

**UNIVERSITI MALAYSIA TERENGGANU**

**CSM3103 – FRONT-END PROGRAMMING**

**BACHELOR OF COMPUTER SCIENCE (MOBILE COMPUTING) WITH HONORS**

**LAB 4**

**SEMESTER II 2023/2024**

**Prepared for:**

DR RABIEI BIN MAMAT

**Prepared by:**

AHMAD SHAZRUL IZZUAN BIN AHMAD FITIRI @ FITRI

(S67250)

**Link Github :**

<https://github.com/Shazrul-Izzuan/S67250_Lab-4Front.git>

# Task 1 – JavaScript Function

Code :

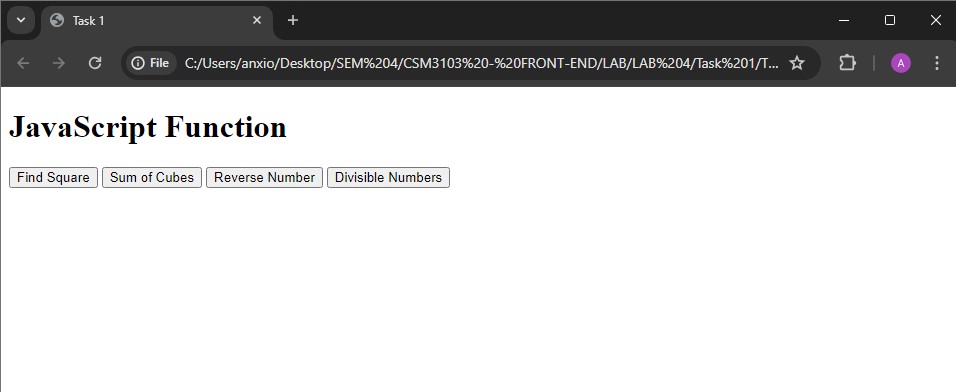
## Html

|  |
| --- |
| <!DOCTYPE html>  <html lang="en">  <head>  <meta charset="UTF-8">  <meta name="viewport" content="width=device-width, initial-scale=1.0"> <title>Task 1</title>  <script src="Task 1.js" defer></script>  </head>  <body>  <h1> JavaScript Function</h1>  <div id="output"></div>    <button onclick="findSquare()">Find Square</button>  <button onclick="sumOfCubes()">Sum of Cubes</button>  <button onclick="reverseNumber()">Reverse Number</button>  <button onclick="divisibleByZ(parseInt(prompt('Enter a number to find divisible numbers between 1 and 100:')))">Divisible Numbers</button> </body>  </html> |

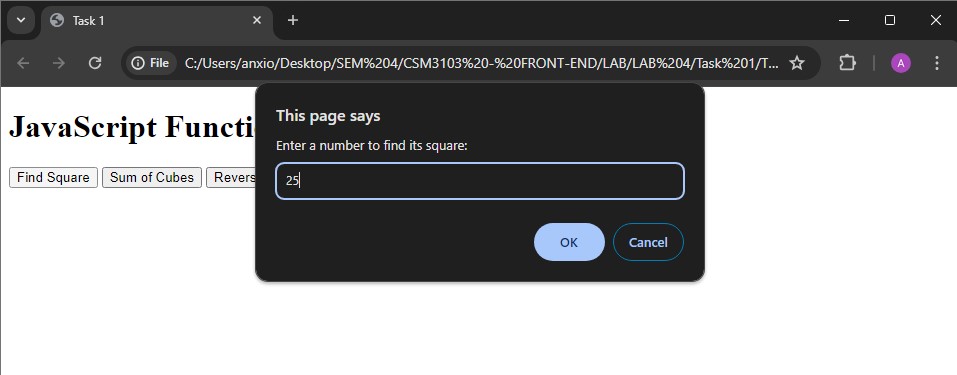
## Js

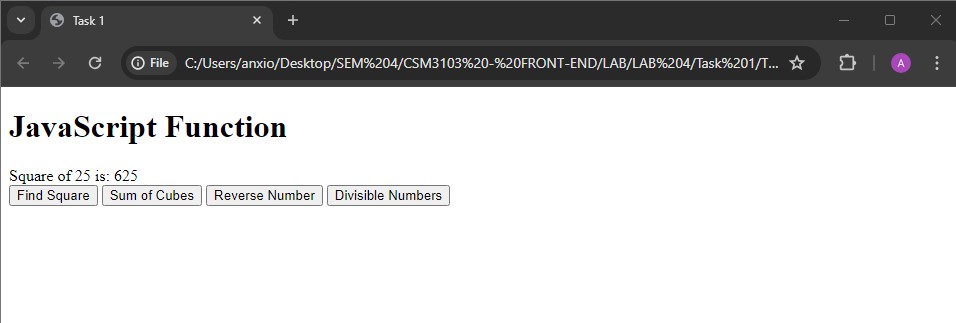
|  |
| --- |
| function findSquare() { let number = parseInt(prompt("Enter a number to find its square:")); let square = number \* number;  document.getElementById("output").innerText = `Square of ${number} is: ${square}`;  }    function sumOfCubes() {  let num1 = parseInt(prompt("Enter the first number:")); let num2 = parseInt(prompt("Enter the second number:")); let sum = Math.pow(num1, 3) + Math.pow(num2, 3); document.getElementById("output").innerText = `Sum of cubes of ${num1} and ${num2} is: ${sum}`;  }    function reverseNumber() {  let number = parseInt(prompt("Enter a number to reverse:")); let reversed = 0; while (number > 0) {  reversed = (reversed \* 10) + (number % 10); number = Math.floor(number / 10);  } |
| document.getElementById("output").innerText = `Reversed number is: ${reversed}`;  }    function divisibleByZ(z) {  let output = ""; for (let i = 1; i <= 100; i++) { if (i % z === 0) {  output += i + ", "; }  }  document.getElementById("output").innerText = `Numbers between 1 and 100 divisible by ${z} are: ${output}`;  } |

Output :

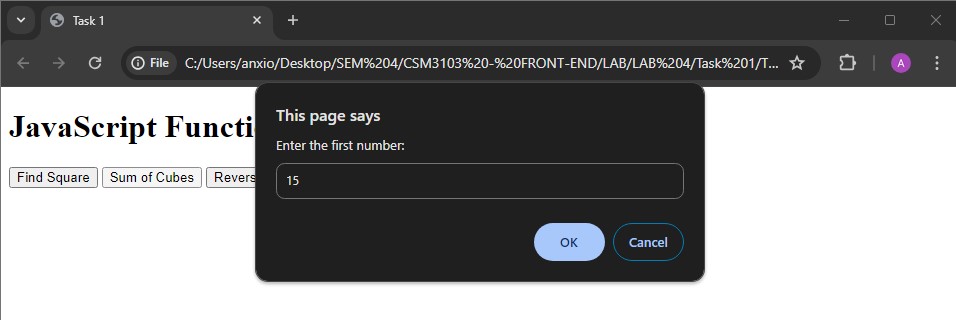


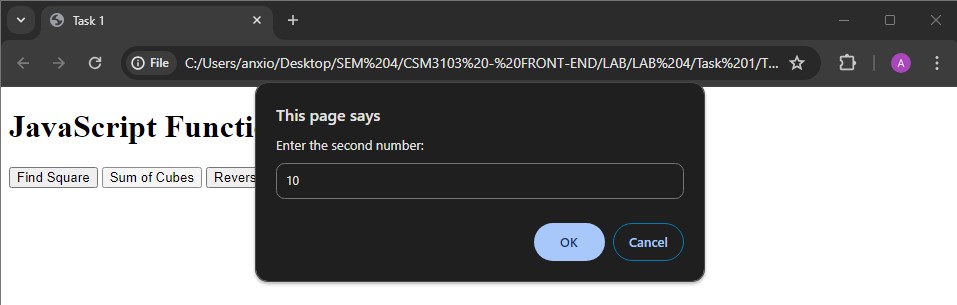
## Find Square





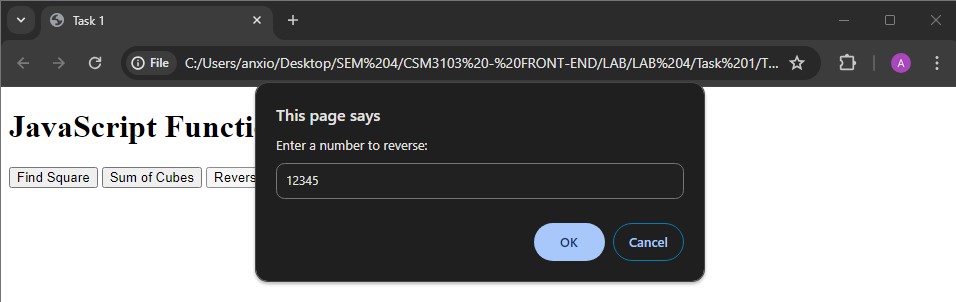
## Sum of Cubes



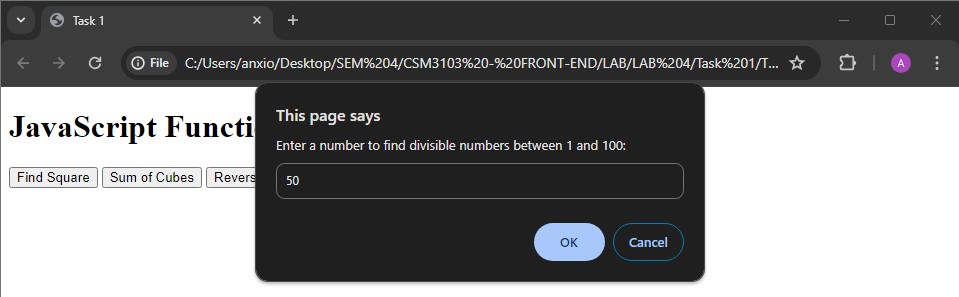


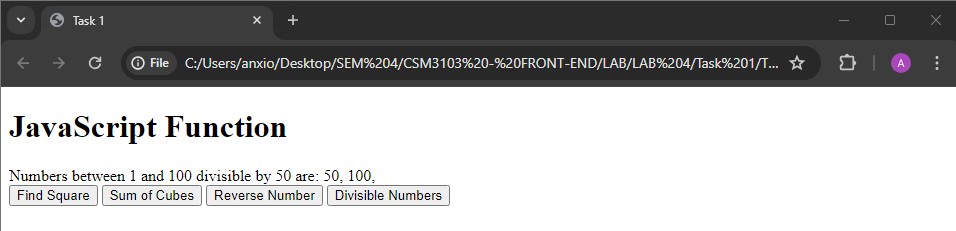


## Reverse Number



## • Divisible Numbers





# Task 2 - JavaScript Recursion Function

Code :

## Html

|  |
| --- |
| <!DOCTYPE html>  <html lang="en">  <head>  <meta charset="UTF-8">  <meta name="viewport" content="width=device-width, initial-scale=1.0"> <title>Task 2</title>  <script src="Task 2.js" defer></script>  </head>  <body>  <h1>Recursion Functions</h1> <div id="output"></div>    <button onclick="sumOfDigits()">Sum of Digits</button>  <button onclick="power(parseInt(prompt('Enter base:')), parseInt(prompt('Enter exponent:')))">Power</button>  </body>  </html> |

## Js

|  |
| --- |
| function sumOfDigits() {  let number = parseInt(prompt("Enter a number to find sum of its digits:")); let sum = calculateSumOfDigits(number);  document.getElementById("output").innerText = `Sum of digits of ${number} is: ${sum}`;  }    function calculateSumOfDigits(number) {  if (number === 0) {  return 0;  } else {  return (number % 10) + calculateSumOfDigits(Math.floor(number / 10));  }  }    function power(x, y) {  let result = calculatePower(x, y); document.getElementById("output").innerText = `${x} raised to the power ${y} is: ${result}`;  }    function calculatePower(x, y) {  if (y === 0) { return 1; |

} else if (y > 0) {

return x \* calculatePower(x, y - 1);

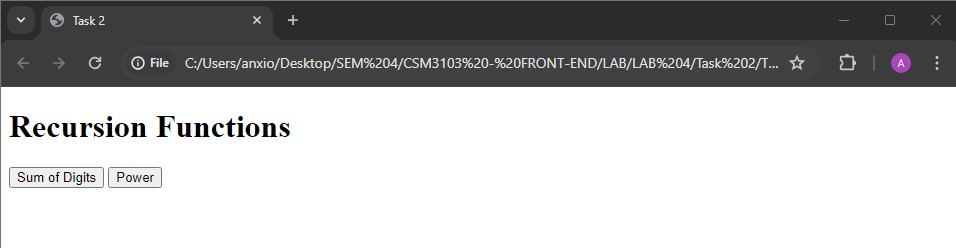
} else {

return 1 / calculatePower(x, -y);

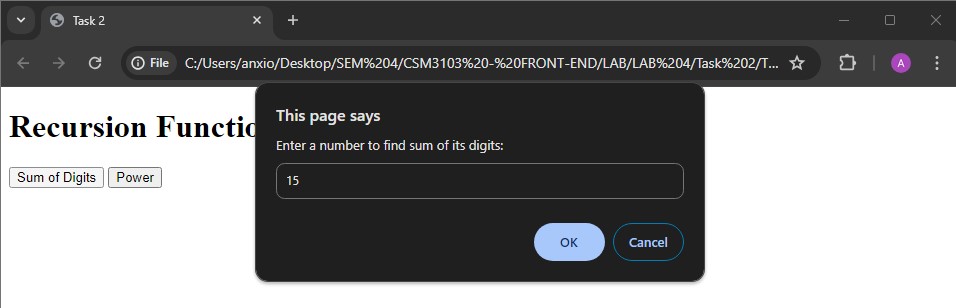
}

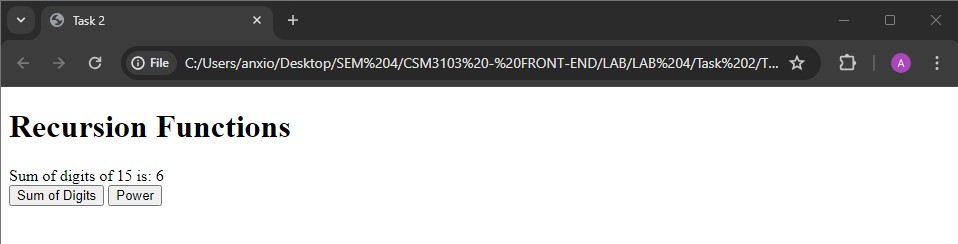
}

Output :

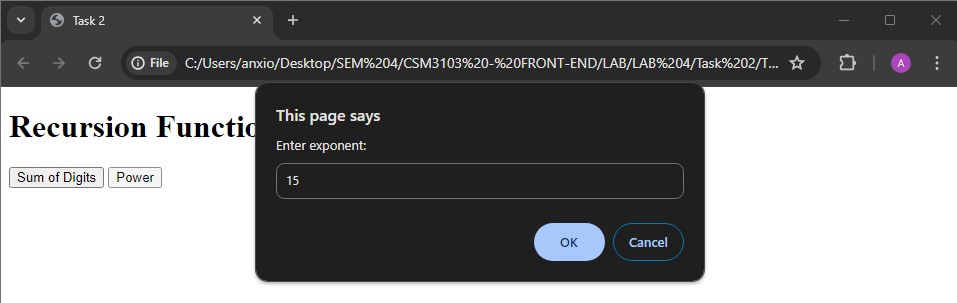
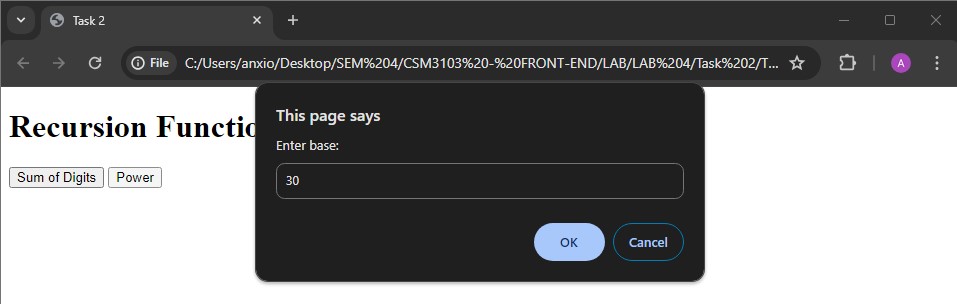


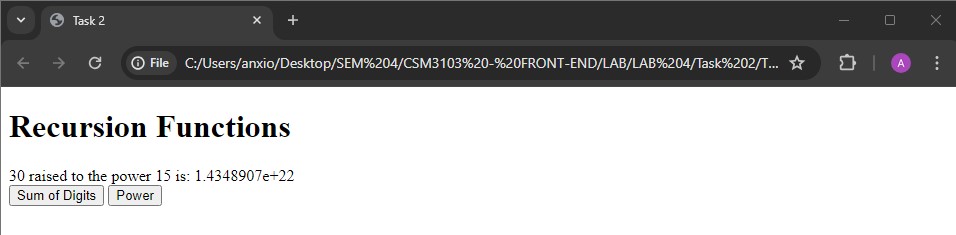
## • Sum of digits





## • Power





# Task 3 – JavaScript Object and Prototype

Code :

## Html

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

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

<title>Task 3</title>

</head>

<body>

<h2>Product Details</h2>

<form id="productForm">

<label for="productName">Product Name:</label>

<input type="text" id="productName" required><br><br>

<label for="quantity">Quantity:</label>

<input type="number" id="quantity" required><br><br>

<label for="price">Price:</label>

<input type="number" id="price" required><br><br>

<button type="button" onclick="addProduct()">Add Product</button>

</form>

<h2>Book Details</h2>

<form id="bookForm">

<label for="bookName">Book Name:</label>

<input type="text" id="bookName" required><br><br>

<label for="authorName">Author Name:</label>

<input type="text" id="authorName" required><br><br>

<label for="bookPrice">Price:</label>

<input type="number" id="bookPrice" required><br><br>

<button type="button" onclick="addBook()">Add Book</button>

</form>

<h2>Output</h2>

<div id="output"></div>

<script src="Task 3.js"></script>

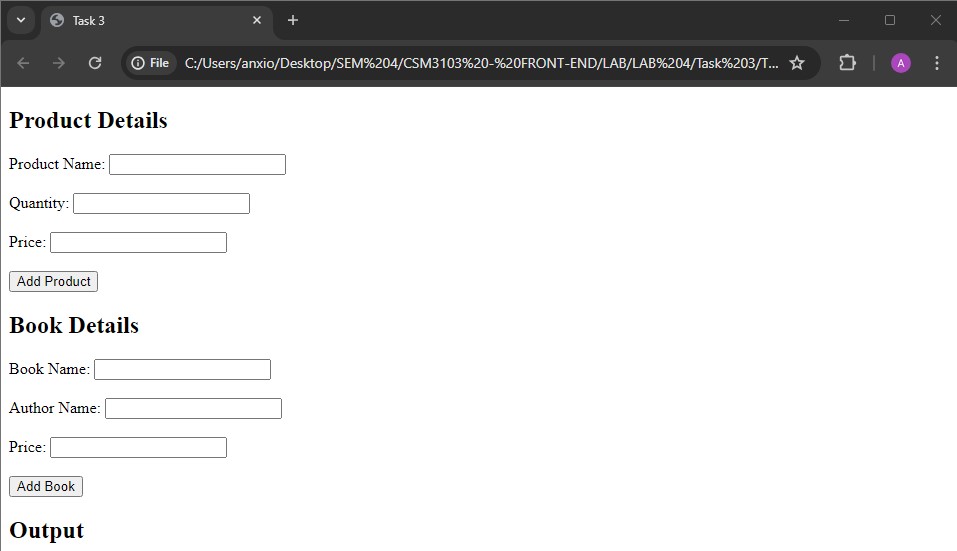
</body>

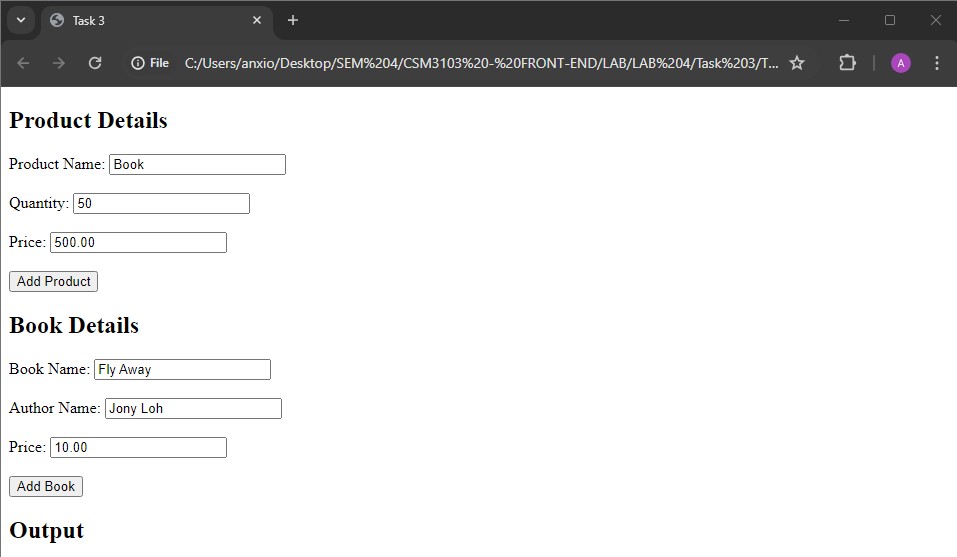
</html>

## Js

|  |
| --- |
| function Product(name, quantity, price) { this.name = name; this.quantity = quantity; this.price = price;  }    function addProduct() {  const productName = document.getElementById('productName').value; const quantity = parseInt(document.getElementById('quantity').value); const price = parseFloat(document.getElementById('price').value);    const product = new Product(productName, quantity, price);    displayOutput(product);  }      function Book(bookName, authorName) { this.bookName = bookName;  this.authorName = authorName;  }    Book.prototype.price = null;    function addBook() {  const bookName = document.getElementById('bookName').value; const authorName = document.getElementById('authorName').value; const bookPrice = parseFloat(document.getElementById('bookPrice').value);    const book = new Book(bookName, authorName); book.price = bookPrice;    displayOutput(book);  }    function displayOutput(obj) {  let outputDiv = document.getElementById('output'); let outputHTML = '';    for (let prop in obj) {  if (obj.hasOwnProperty(prop)) {  outputHTML += `<strong>${prop}:</strong> ${obj[prop]}<br>`;  }  }    outputDiv.innerHTML += outputHTML + '<br>';  } |

Output :







# Task 4 – Event Manager

Code :

## Html

|  |
| --- |
| <!DOCTYPE html>  <html lang="en">  <head>  <meta charset="UTF-8">  <meta name="viewport" content="width=device-width, initial-scale=1.0">  <title>Task 4</title>  <style>  #paragraph { padding: 20px; border: 1px solid black; cursor: pointer;  }    #textfield { padding: 10px; font-size: 16px;  transition: all 0.3s ease;  }  </style>  </head>  <body>  <h1>Number 1 - Change the paragraph color</h1> <p id="paragraph">Click me!</p>    <h1>Number 2 - Text Field Events</h1>  <input type="text" id="textfield" placeholder="Type something...">    <script src="eventmanager.js"></script> <script src="textfield.js"></script>  </body>  </html> |

## Js (eventmanager)

const paragraph = document.getElementById('paragraph');

paragraph.onclick = function() {

paragraph.style.backgroundColor = 'yellow';

};

paragraph.ondblclick = function() {

paragraph.style.backgroundColor = 'blue';

};

paragraph.onmouseover = function() {

paragraph.style.backgroundColor = 'red';

};

paragraph.onmouseout = function() { paragraph.style.backgroundColor = 'green';

};

## Js (textfield)

const textfield = document.getElementById('textfield');

textfield.onchange = function() {

textfield.style.border = '2px solid blue';

};

textfield.onfocus = function() { textfield.style.backgroundColor = '#f0f0f0';

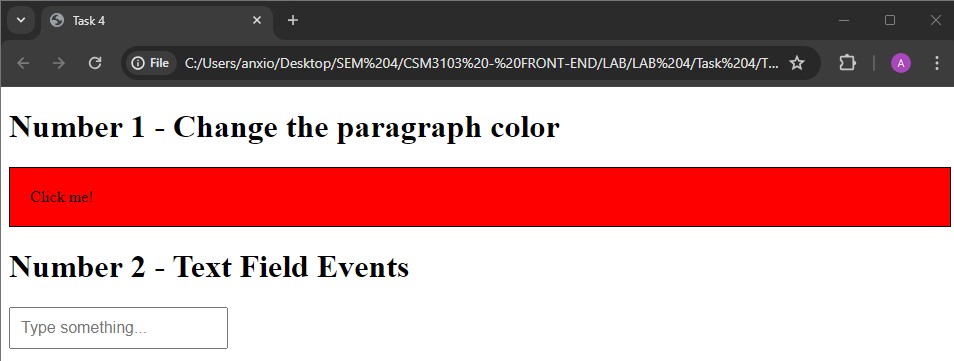
};

textfield.onblur = function() {

textfield.style.backgroundColor = 'white';

};

Output :



# Task 5

Code :

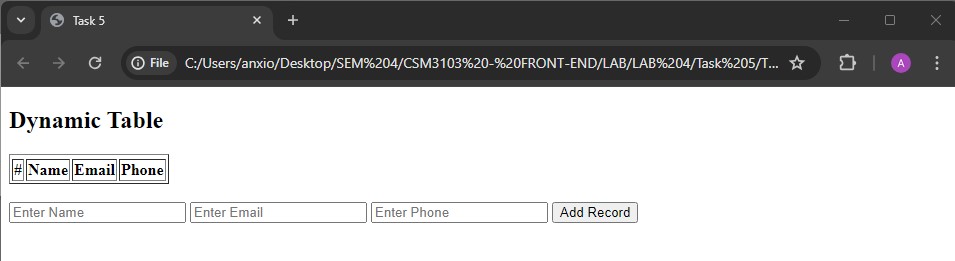
## Html

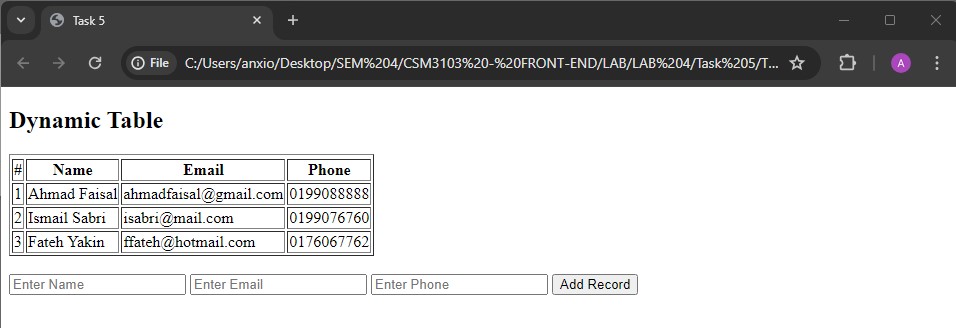
|  |
| --- |
| <!DOCTYPE html>  <html lang="en">  <head>  <meta charset="UTF-8">  <meta name="viewport" content="width=device-width, initial-scale=1.0"> <title>Task 5</title>  </head>  <body>    <h2>Dynamic Table</h2>    <table id="myTable" border="1">  <thead>  <tr>  <th>#</th>  <th>Name</th>  <th>Email</th>  <th>Phone</th>  </tr>  </thead>  <tbody>  </tbody>  </table>    <br>    <input type="text" id="name" placeholder="Enter Name">  <input type="text" id="email" placeholder="Enter Email">  <input type="text" id="phone" placeholder="Enter Phone"> <button onclick="addRow()">Add Record</button>    <script src="Task 5.js"></script>    </body>  </html> |

## Js

|  |
| --- |
| function addRow() {  var table = document.getElementById("myTable").getElementsByTagName('tbody')[0]; var newRow = table.insertRow(table.rows.length); var cells = []; for (var i = 0; i < 4; i++) {  cells.push(newRow.insertCell(i));  }  cells[0].innerHTML = table.rows.length;  cells[1].innerHTML = document.getElementById("name").value; cells[2].innerHTML = document.getElementById("email").value; cells[3].innerHTML = document.getElementById("phone").value;    document.getElementById("name").value = ""; document.getElementById("email").value = ""; document.getElementById("phone").value = "";  }    window.onload = function() {  var table = document.getElementById("myTable"); var header = table.createTHead(); var row = header.insertRow(0);  var headerCells = [];  for (var i = 0; i < headerCells.length; i++) { var cell = row.insertCell(i); cell.innerHTML = headerCells[i];  }  }    document.addEventListener('DOMContentLoaded', function() { var table = document.getElementById("myTable"); table.onclick = function(e) {  if (e.target.tagName.toLowerCase() === 'td') { var index = e.target.parentNode.rowIndex; table.deleteRow(index);  }  };  }); |

Output :





# Task 6

Code :

## Html

|  |
| --- |
| <!DOCTYPE html>  <html lang="en">  <head>  <meta charset="UTF-8">  <meta name="viewport" content="width=device-width, initial-scale=1.0"> <title>Task 6</title>  <style>  #container { position: relative; width: 400px; height: 400px;  border: 2px solid black;  }  .small-square { position: absolute; width: 20px; height: 20px;  background-color: red;  }  </style>  </head>  <body>  <div id="container">  <div id="square1" class="small-square"></div> <div id="square2" class="small-square"></div> </div>  <button id="startBtn">Start Animation</button>  <button id="stopBtn">Stop Animation</button>  <script src="Task 6.js"></script>  </body>  </html> |

## Js

|  |
| --- |
| let intervalId;    function moveSquares() {  const container = document.getElementById('container'); const square1 = document.getElementById('square1'); const square2 = document.getElementById('square2');  const containerWidth = container.clientWidth; const containerHeight = container.clientHeight; const squareWidth = square1.clientWidth; |
| const squareHeight = square1.clientHeight;    intervalId = setInterval(() => {  const randomX1 = Math.floor(Math.random() \* (containerWidth - squareWidth)); const randomY1 = Math.floor(Math.random() \* (containerHeight - squareHeight)); const randomX2 = Math.floor(Math.random() \* (containerWidth - squareWidth)); const randomY2 = Math.floor(Math.random() \* (containerHeight - squareHeight));    square1.style.left = randomX1 + 'px'; square1.style.top = randomY1 + 'px'; square2.style.left = randomX2 + 'px'; square2.style.top = randomY2 + 'px';  }, 1000);  }    function stopAnimation() {  clearInterval(intervalId);  }    document.getElementById('startBtn').addEventListener('click', moveSquares); document.getElementById('stopBtn').addEventListener('click', stopAnimation); |

Output :

