



Web-Based Interface Design

(CSF 3133)

LAB REPORT 6

NAME: WONG CAI YI

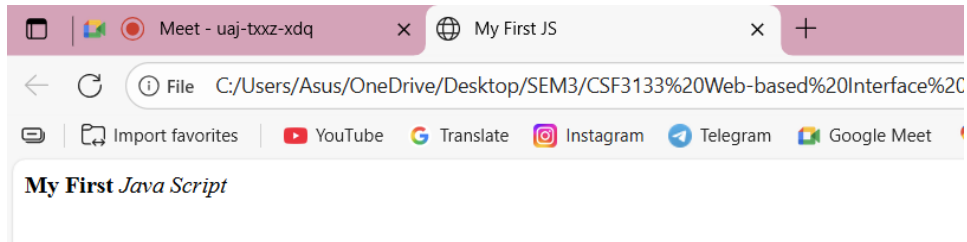
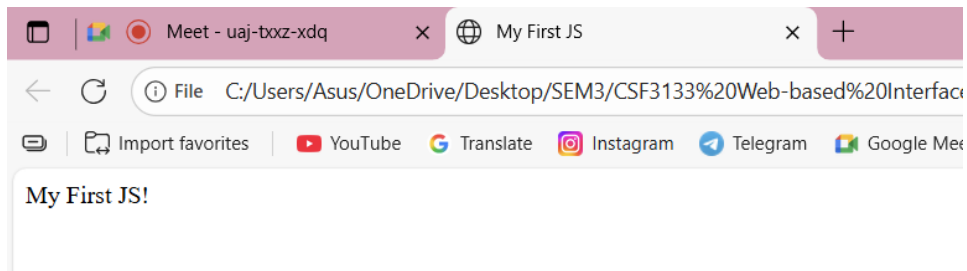
MATRIC NO: S76084

DATE: 03 December 2025

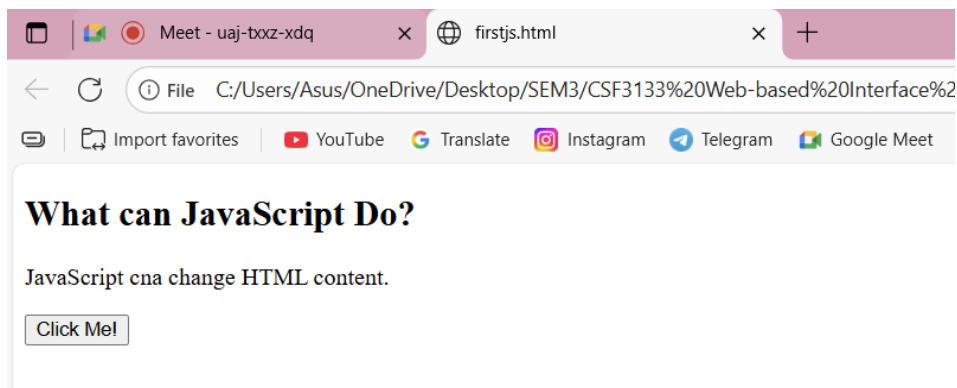
GROUP: K4

PROGRAMME: COMPUTER SCIENCE (SOFTWARE ENGINEERING)

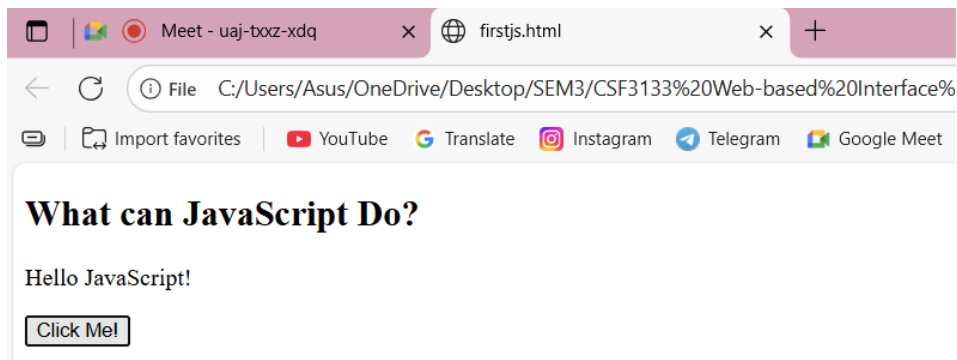
6.3 Basic JavaScript



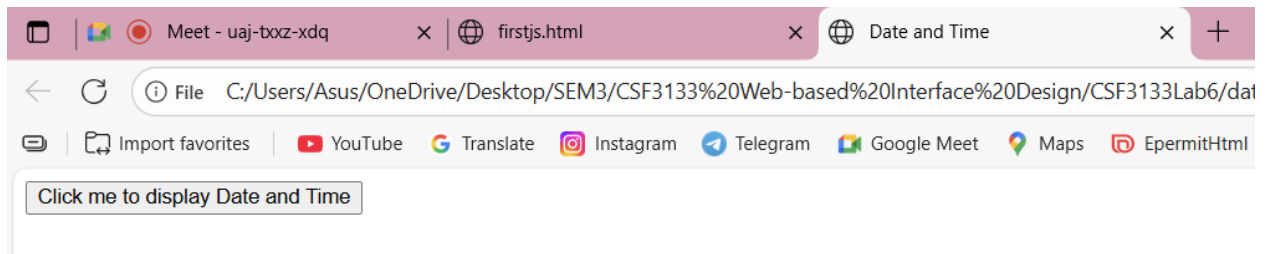
6.4 getElementById()



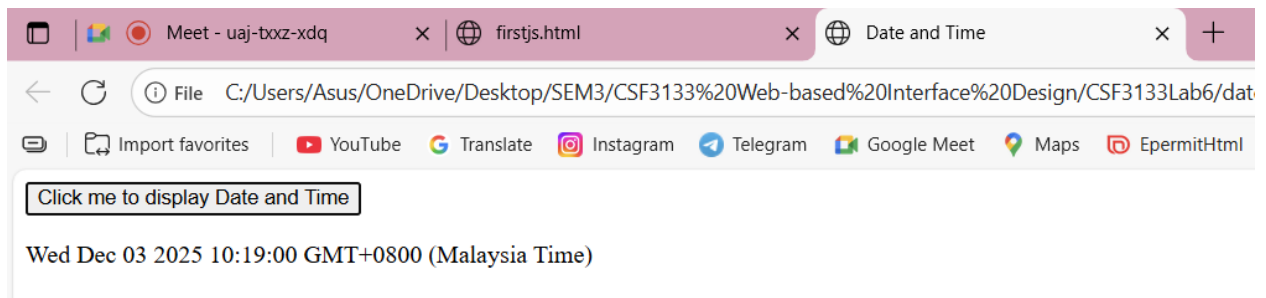
After click the click me button,



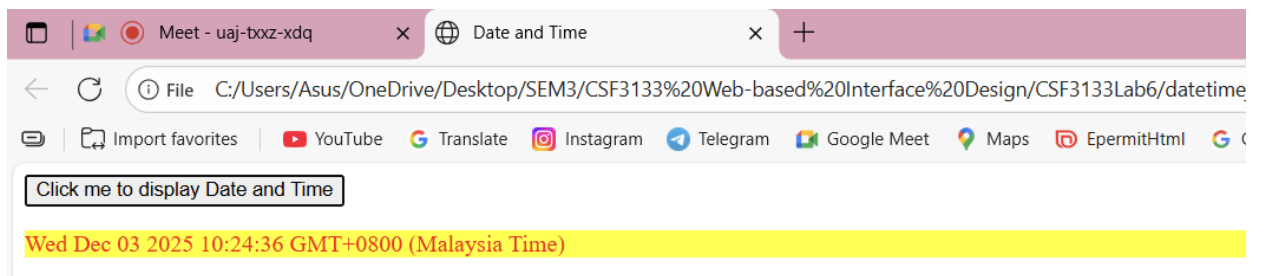
6.5 Date and Time



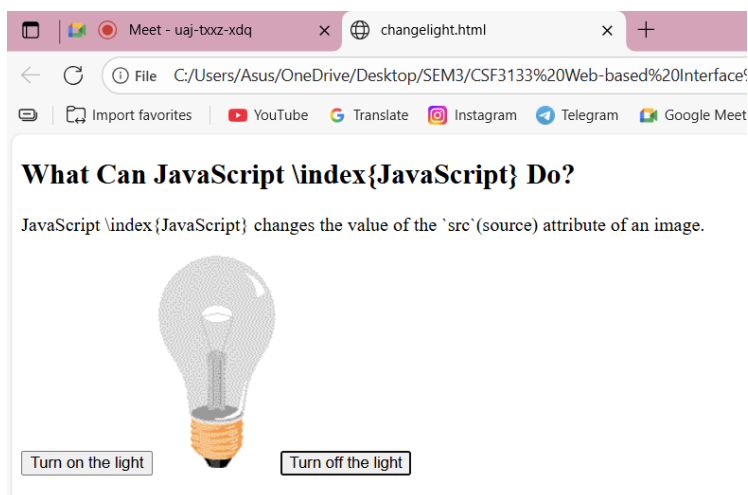
After click the button,



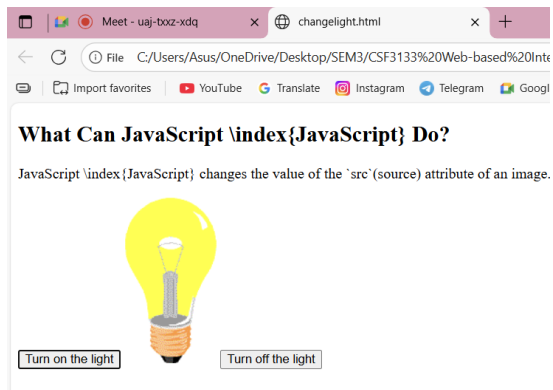
After add the script and click the button,



6.6 Change HTML Attribute Values



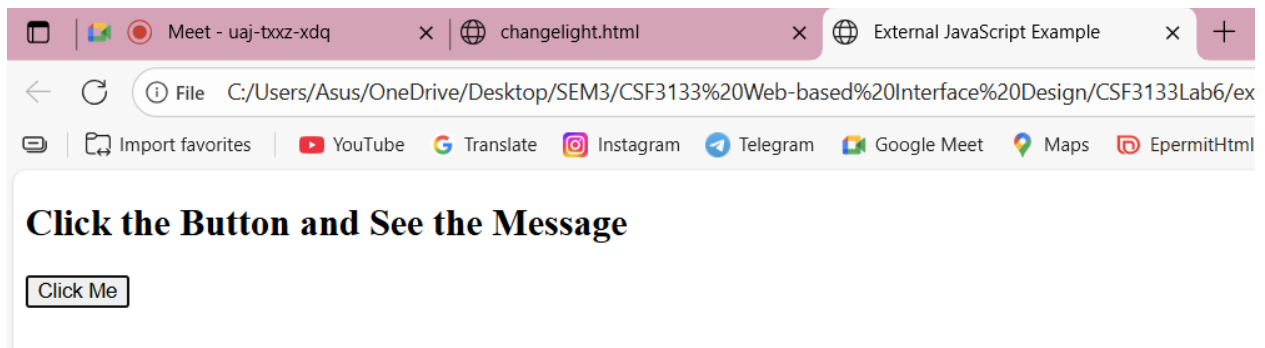
When click the turn on the light button,



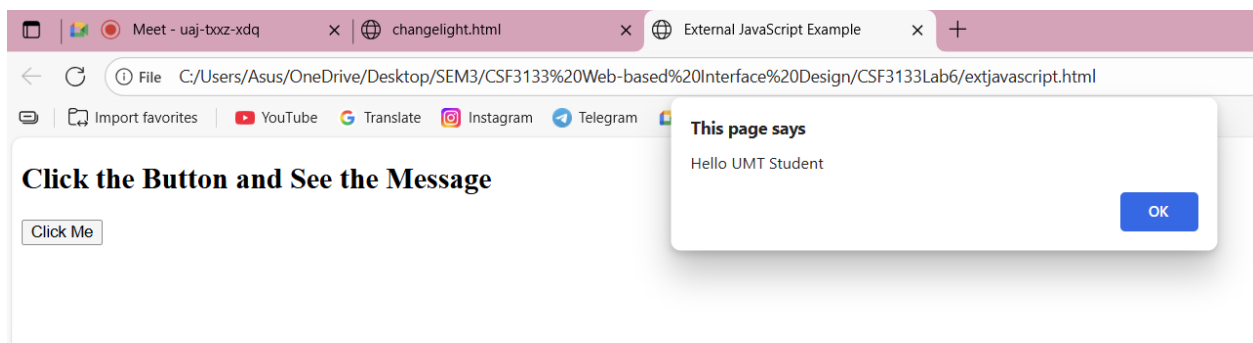
When click the turn off the light button,



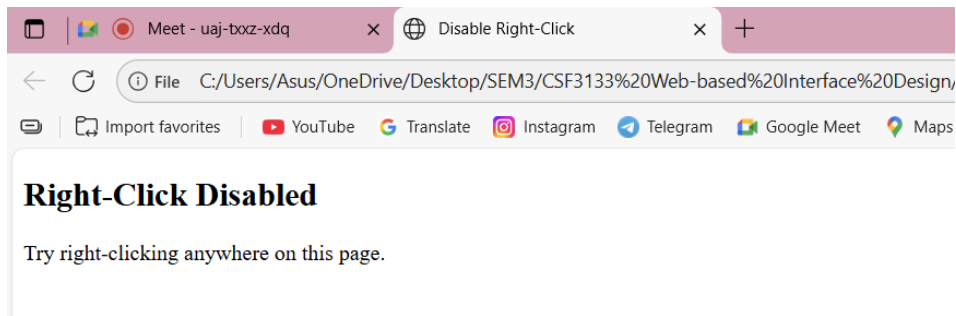
6.7 Link to file JavaScript



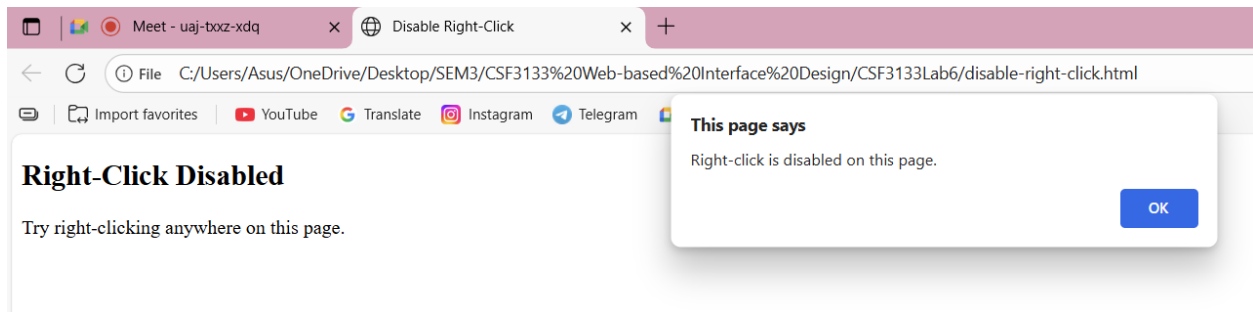
After click button,



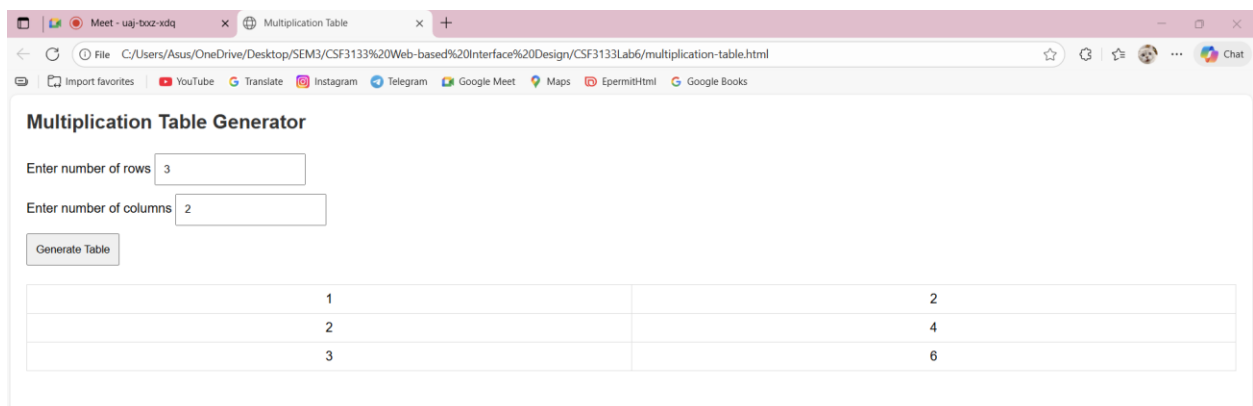
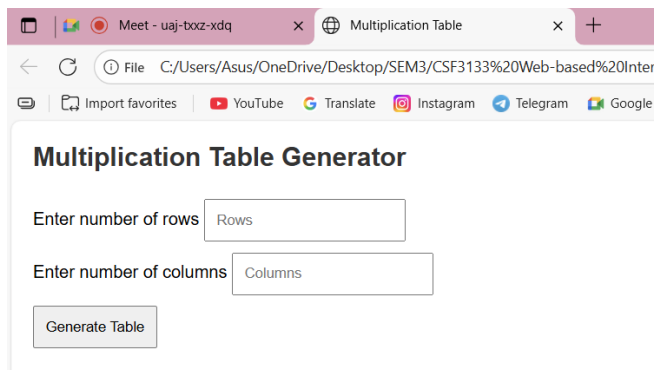
6.8 Working with events



When I right-click in the page,

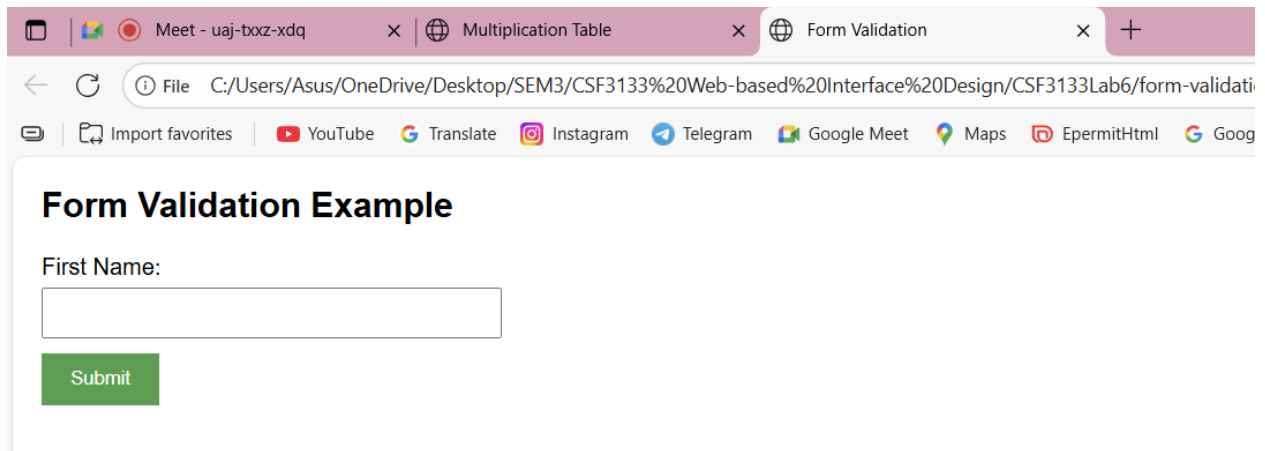


6.9 Multiplication Table



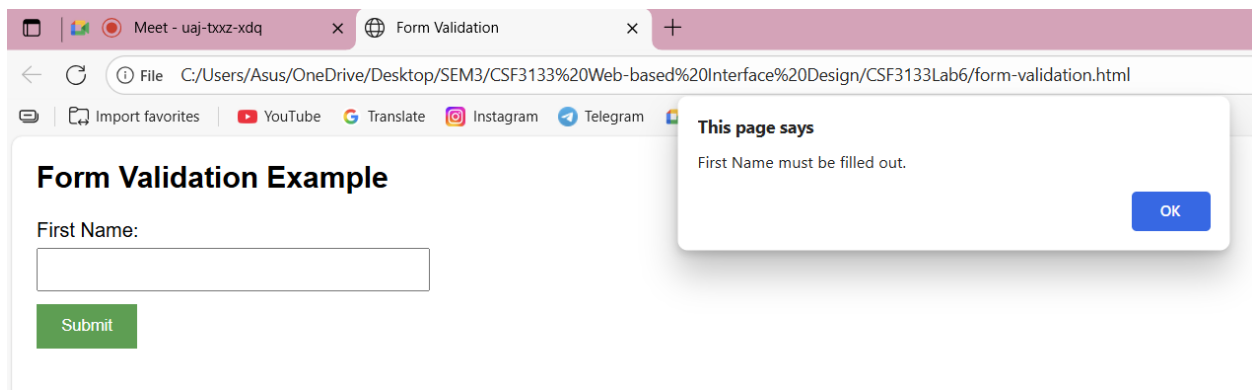
6.10 Client-side validation

6.10.1 Form Validation



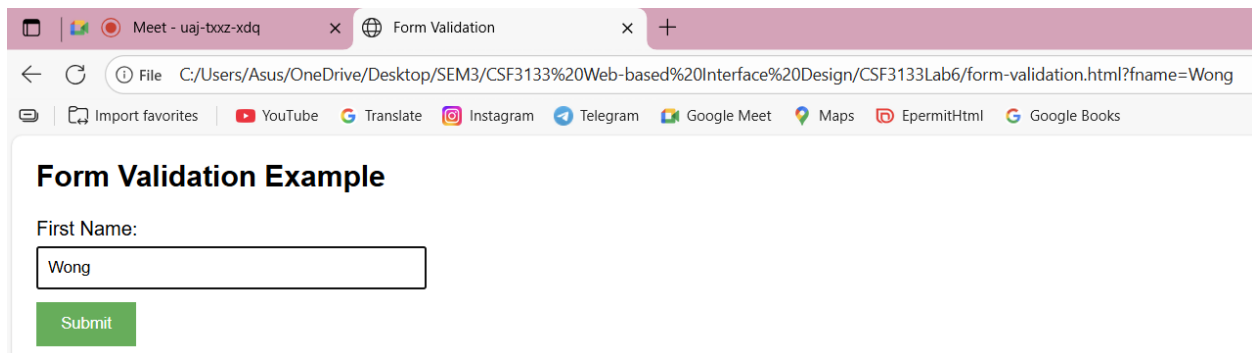
The screenshot shows a web browser with three tabs: "Meet - uaj-bxxz-xdq", "Multiplication Table", and "Form Validation". The address bar shows the file path: `C:/Users/Asus/OneDrive/Desktop/SEM3/CSF3133%20Web-based%20Interface%20Design/CSF3133Lab6/form-validati`. The page title is "Form Validation Example". Below the title, there is a label "First Name:" followed by an empty text input field. Below the input field is a green "Submit" button.

If I left the first name empty and submit,



The screenshot shows the same web browser as before, but now a validation error message is displayed. The address bar shows the full file path: `C:/Users/Asus/OneDrive/Desktop/SEM3/CSF3133%20Web-based%20Interface%20Design/CSF3133Lab6/form-validation.html`. The page title is "Form Validation Example". The "First Name:" label and empty input field are still present. A green "Submit" button is also visible. A white dialog box with a blue border is overlaid on the right side of the page. It has the title "This page says" and the message "First Name must be filled out." with an "OK" button.

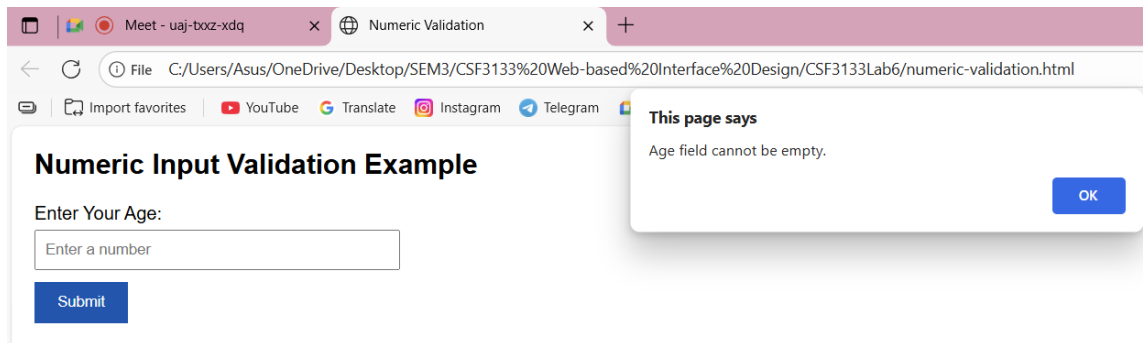
Else, successful (changed in website address)



The screenshot shows the same web browser as before, but now the form has been successfully submitted. The address bar shows the file path with the first name parameter: `C:/Users/Asus/OneDrive/Desktop/SEM3/CSF3133%20Web-based%20Interface%20Design/CSF3133Lab6/form-validation.html?fname=Wong`. The page title is "Form Validation Example". The "First Name:" label is followed by a text input field containing the name "Wong". Below the input field is a green "Submit" button.

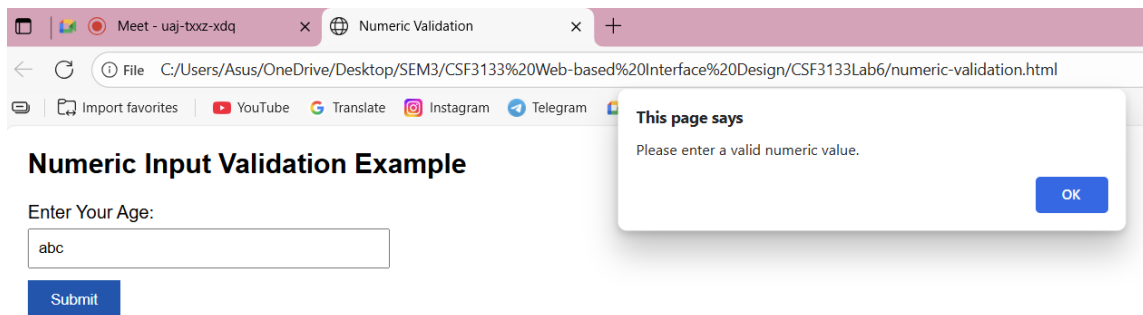
6.10.2 Numeric Input Validation

Case1: Empty input



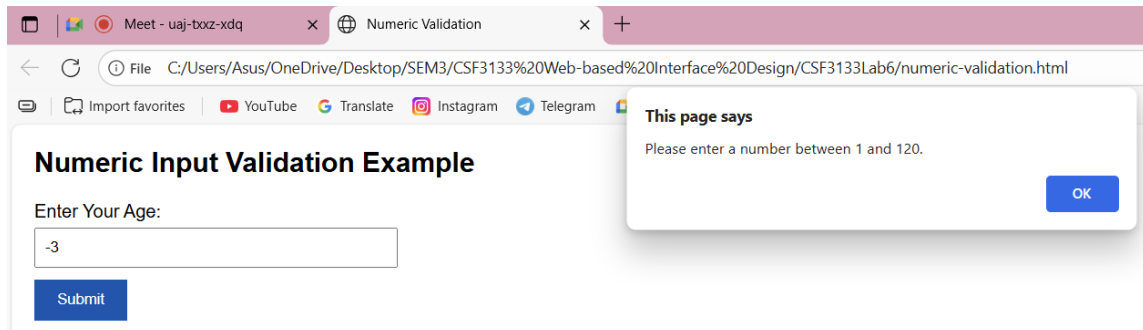
The screenshot shows a web browser window with two tabs: "Meet - uaj-bxz-xdq" and "Numeric Validation". The address bar shows the file path: "C:/Users/Asus/OneDrive/Desktop/SEM3/CSF3133%20Web-based%20Interface%20Design/CSF3133Lab6/numeric-validation.html". The page title is "Numeric Input Validation Example". Below the title, there is a label "Enter Your Age:" followed by a text input field containing the placeholder text "Enter a number". A blue "Submit" button is located below the input field. A white alert box with a blue border is displayed on the right side of the page, containing the text "This page says" and "Age field cannot be empty." with an "OK" button.

Case2: Non-numeric value



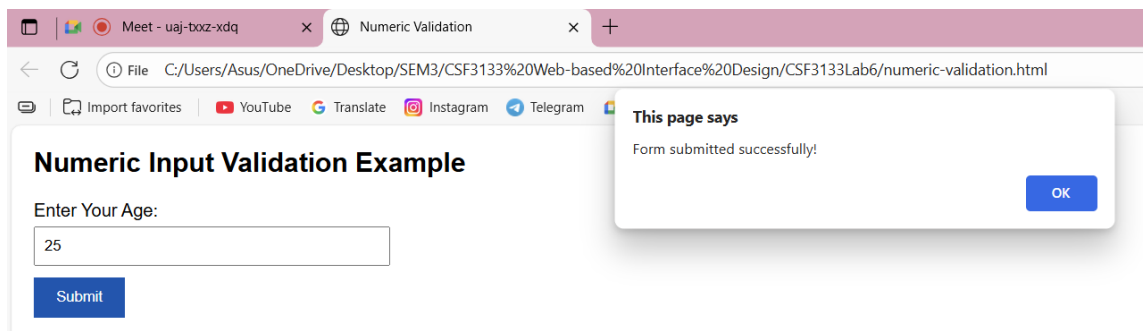
The screenshot shows the same web browser window as Case 1. The input field now contains the text "abc". A white alert box with a blue border is displayed on the right side of the page, containing the text "This page says" and "Please enter a valid numeric value." with an "OK" button.

Case3: Number outside range



The screenshot shows the same web browser window as Case 1. The input field now contains the text "-3". A white alert box with a blue border is displayed on the right side of the page, containing the text "This page says" and "Please enter a number between 1 and 120." with an "OK" button.

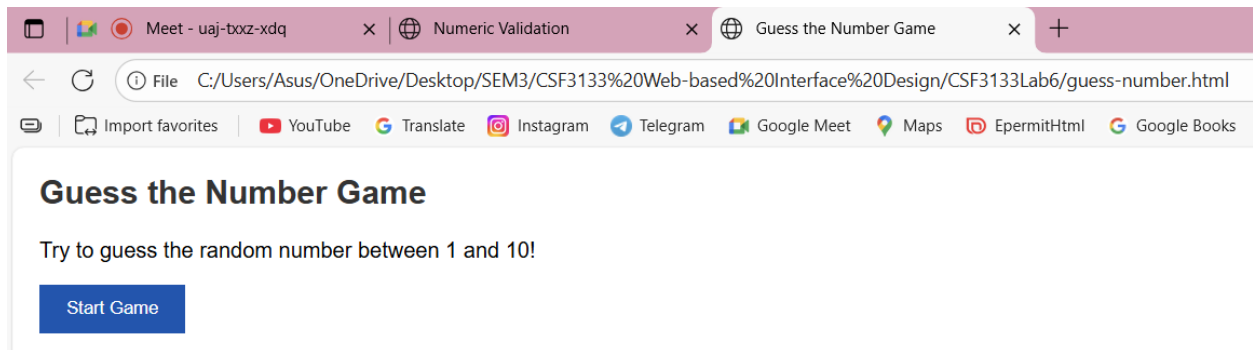
Case4: Valid number



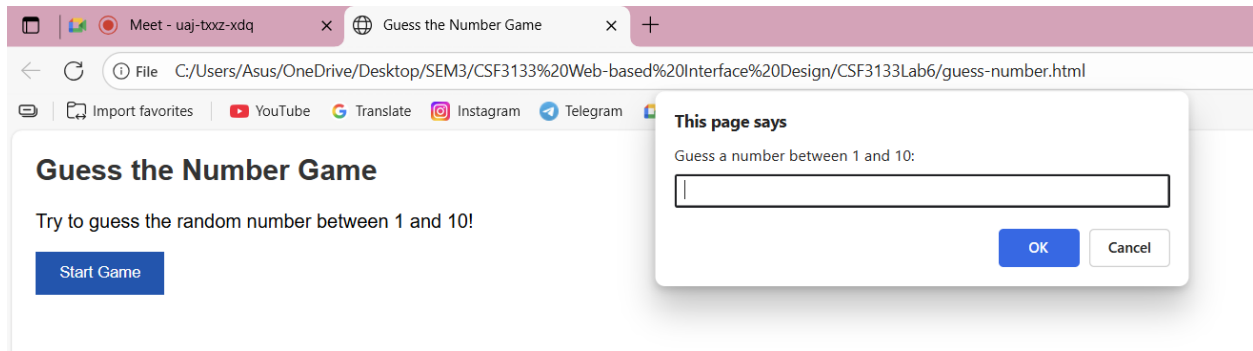
The screenshot shows the same web browser window as Case 1. The input field now contains the text "25". A white alert box with a blue border is displayed on the right side of the page, containing the text "This page says" and "Form submitted successfully!" with an "OK" button.

6.11 Math Function

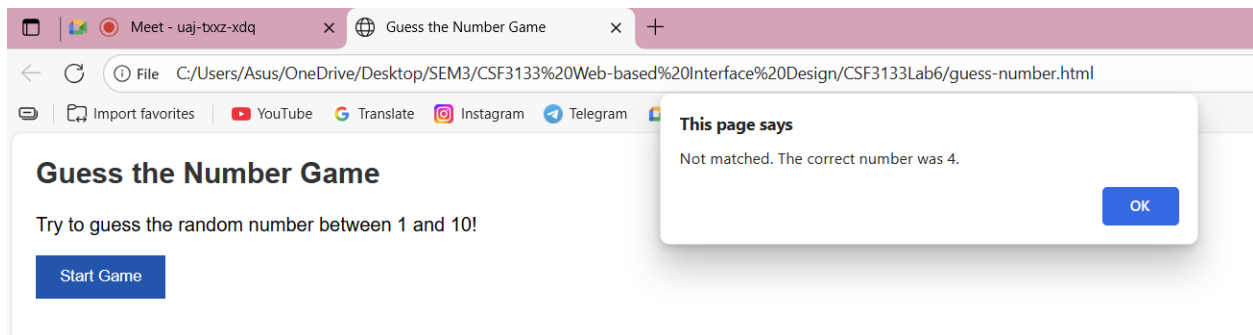
6.11.1 Guess a Number



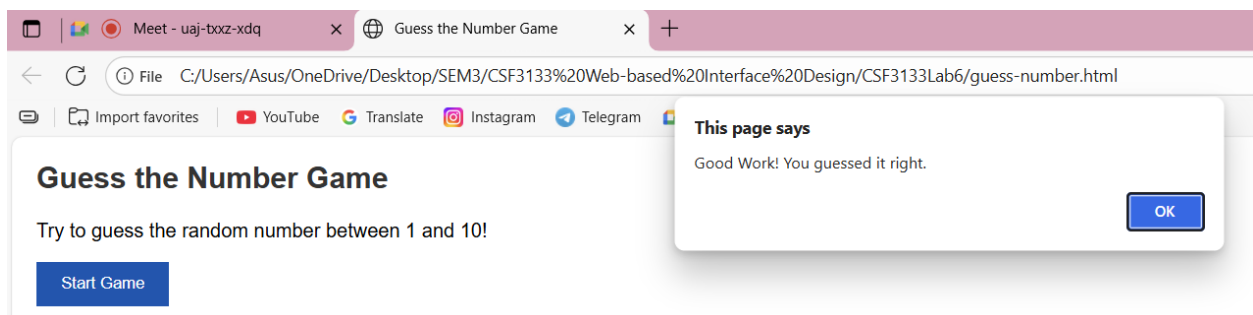
After click button,



Not matched.



Matched.



6.12 Lab Exercise 6

6.12.1 Quiz Application

```
LabExercise > index.html > html > body > header.header > h1
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <meta charset="UTF-8">
5      <meta name="viewport" content="width=device-width, initial-scale=1.0">
6      <link rel="stylesheet" href="style.css">
7      <script src="script.js" defer></script>
8      <title>Queue Quiz</title>
9  </head>
10 <body>
11
12     <header class="header">
13         <h1>Queue Quiz</h1>
14         <p class="subtitle">
15             This quiz is designed to test your understanding of the queue data structure,
16             including its concepts, operations, and applications. <br>
17             Answer each question carefully and apply what you have learned about how queues
18             work in real-life and computer systems.
19         </p>
20     </header>
21
22     <div class="quiz-container">
23         <div class="timer">
24             Time Left: <span id="time">30</span>s
25         </div>
26
27         <div class="question" id="question">
28             Question will appear here
29         </div>
30
31         <div class="options" id="options"></div>
32
33         <p id="result"></p>
34
35         <div class="result">
36             Your score: <span id="score">0</span>
37         </div>
38
39         <button class="restart-btn" id="restartBtn">Restart Quiz</button>
40     </div>
41
42 </body>
43 </html>
44
45
```

```

1  /* Header */
2  ✓ header {
3      text-align: center;
4      background-color: ■ rgb(207, 240, 207);
5      color: ■ #237626;
6      padding: 20px;
7      width: 97%;
8  }
9
10 /* Page Layout */
11 ✓ body {
12     font-family: Arial, sans-serif;
13     background-color: ■ #f4f4f9;
14     color: ■ #333;
15     margin: 0;
16     padding: 0;
17
18     display: flex;
19     justify-content: center;
20     align-items: center;
21
22     height: 100vh;
23     flex-direction: column;
24 }
25
26 /* Quiz Box */
27 ✓ .quiz-container {
28     background: ■ #fff;
29     border-radius: 12px;
30     box-shadow: 0 4px 6px ■ rgba(0, 0, 0, 0.1);
31
32     width: 90%;
33     max-width: 600px;
34     padding: 20px;
35
36     text-align: center;
37     margin-top: 20px;

```

```

38 }
39
40 /* Question Text */
41 .question {
42     font-size: 1.2em;
43     margin-bottom: 20px;
44 }
45
46 /* Options */
47 .options {
48     display: flex;
49     flex-direction: column;
50     gap: 10px;
51 }
52
53 .option {
54     padding: 10px;
55     border: 2px solid ■ #ccc;
56     border-radius: 8px;
57     cursor: pointer;
58     transition: 0.3s;
59 }
60
61 /* Hover Effect */
62 .option:hover {
63     background-color: ■ #007bff;
64     color: ■ #fff;
65 }
66
67 /* Timer */
68 .timer {

```

```
69     font-size: 1.2em;
70     margin-bottom: 20px;
71     color: ■ #ff5722;
72 }
73
74 /* Score Result */
75 .result {
76     font-size: 1.5em;
77     color: ■ #4caf50;
78     display: none;
79     margin-top: 15px;
80 }
81
82 /* Restart Button */
83 .restart-btn {
84     background-color: ■ #007bff;
85     color: ■ #fff;
86
87     border: none;
88     padding: 10px 20px;
89     font-size: 1em;
90
91     border-radius: 8px;
92     cursor: pointer;
93
94     margin-top: 20px;
95     display: none;
96 }
97
98 .restart-btn:hover {
99     background-color: ■ #0056b3;
100 }
```

```

LabExercise > JS script.js > quizData
1  const quizData = [
2      {
3          question: "What is the basic principle of a queue?",
4          options: ["LIFO", "FIFO", "FILO", "Random"],
5          answer: "FIFO"
6      },
7      {
8          question: "Which operation removes an item from a queue?",
9          options: ["Push", "Pop", "Dequeue", "Insert"],
10         answer: "Dequeue"
11     },
12     {
13         question: "Which operation adds an item to a queue?",
14         options: ["Shift", "Enqueue", "Pop", "Remove"],
15         answer: "Enqueue"
16     },
17     {
18         question: "Where is an element inserted in a queue?",
19         options: ["Front", "Middle", "Rear", "Anywhere"],
20         answer: "Rear"
21     },
22     {
23         question: "Where is an element removed from a queue?",
24         options: ["Front", "Back", "Middle", "Bottom"],
25         answer: "Front"
26     },
27     {
28         question: "Which of the following is NOT a type of queue?",
29         options: ["Circular Queue", "Priority Queue", "Double Queue", "Simple Queue"],
30         answer: "Double Queue"
31     },
32     {
33         question: "Which data structure is best for task scheduling?",
34         options: ["Tree", "Queue", "Stack", "Graph"],
35         answer: "Queue"
36     },
37     {

```

```

38         question: "What happens when a queue is full?",
39         options: ["Underflow", "Overflow", "Crash", "Reset"],
40         answer: "Overflow"
41     }
42 ];
43
44 let currentQuestion = 0;
45 let score = 0;
46 let timeLeft = 30;
47 let timerInterval;
48 const timerEl = document.getElementById('time');
49 const questionEl = document.querySelector('.question');
50 const optionsEl = document.querySelector('.options');
51 const resultEl = document.querySelector('.result');
52 const scoreEl = document.getElementById('score');
53 const restartBtn = document.querySelector('.restart-btn');
54
55 // Function to load the question
56 function loadQuestion() {
57     if (currentQuestion >= quizData.length) {
58         endQuiz();
59         return;
60     }
61     clearInterval(timerInterval);
62     timeLeft = 30;
63     timerEl.textContent = timeLeft;
64     startTimer();
65     const currentQuiz = quizData[currentQuestion];
66     questionEl.textContent = currentQuiz.question;
67     optionsEl.innerHTML = ''; // Clear previous options
68     currentQuiz.options.forEach(option => {
69         const button = document.createElement('button');
70         button.classList.add('option');
71         button.textContent = option;

```

```

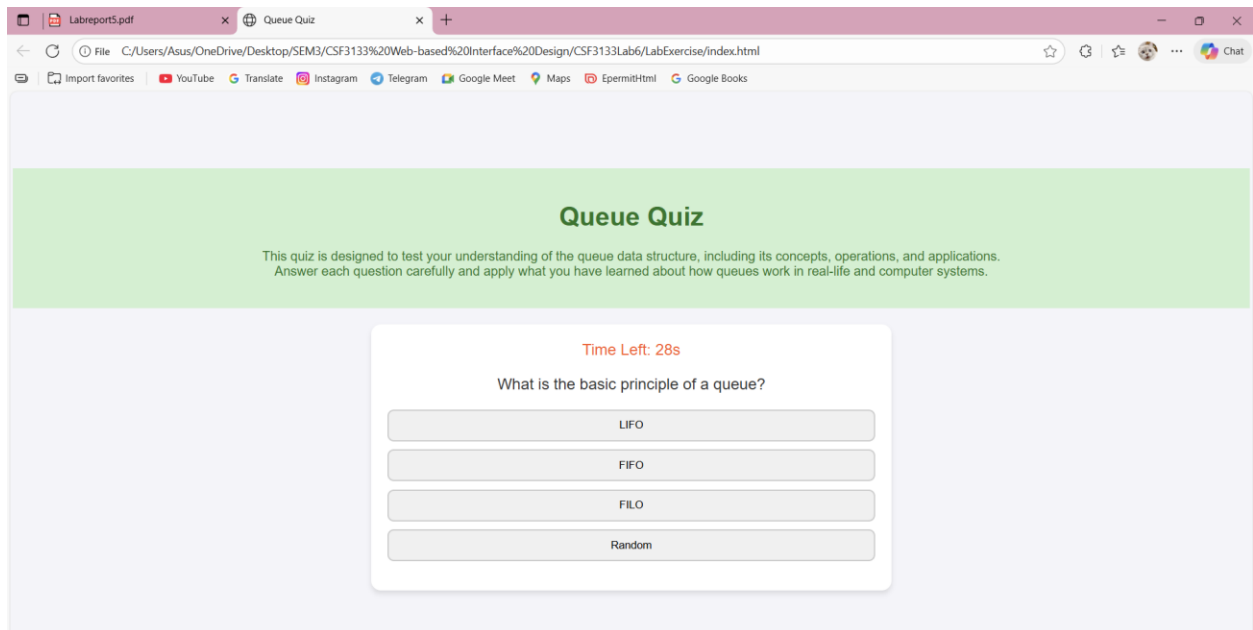
72         button.onclick = () => checkAnswer(option);
73         optionsEl.appendChild(button);
74     });
75 }
76
77 //check ans
78 function checkAnswer(selectedOption) {
79     let result = document.getElementById("result");
80     let correctAnswer = quizData[currentQuestion].answer;
81
82     if (selectedOption === correctAnswer) {
83         result.textContent = "Correct!";
84         result.style.color = "green";
85         score++;
86     } else {
87         result.textContent = "Incorrect! The correct answer is: " + correctAnswer;
88         result.style.color = "red";
89     }
90
91     currentQuestion++;
92     loadQuestion();
93 }
94
95 // Start the timer
96 function startTimer() {
97     timerInterval = setInterval(() => {
98         timeLeft--;
99         timerEl.textContent = timeLeft;
100         if (timeLeft <= 0) {
101             clearInterval(timerInterval);
102             endQuiz();
103         }
104     }, 1000);

```

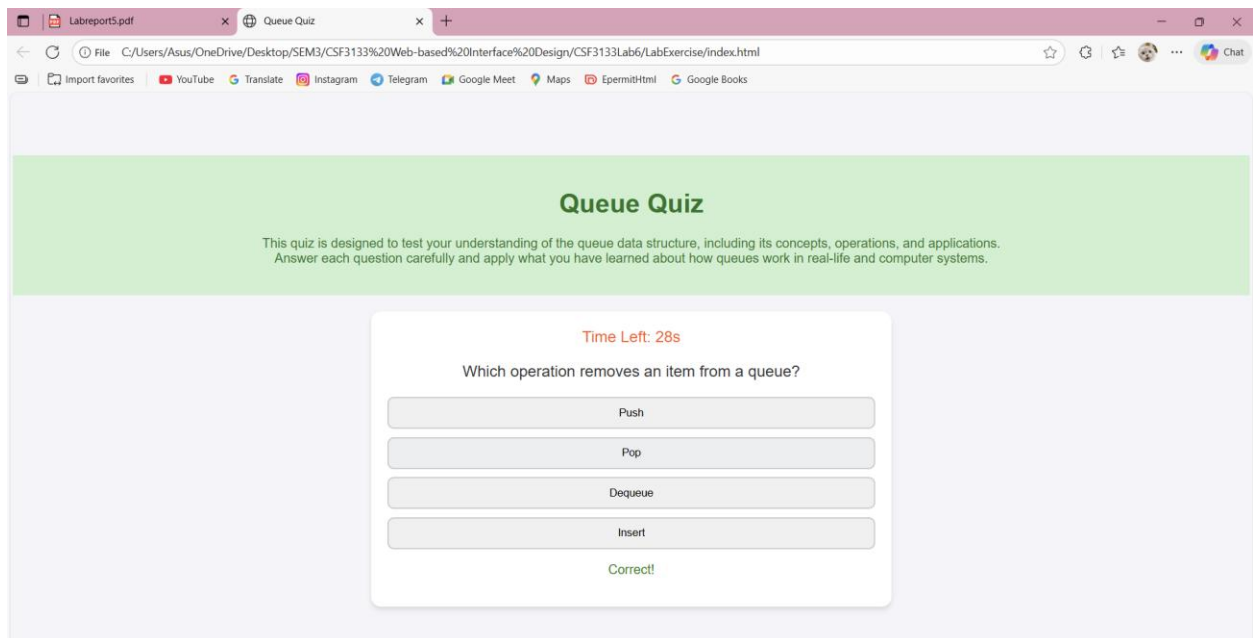
```

105 }
106
107 // End the quiz and show the results
108 function endQuiz() {
109     clearInterval(timerInterval);
110     questionEl.style.display = 'none';
111     optionsEl.style.display = 'none';
112     resultEl.style.display = 'block';
113     scoreEl.textContent = score;
114     restartBtn.style.display = 'block';
115 }
116
117 // Restart the quiz
118 restartBtn.addEventListener('click', () => {
119     // Reset variables
120     currentQuestion = 0;
121     score = 0;
122     timeLeft = 30;
123     timerEl.textContent = timeLeft;
124
125     // Reset the display
126     questionEl.style.display = 'block';
127     optionsEl.style.display = 'flex'; // Ensure options are displayed correctly
128     resultEl.style.display = 'none';
129     restartBtn.style.display = 'none';
130
131     // Load the first question
132     loadQuestion();
133 });
134
135 // Initialize the quiz with the first question
136 loadQuestion();

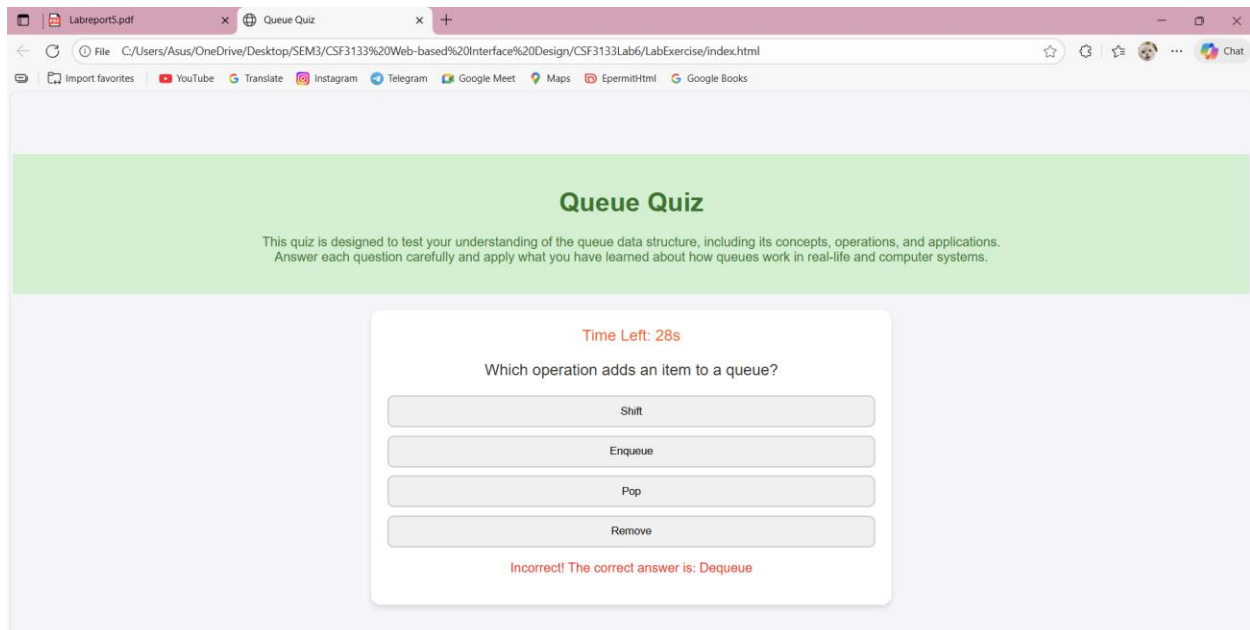
```



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.

