|  |  |
| --- | --- |
| **Ex. No. 6** | **Dynamic Website Design with JQuery Actions** |
| **Date of Exercise** | 03.09.2025 |
| **Git-hub Host Link** | [**https://aadinine.github.io/ex66/**](https://aadinine.github.io/ex66/) |

**Aim**

To create dynamic web applications using jQuery for DOM manipulation, event handling, and interactive functionality implementation.

**Description**

This experiment covers:

1. **jQuery Basics**
   * Including jQuery library in HTML
   * jQuery syntax and selectors
   * Document ready function
2. **jQuery Event Handling**
   * click() event for button interactions
   * keyup() event for real-time calculations
   * change() event for dropdown selections
3. **jQuery DOM Manipulation**
   * Retrieving and updating values with val()
   * Modifying content with html()
   * Dynamic content creation
4. **Application Development**
   * Budget calculator with percentage-based allocation
   * Employee payroll calculator with tax calculations
   * GST calculator with dynamic updates

**Procedure:**

**Question 1: Arithmetic Calculator**

**Step 1:** Open VS Code and create new files  
**Step 2:** Set up HTML5 structure in ex5.html with:

* Two number input fields
* Four operation buttons (Add/Subtract/Multiply/Divide)
* Result display div

**Step 3:** Style in ex5.css with:

* Calculator container with box-shadow
* Responsive button grid
* Hover effects for buttons

**Step 4:** Add JavaScript in ex5.js:

* Create calculate() function with switch-case for operations
* Handle division-by-zero error
* Update results using textContent

**Step 5:** Link all files and test in browser  
**Step 6:** Upload to GitHub and enable Pages

**Question 2: Real-Time BMI Calculator**

**Step 1:** Create new files in VS Code  
**Step 2:** Build HTML structure in ex6.html:

* Weight (kg) and height (cm) inputs with oninput
* BMI value and status display divs

**Step 3:** Style in ex6.css:

* Clean minimalist design
* Color-coded status indicators
* Responsive input fields

**Step 4:** Implement JavaScript in ex6.js:

* Automatic cm-to-meter conversion
* Real-time BMI calculation on input change
* Color-coded classification (Underweight → Obese)

**Step 5:** Test responsiveness in browser  
**Step 6:** Deploy via GitHub Pages

**Program 1:**

Html:

Css:

Js:

**Program 2:**

Html:

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

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

    <title>BMI Calculator - URK23CS1010</title>

    <link rel="stylesheet" href="ex6.css">

</head>

<body>

    <div class="bmi-calculator">

        <h1>Body Mass Index (BMI)</h1>

        <p class="subtitle">A measure of body fat in adults</p>

        <div class="input-group">

            <label for="weight">Weight (kg):</label>

            <input type="number" id="weight" placeholder="Enter weight" oninput="calculateBMI()">

        </div>

        <div class="input-group">

            <label for="height">Height (cm):</label>

            <input type="number" id="height" placeholder="Enter height" oninput="calculateBMI()">

        </div>

        <div class="result-container">

            <div id="bmi-value">-</div>

            <div id="bmi-status">Enter your details</div>

        </div>

    </div>

    <script src="ex6.js"></script>

</body>

</html>

Css:

body {

    font-family: Arial, sans-serif;

    display: flex;

    justify-content: center;

    align-items: center;

    height: 100vh;

    margin: 0;

    background-color: #f5f5f5;

}

.calculator {

    background-color: white;

    padding: 20px;

    border-radius: 10px;

    box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);

    text-align: center;

    width: 300px;

}

h1 {

    color: #333;

    margin-bottom: 20px;

}

.inputs {

    margin-bottom: 15px;

}

input {

    width: 100%;

    padding: 8px;

    margin: 5px 0;

    box-sizing: border-box;

    border: 1px solid #ddd;

    border-radius: 4px;

}

.buttons {

    display: grid;

    grid-template-columns: repeat(2, 1fr);

    gap: 10px;

    margin-bottom: 15px;

}

button {

    padding: 10px;

    background-color: #4CAF50;

    color: white;

    border: none;

    border-radius: 4px;

    cursor: pointer;

    transition: background-color 0.3s;

}

button:hover {

    background-color: #45a049;

}

.result {

    font-size: 18px;

    font-weight: bold;

    padding: 10px;

    background-color: #f0f0f0;

    border-radius: 4px;

}

Js:

function calculateBMI() {

    const weight = parseFloat(document.getElementById('weight').value);

    const height = parseFloat(document.getElementById('height').value) / 100; // Convert cm to m

    const bmiValue = document.getElementById('bmi-value');

    const bmiStatus = document.getElementById('bmi-status');

    if (weight > 0 && height > 0) {

        const bmi = weight / (height \* height);

        const roundedBMI = bmi.toFixed(1);

        bmiValue.textContent = roundedBMI;

        // Determine status with color coding

        if (bmi < 18.5) {

            updateStatus("Underweight", "#3498db");

        } else if (bmi < 25) {

            updateStatus("Normal", "#2ecc71");

        } else if (bmi < 30) {

            updateStatus("Overweight", "#f39c12");

        } else {

            updateStatus("Obese", "#e74c3c");

        }

    } else {

        bmiValue.textContent = "-";

        bmiStatus.textContent = "Enter valid values";

        bmiStatus.style.color = "#7f8c8d";

    }

}

function updateStatus(status, color) {

    const bmiStatus = document.getElementById('bmi-status');

    bmiStatus.textContent = status;

    bmiStatus.style.color = color;

    document.getElementById('bmi-value').style.color = color;

}

**Output**

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

**Result**

Successfully implemented:

1. A **real-time BMI calculator** that updates dynamically and classifies results visually
2. An **interactive arithmetic calculator** with four operations
3. Demonstrated DOM manipulation and event handling as specified in the objectives