Ex. No. 5	JavaScript – DOM Elements, Events and Functions
Date of Exercise	13/08/2025
Git-hub Host Link	https://loetkarthik.github.io/web/exp5._1 https://loetkarthik.github.io/web/exp5

Aim

To design and develop interactive web pages using JavaScript for:

Performing basic arithmetic operations (addition, subtraction, multiplication, and division) on two given numbers.

Calculating the Equated Monthly Installment (EMI) for a bank loan based on the loan amount, annual interest rate, and tenure.

Description

This project includes two JavaScript-based web applications integrated with HTML and CSS for structure and styling. The first is a Simple Calculator that takes two numbers as input and performs addition, subtraction, multiplication, and division, displaying the result instantly upon button clicks. The second is a Bank Loan EMI Calculator that calculates the monthly EMI, total interest, and total repayment for a given loan amount, annual interest rate, and tenure using the standard EMI formula P×R×(1+R)N(1+R)N−1\frac{P \times R \times (1 + R)^N}{(1 + R)^N - 1}(1+R)N−1P×R×(1+R)N , where PPP is the principal, RRR is the monthly interest rate, and NNN is the tenure in months. Both applications are designed to be interactive, user-friendly, and visually appealing through the use of responsive layout and clean styling.

Procedure:

1. **Start the HTML Document**
    - Begin with the <!DOCTYPE html> declaration.
    - Open the <html> tag and set the language attribute.
2. **Create the Head Section**
    - Inside the <head> tag, include:
        - A meta tag for character set.
        - A meta tag for viewport settings.
        - A title for the web page.

3. **Add Page Heading and Sections**
    - Use <h1> for the main page heading.
    - Use <h2> for subheadings in each section.
    - Use <p> for short text under each subheading.
4. **Include Various Hyperlinks**
    - Add simple examples of these:
        - External link
        - Internal link
        - Bookmark link
        - Email link
        - Phone link
        - Clickable image link
5. **Add Lists**
    - Insert a **definition list** using <dl>, <dt>, and <dd>.
    - Insert an **ordered list** with a nested ordered list inside it.
    - Insert an **unordered list** with a nested unordered list inside it.
6. **Insert a Table**
    - Use <table> to create rows and columns.
    - Include a heading row and multiple data rows.
7. **Add an Image**
    - Use the <img> tag to add an image with an alternative text.
8. **Insert HTML Comments**
    - Use <!-----> to write comments in the code.
9. **Close All Tags Properly**

- Ensure that each HTML tag is correctly opened and closed.

Program

1.Question:

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

  <head>
    <meta charset="UTF-8">

    <title>
      Document
    </title>

    <link rel="stylesheet" href="exp5.css">
    <script src="exp5.js"></script>

  </head>

  <body class="body">

    <form class="form" action="/submit-form.pvp" method="post">

        <h1>
          Simple calculator

        </h1>

        <input class="num js-num1"  placeholder="Enter the Number" type="text" name="num"
required>



        <br><br>

        <input class="num js-num2"  placeholder="Enter the Number" type="text" name="num"
required>
```

```
        <br><br>

        <input class="sub js-sub" type="button" value="Add" onclick="add()">
        <input class="sub js-sub" type="button" value="Subtract">

        <br><br>

        <input class="sub js-sub" type="button" value="Multiply">
        <input class="sub js-sub" type="button" value="Divide" >

        <p class="js-result">
          Result :
        </p>

      </form>

  </body>

</html>
```

JS_FILE:

```
const var1 = document.querySelector('.js-num1');
const var2 = document.querySelector('.js-num2');

const num1 = Number(var1.value);
const num2 = Number(var2.value);

function add() {



  const result = num1 + num2;
  console.log("Result:", result);

}
```

5. Question

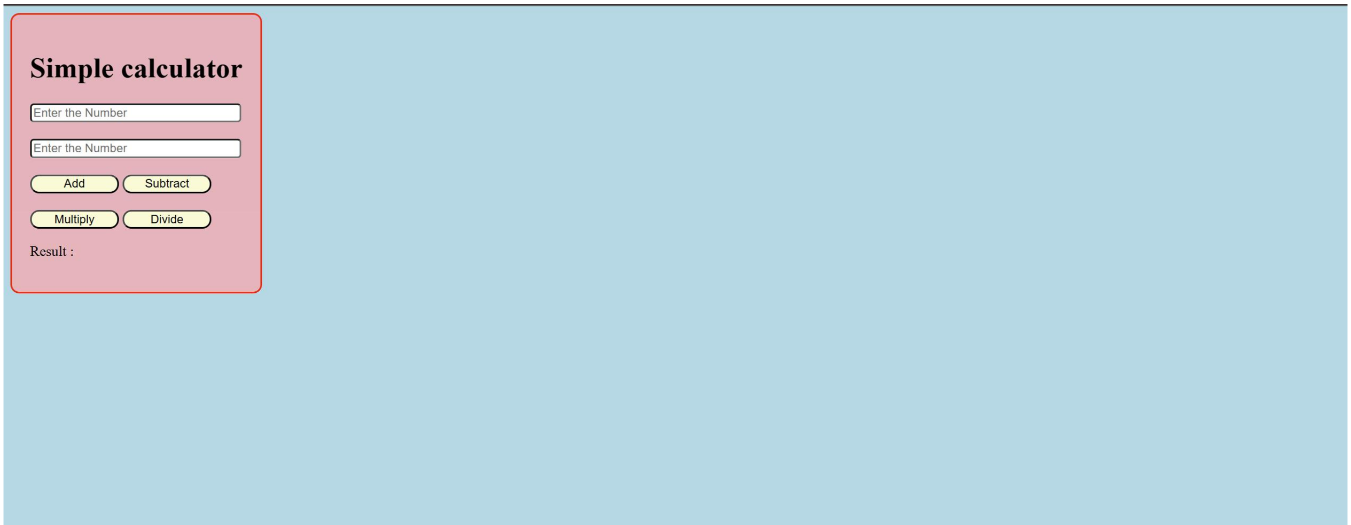
```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Loan EMI Calculator</title>
  <style>
    body {
      font-family: Arial, sans-serif;
      background-color: #f2f2f7;
      display: flex;
      justify-content: center;
      align-items: center;
      height: 100vh;
    }
    .container {
      background: white;
      padding: 20px 30px;
      border-radius: 10px;
      box-shadow: 0 4px 10px rgba(0, 0, 0, 0.1);
      width: 350px;
    }
    h2 {
      text-align: center;
    }
    label {
      font-weight: bold;
      margin-top: 10px;
      display: block;
    }
```

```
        input {
            width: 100%;
            padding: 8px;
            margin-top: 5px;
            border: 1px solid #ccc;
            border-radius: 5px;
        }
        button {
            width: 100%;
            margin-top: 15px;
            padding: 10px;
            background-color: #4CAF50;
            border: none;
            color: white;
            font-size: 16px;
            border-radius: 5px;
            cursor: pointer;
        }
        button:hover {
            background-color: #45a049;
        }
        .result {
            margin-top: 15px;
            padding: 10px;
            border-radius: 5px;
            background: #f9f9f9;
            font-size: 14px;
        }
        .emi {
            color: blue;
            font-weight: bold;
        }
    </style>
</head>
```

```
<body>
  <div class="container">
    <h2>Loan EMI Calculator</h2>
    <label for="amount">Loan Amount (₹):</label>
    <input type="number" id="amount" placeholder="Enter loan amount">

    <label for="rate">Annual Interest Rate (%):</label>
    <input type="number" step="0.01" id="rate" placeholder="Enter interest rate">

    <label for="years">Tenure (Years):</label>
    <input type="number" id="years" placeholder="Enter years">

    <button onclick="calculateEMI()">Calculate EMI</button>

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

  <script>
    function calculateEMI() {
      let P = parseFloat(document.getElementById('amount').value);
      let annualRate = parseFloat(document.getElementById('rate').value);
      let years = parseInt(document.getElementById('years').value);

      if (isNaN(P) || isNaN(annualRate) || isNaN(years) || P <= 0 || annualRate <= 0 || years <= 0) {
        document.getElementById('output').innerHTML = "Please enter valid values.";
        return;
      }

      let R = annualRate / 12 / 100; // Monthly interest rate
      let N = years * 12; // Months

      let EMI = (P * R * Math.pow(1 + R, N)) / (Math.pow(1 + R, N) - 1);
      let totalPayment = EMI * N;
      let totalInterest = totalPayment - P;
      document.getElementById('output').innerHTML = `
        Loan Amount: ₹${P.toLocaleString()} <br>
        Total Interest: ₹${Math.round(totalInterest).toLocaleString()} <br>
        EMI: <span class="emi">₹${Math.round(EMI).toLocaleString()}</span> <br>
        Total Repayment: ₹${Math.round(totalPayment).toLocaleString()}
      `;
    }
  </script>
</body>
</html>
```

Output:

1. Simple Calculator

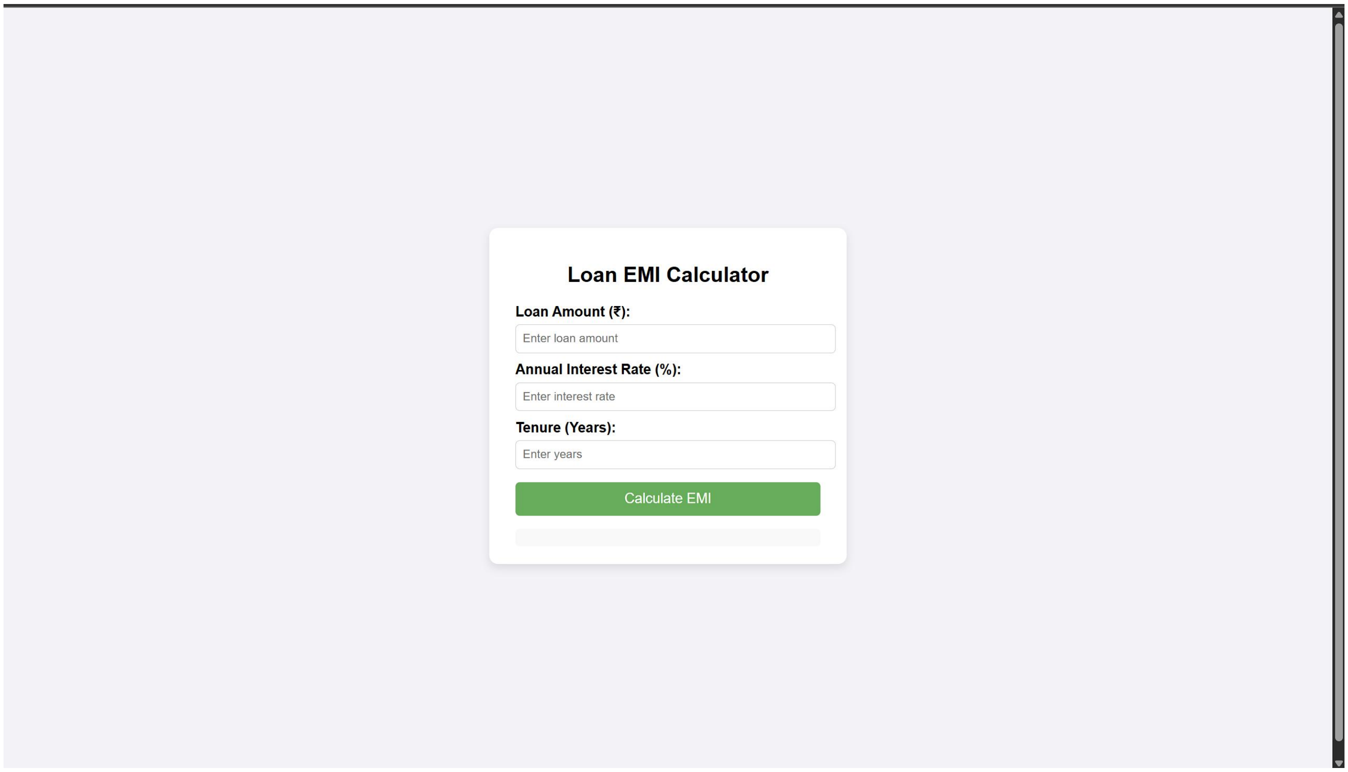


Simple calculator

Result :

Output

2. Loan EMI Calculator



Loan EMI Calculator

Loan Amount (₹):

Enter loan amount

Annual Interest Rate (%):

Enter interest rate

Tenure (Years):

Enter years

Result

Two interactive web applications were successfully developed using HTML, CSS, and JavaScript. The first, a Simple Calculator, accurately performed addition, subtraction, multiplication, and division for any two user-provided numbers. The second, a Bank Loan EMI Calculator, correctly computed the EMI, total interest, and total repayment amount based on the entered loan amount, annual interest rate, and tenure, using the given formula. Both applications functioned as expected, providing instant and accurate results with a clean and user-friendly interface.