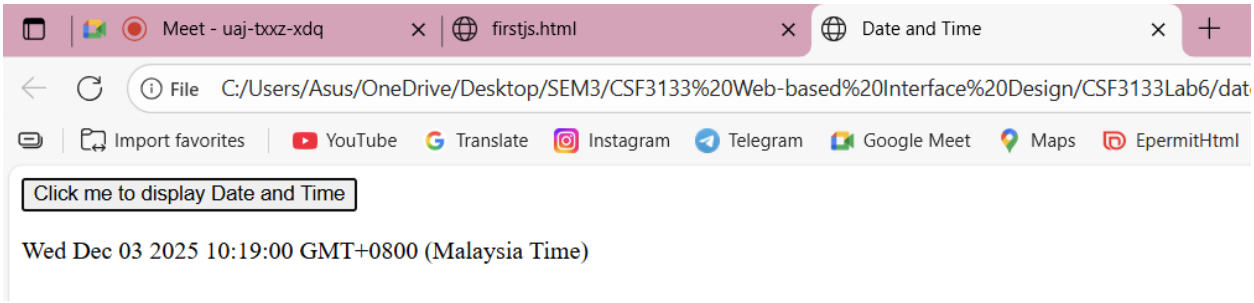
After click the click me button,



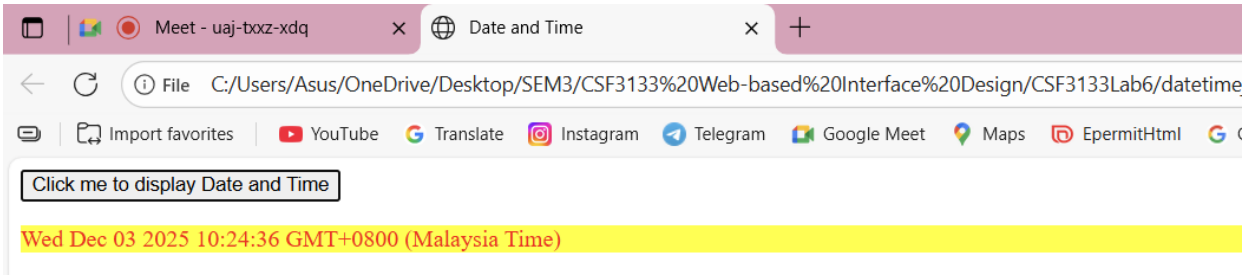
6.5 Date and Time



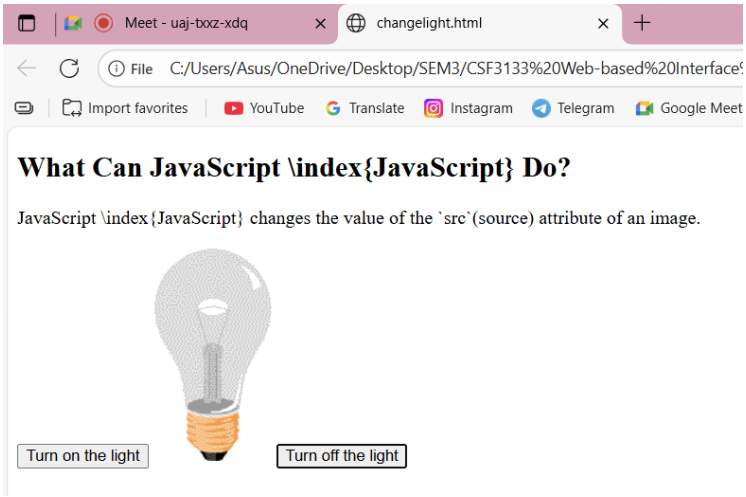
After click the button,



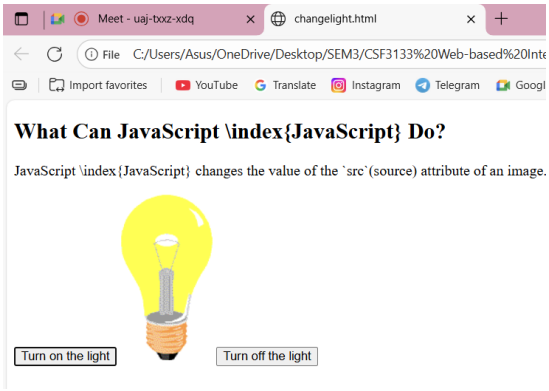
After add the script and click the button,



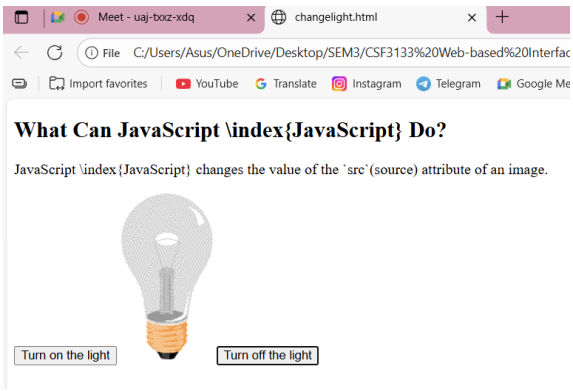
6.6 Change HTML Attribute Values



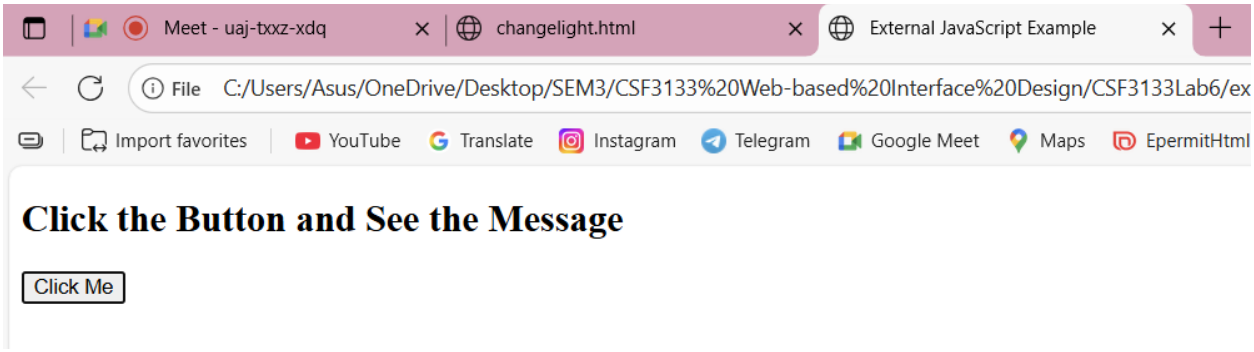
When click the turn on the light button,



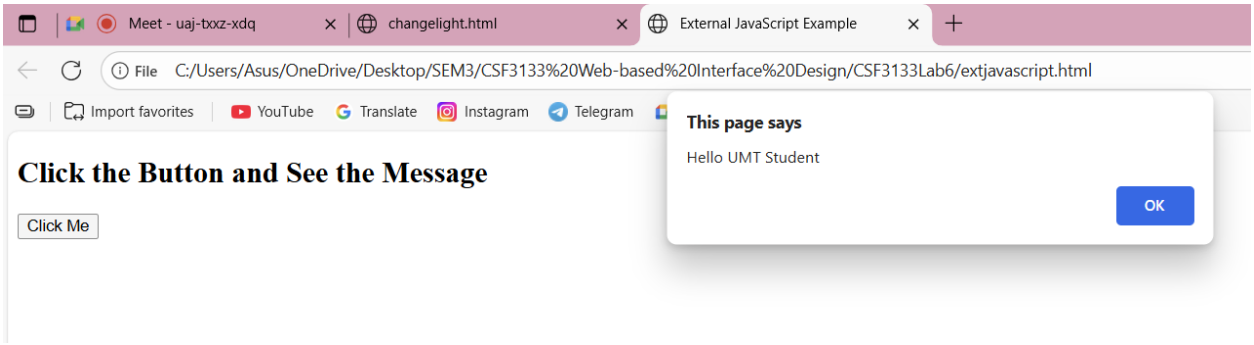
When click the turn off the light button,



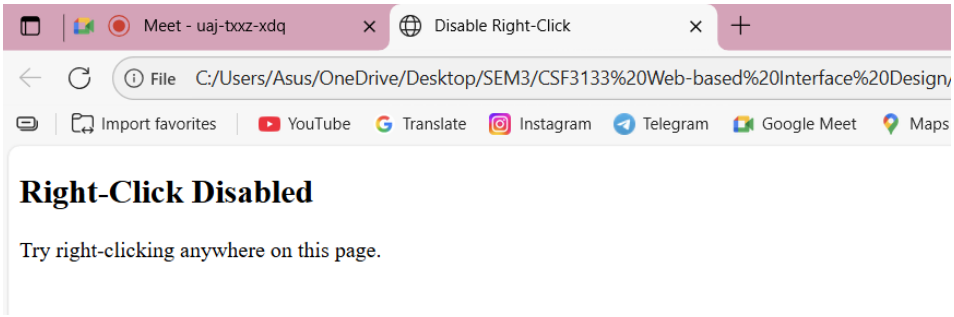
6.7 Link to file JavaScript



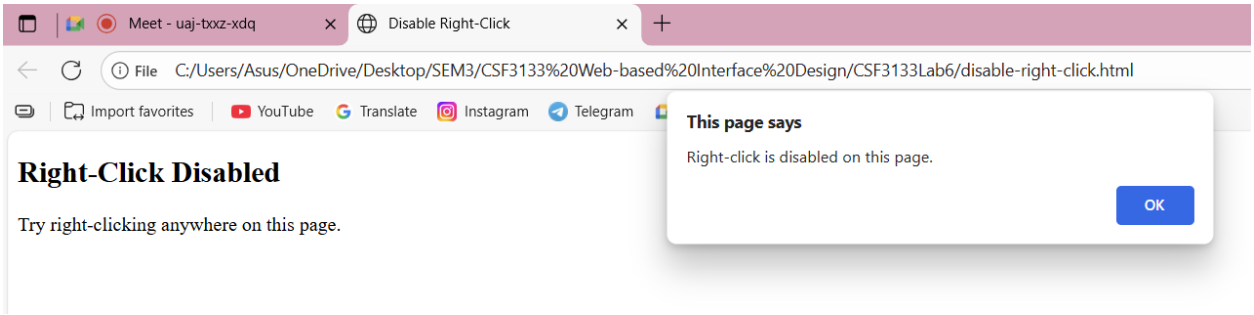
After click button,



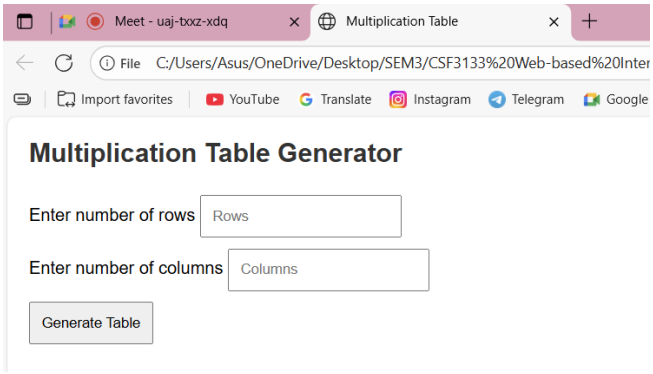
6.8 Working with events

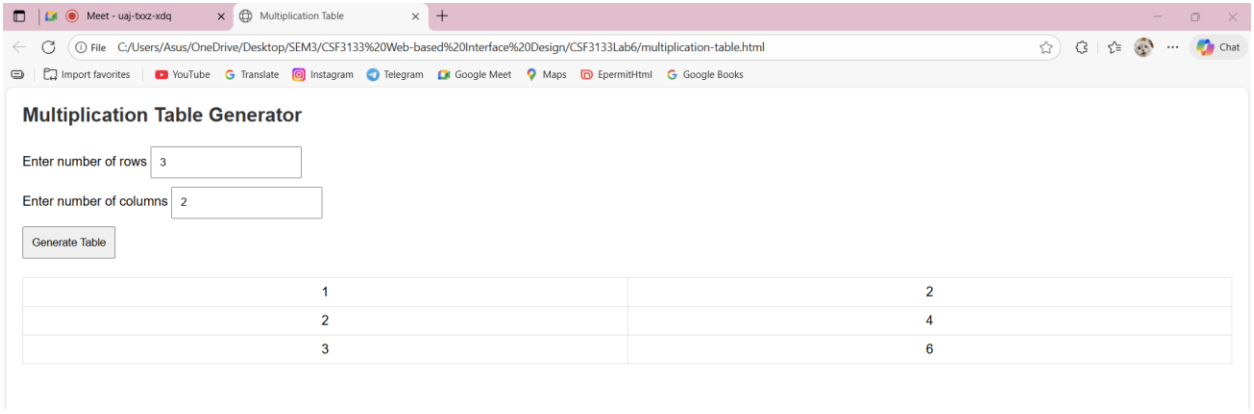


When I right-click in the page,



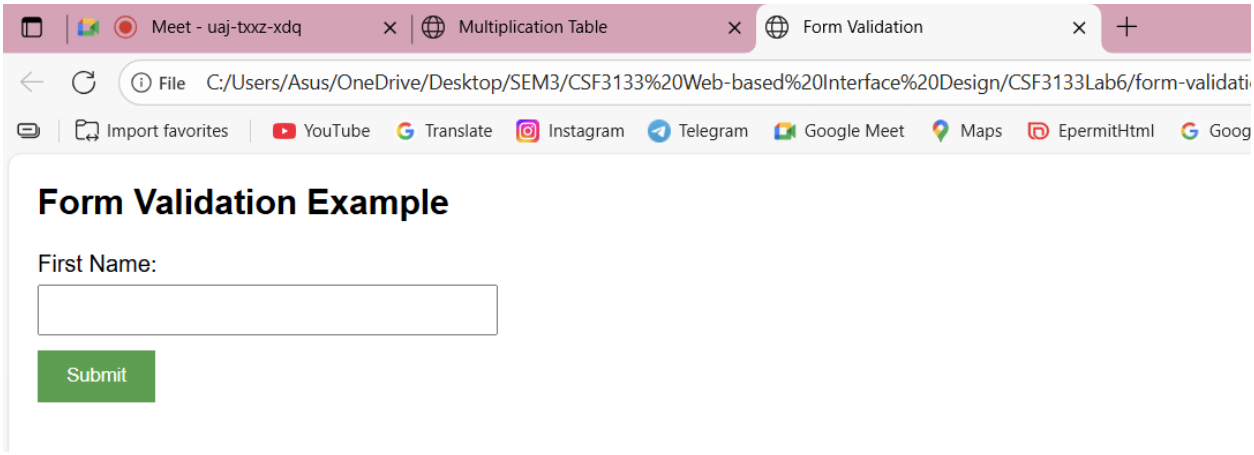
6.9 Multiplication Table



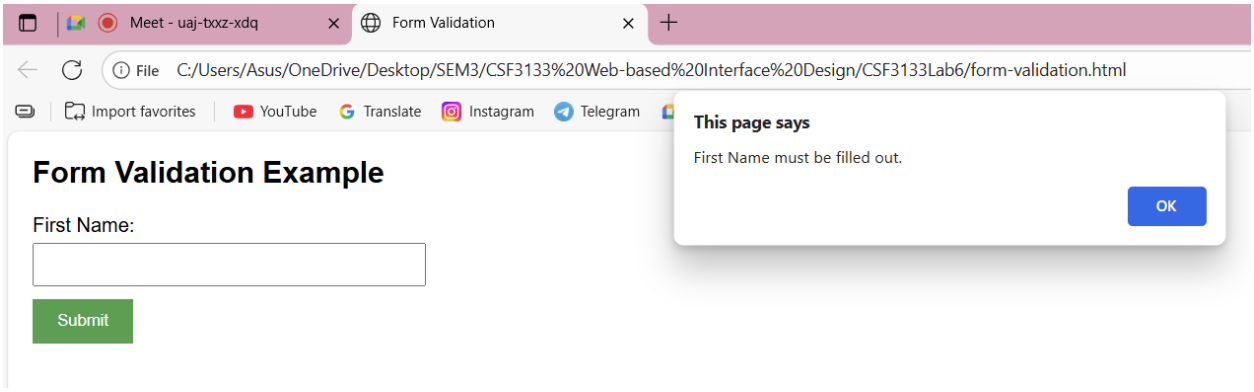


6.10 Client-side validation

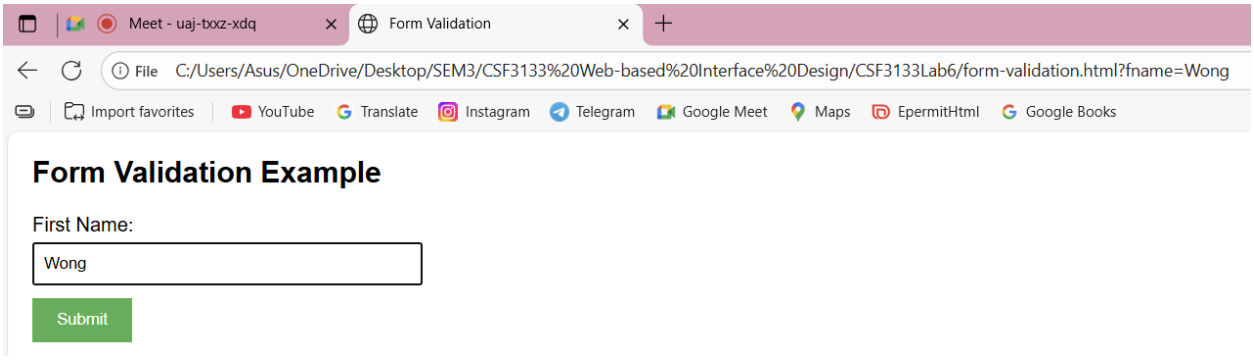
6.10.1 Form Validation



If I left the first name empty and submit,

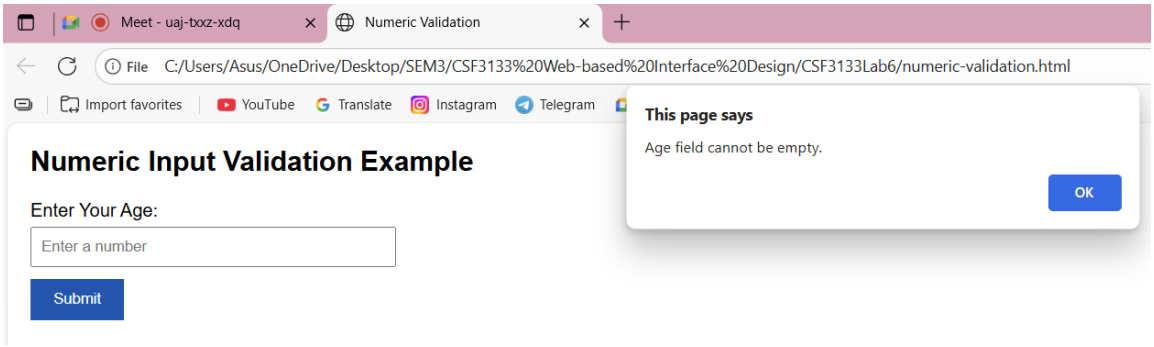


Else, successful (changed in website address)

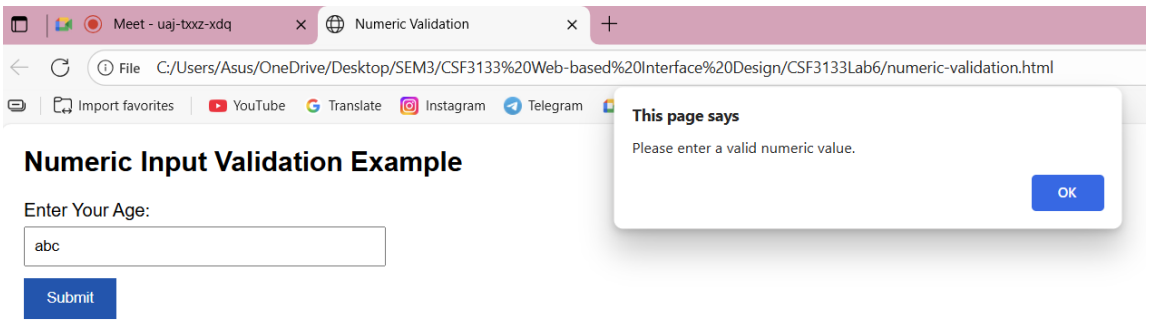


6.10.2 Numeric Input Validation

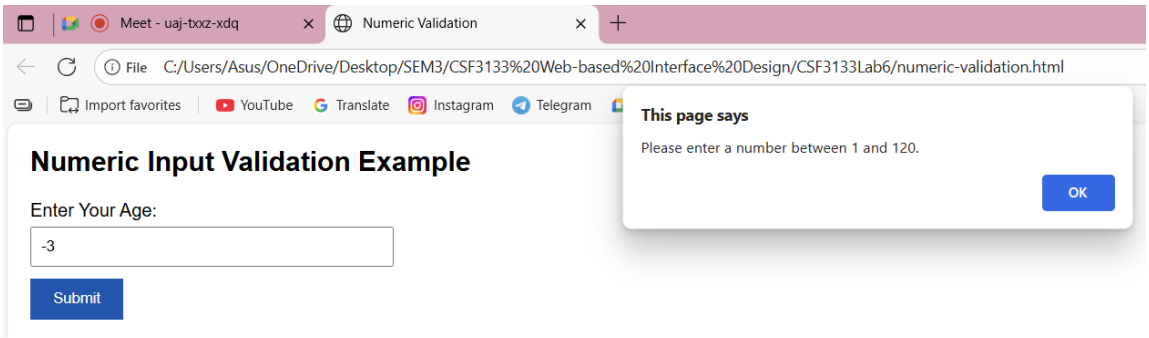
Case1: Empty input



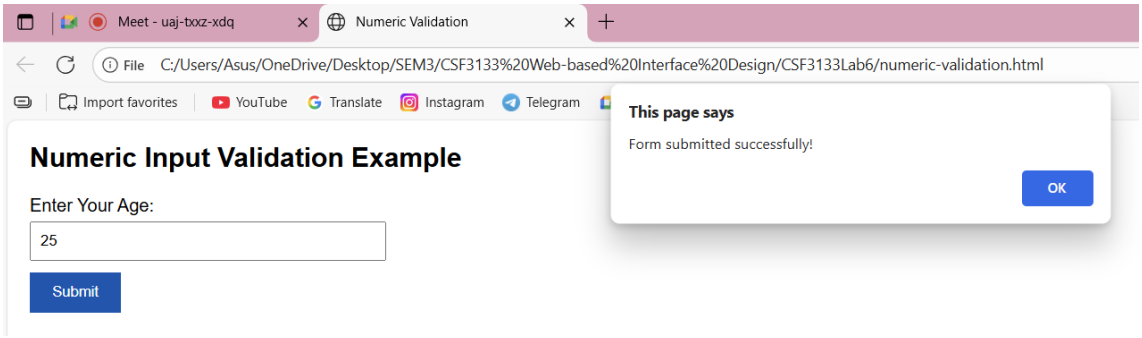
Case2: Non-numeric value



Case3: Number outside range

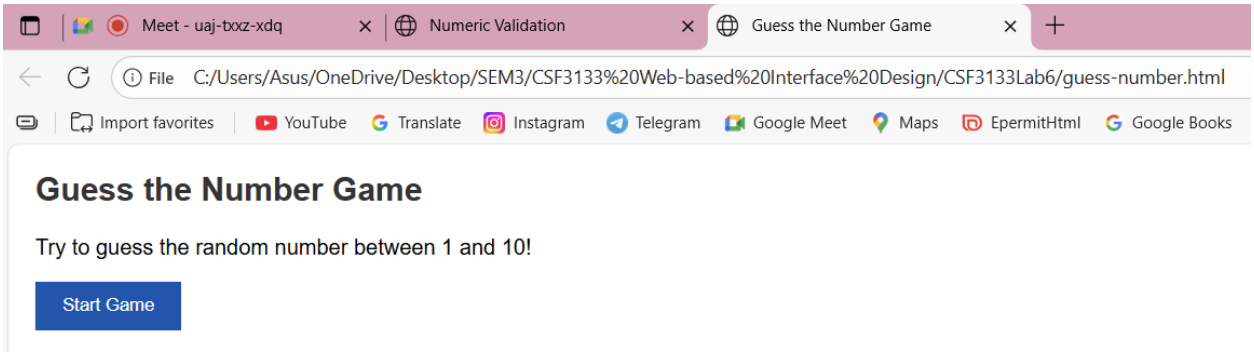


Case4: Valid number

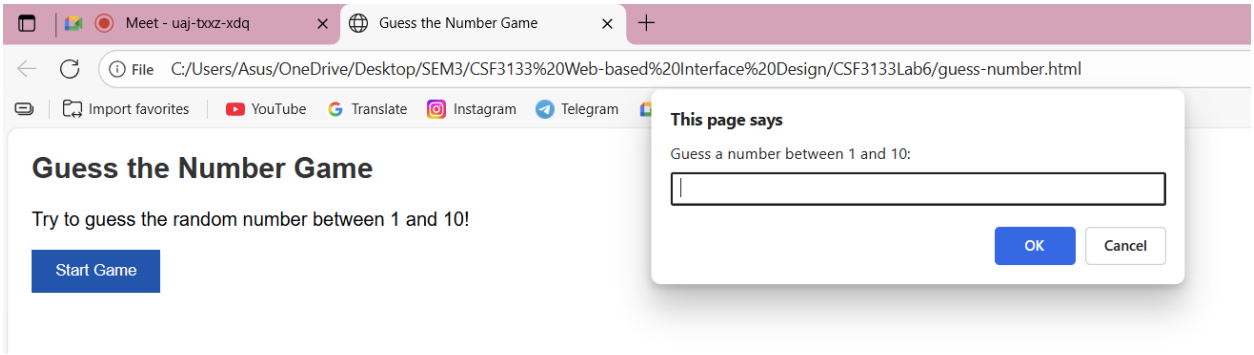


6.11 Math Function

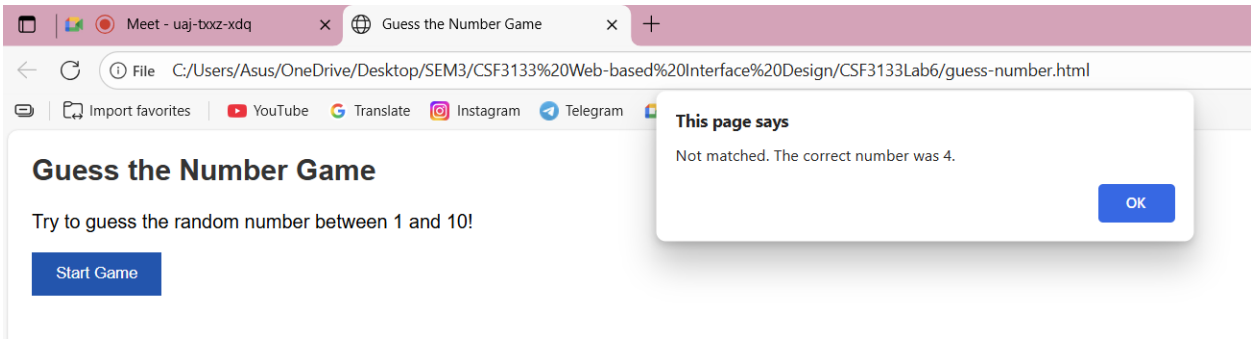
6.11.1 Guess a Number



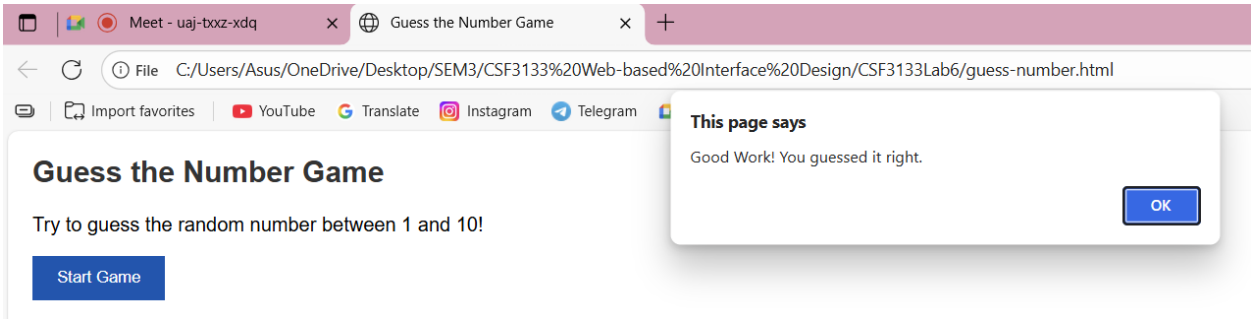
After click button,



Not matched.



Matched.



6.12 Lab Exercise 6

6.12.1 Quiz Application

LabExercise > index.html > html > body > header.header > h1

```html
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <link rel="stylesheet" href="style.css">
    <script src="script.js" defer></script>
    <title>Queue Quiz</title>
</head>
<body>

    <header class="header">
        <h1>Queue Quiz</h1>
        <p class="subtitle">
            This quiz is designed to test your understanding of the queue data structure,
            including its concepts, operations, and applications. <br>
            Answer each question carefully and apply what you have learned about how queues
            work in real-life and computer systems.
        </p>
    </header>

    <div class="quiz-container">
        <div class="timer">
            Time Left: <span id="time">30</span>s
        </div>

        <div class="question" id="question">
            Question will appear here
        </div>

        <div class="options" id="options"></div>

        <p id="result"></p>

        <div class="result">
            Your score: <span id="score">0</span>
        </div>
```

```html

        <button class="restart-btn" id="restartBtn">Restart Quiz</button>
    </div>

</body>
</html>


```

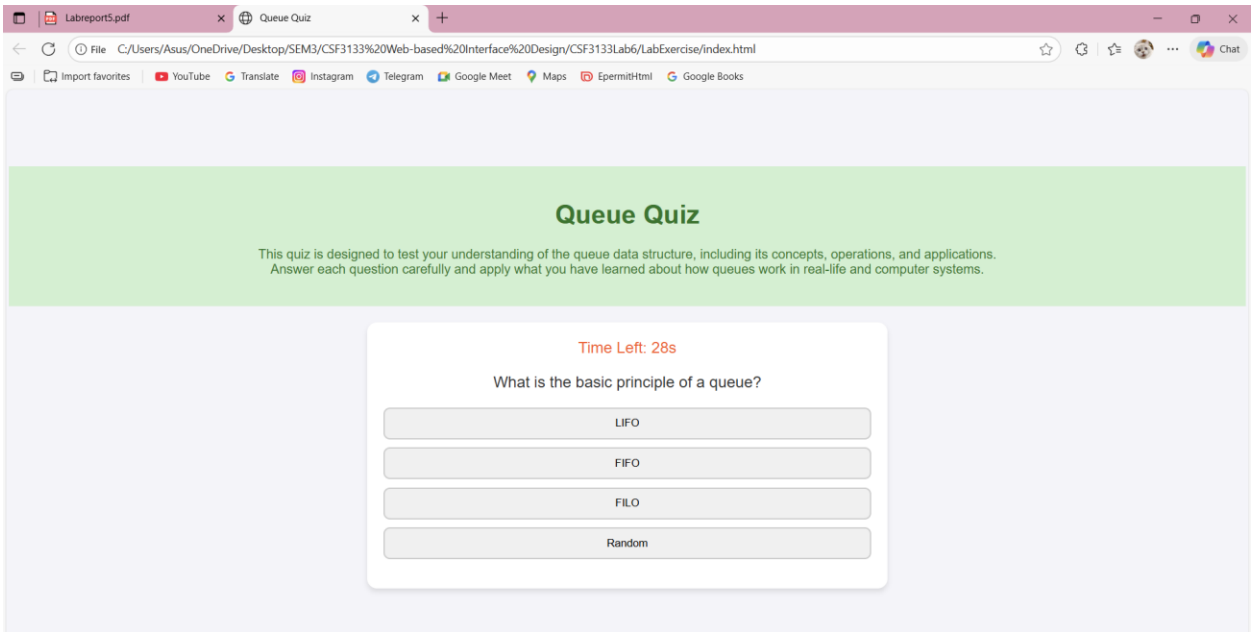
```css
/* Header */
header {
    text-align: center;
    background-color: rgb(207, 240, 207);
    color: #237626;
    padding: 20px;
    width: 97%;
}

/* Page Layout */
body {
    font-family: Arial, sans-serif;
    background-color: #f4f4f9;
    color: #333;
    margin: 0;
    padding: 0;

    display: flex;
    justify-content: center;
    align-items: center;

    height: 100vh;
    flex-direction: column;
}

/* Quiz Box */
.quiz-container {
    background: #fff;
    border-radius: 12px;
    box-shadow: 0 4px 6px rgba(0, 0, 0, 0.1);

    width: 90%;
    max-width: 600px;
    padding: 20px;

    text-align: center;
    margin-top: 20px;
}

/* Question Text */
.question {
    font-size: 1.2em;
    margin-bottom: 20px;
}

/* Options */
.options {
    display: flex;
    flex-direction: column;
    gap: 10px;
}

.option {
    padding: 10px;
    border: 2px solid #ccc;
    border-radius: 8px;
    cursor: pointer;
    transition: 0.3s;
}

/* Hover Effect */
.option:hover {
    background-color: #007bff;
    color: #fff;
}

/* Timer */
.timer {
```

```
    font-size: 1.2em;
    margin-bottom: 20px;
    color: #ff5722;
}

/* Score Result */
.result {
    font-size: 1.5em;
    color: #4caf50;
    display: none;
    margin-top: 15px;
}

/* Restart Button */
.restart-btn {
    background-color: #007bff;
    color: #fff;

    border: none;
    padding: 10px 20px;
    font-size: 1em;

    border-radius: 8px;
    cursor: pointer;

    margin-top: 20px;
    display: none;
}

.restart-btn:hover {
    background-color: #0056b3;
}
```

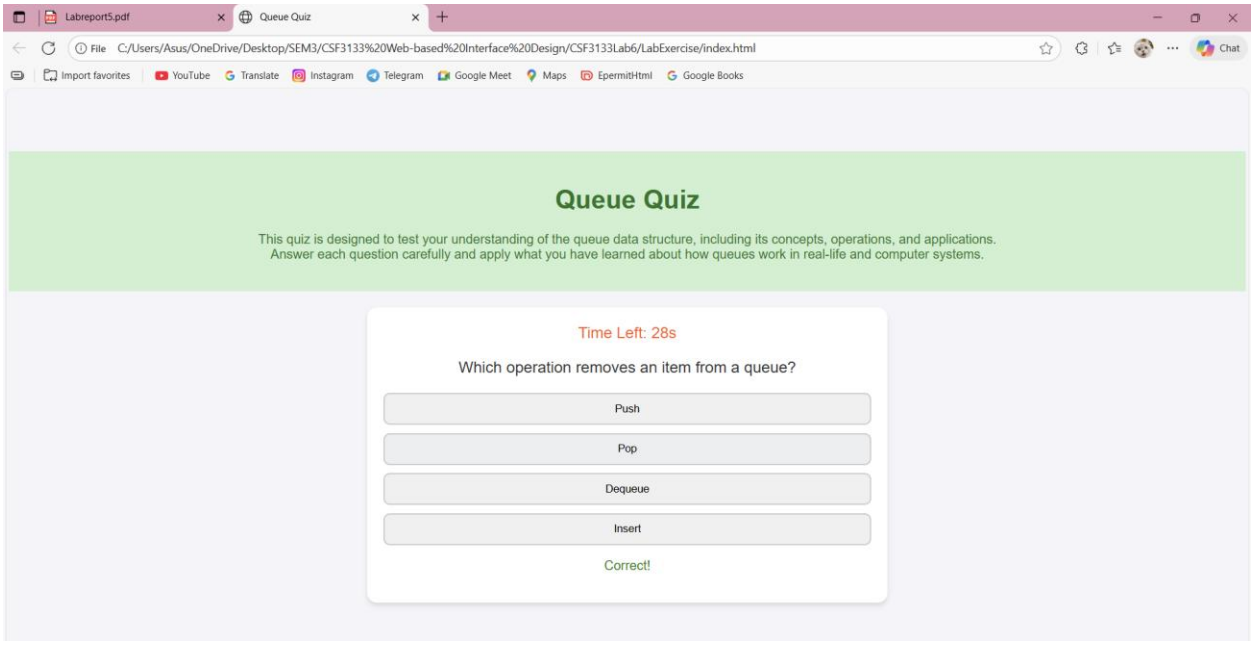
LabExercise > JS script.js > quizData

```
const quizData = [
        {
            question: "What is the basic principle of a queue?",
            options: ["LIFO", "FIFO", "FILO", "Random"],
            answer: "FIFO"
        },
        {
            question: "Which operation removes an item from a queue?",
            options: ["Push", "Pop", "Dequeue", "Insert"],
            answer: "Dequeue"
        },
        {
            question: "Which operation adds an item to a queue?",
            options: ["Shift", "Enqueue", "Pop", "Remove"],
            answer: "Enqueue"
        },
        {
            question: "Where is an element inserted in a queue?",
            options: ["Front", "Middle", "Rear", "Anywhere"],
            answer: "Rear"
        },
        {
            question: "Where is an element removed from a queue?",
            options: ["Front", "Back", "Middle", "Bottom"],
            answer: "Front"
        },
        {
            question: "Which of the following is NOT a type of queue?",
            options: ["Circular Queue", "Priority Queue", "Double Queue", "Simple Queue"],
            answer: "Double Queue"
        },
        {
            question: "Which data structure is best for task scheduling?",
            options: ["Tree", "Queue", "Stack", "Graph"],
            answer: "Queue"
        },
        {
            question: "What happens when a queue is full?",
            options: ["Underflow", "Overflow", "Crash", "Reset"],
            answer: "Overflow"
        }
    ];

    let currentQuestion = 0;
    let score = 0;
    let timeLeft = 30;
    let timerInterval;
    const timerEl = document.getElementById('time');
    const questionEl = document.querySelector('.question');
    const optionsEl = document.querySelector('.options');
    const resultEl = document.querySelector('.result');
    const scoreEl = document.getElementById('score');
    const restartBtn = document.querySelector('.restart-btn');

    // Function to load the question
    function loadQuestion() {
        if (currentQuestion >= quizData.length) {
            endQuiz();
            return;
        }
        clearInterval(timerInterval);
        timeLeft = 30;
        timerEl.textContent = timeLeft;
        startTimer();
        const currentQuiz = quizData[currentQuestion];
        questionEl.textContent = currentQuiz.question;
        optionsEl.innerHTML = ''; // Clear previous options
        currentQuiz.options.forEach(option => {
            const button = document.createElement('button');
            button.classList.add('option');
            button.textContent = option;
```

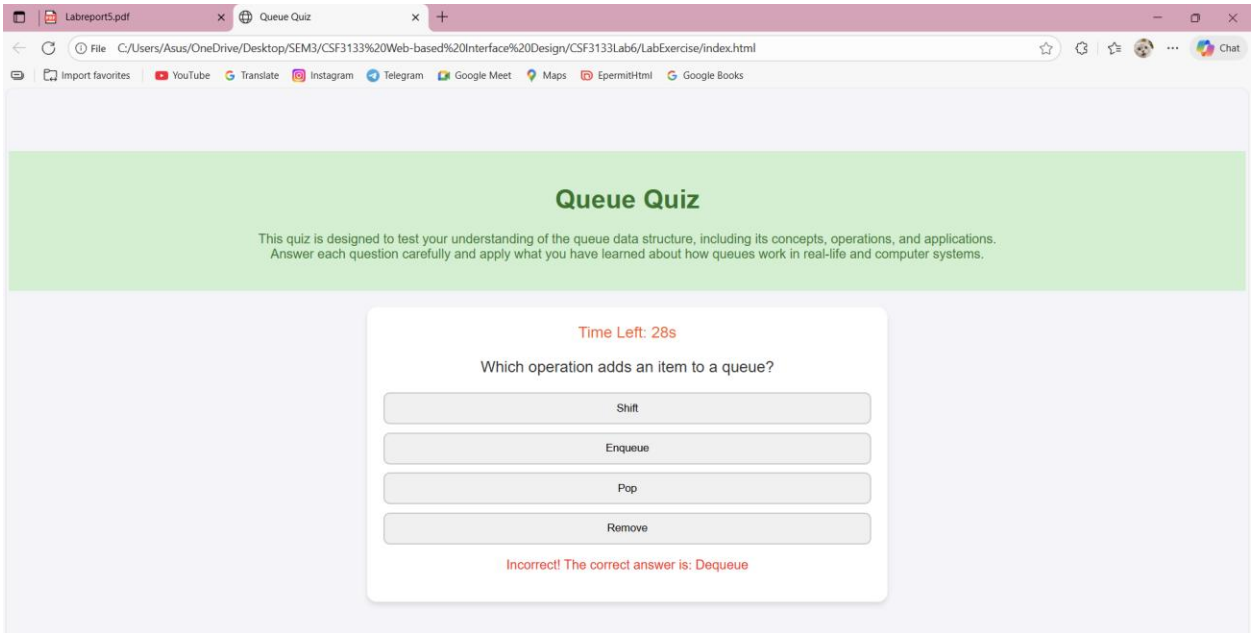
```
            button.onclick = () => checkAnswer(option);
            optionsEl.appendChild(button);
        });
    }

    //check ans
    function checkAnswer(selectedOption) {
        let result = document.getElementById("result");
        let correctAnswer = quizData[currentQuestion].answer;

        if (selectedOption === correctAnswer) {
            result.textContent = "Correct!";
            result.style.color = "green";
            score++;
        } else {
            result.textContent = "Incorrect! The correct answer is: " + correctAnswer;
            result.style.color = "red";
        }

        currentQuestion++;
        loadQuestion();
    }

    // Start the timer
    function startTimer() {
        timerInterval = setInterval(() => {
            timeLeft--;
            timerEl.textContent = timeLeft;
            if (timeLeft <= 0) {
                clearInterval(timerInterval);
                endQuiz();
            }
        }, 1000);
    }

    // End the quiz and show the results
    function endQuiz() {
        clearInterval(timerInterval);
        questionEl.style.display = 'none';
        optionsEl.style.display = 'none';
        resultEl.style.display = 'block';
        scoreEl.textContent = score;
        restartBtn.style.display = 'block';
    }

    // Restart the quiz
    restartBtn.addEventListener('click', () => {
        // Reset variables
        currentQuestion = 0;
        score = 0;
        timeLeft = 30;
        timerEl.textContent = timeLeft;

        // Reset the display
        questionEl.style.display = 'block';
        optionsEl.style.display = 'flex'; // Ensure options are displayed correctly
        resultEl.style.display = 'none';
        restartBtn.style.display = 'none';

        // Load the first question
        loadQuestion();
    });

    // Initialize the quiz with the first question
loadQuestion();
```



If correct, the correct will show below the option.



If the answer is incorrect, the incorrect will show below and the correct answer also will show.



Final, the website will show the overall score.