



**VIT<sup>®</sup>**

**Vellore Institute of Technology**  
(Deemed to be University under section 3 of UGC Act, 1956)  
CHENNAI

# **WEB PROGRAMMING LAB**

**BCSE203E**

**SLOT: L11 + L12 + L19 + L20**

## **LAB MANUAL**

**Exercise 5**

**Date: 23-02-2025**

### **GITHUB REPO LINK:**

[https://github.com/divyanshupatel17/23BAI1214\\_WEB\\_LAB](https://github.com/divyanshupatel17/23BAI1214_WEB_LAB)

### **WEB PROG LAB PORTFOLIO:**

<https://23bai1214-divyanshu-web-lab-portfolio.netlify.app/>

**SUBMITTED TO:**

**PROF. DR. ASHOKA RAJAN R**

**SUBMITTED BY:**

**DIVYANSHU PATEL**

**23BAI1214**

**School of Computer Science and Engineering**

Vellore Institute of Technology, Chennai

Chennai 600127, Tamil Nadu

# 1. Traffic Light

<https://23bai1214-divyanshu-exercise-5-t-1.netlify.app/>

```
<!-- 1. Traffic Light -->

<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Modern Traffic Light</title>
  <style>
    :root {
      --red: rgb(207, 26, 26);
      --yellow: rgb(240, 225, 13);
      --green: rgb(8, 113, 8);
      --button-bg: #2a2a2a;
      --button-text: #ffffff;
    }

    * {
      margin: 0;
      padding: 0;
      box-sizing: border-box;
    }

    body {
      min-height: 100vh;
      display: grid;
      place-items: center;
      gap: 2rem;
      background: linear-gradient(135deg, #1a1a1a 0%, #303030 100%);
      font-family: system-ui, -apple-system, sans-serif;
      padding: 2rem;
    }

    .traffic-light {
      background: linear-gradient(145deg, #333 0%, #222 100%);
      padding: 1.5rem;
      border-radius: 4rem;
      display: flex;
      flex-direction: column;
      gap: 1.5rem;
      box-shadow:
        0 10px 20px rgba(0, 0, 0, 0.2),
        inset 0 2px 4px rgba(255, 255, 255, 0.1);
    }

    .signal {
      width: 5rem;
      height: 5rem;
    }
  </style>
</head>
<body>
  <div class="traffic-light">
    <div class="signal red"></div>
    <div class="signal yellow"></div>
    <div class="signal green"></div>
  </div>
</body>
</html>
```

```
border-radius: 50%;  
position: relative;  
transition: all 0.3s ease;  
}  
  
.signal::before {  
content: '';  
position: absolute;  
top: 0;  
left: 0;  
right: 0;  
bottom: 0;  
border-radius: 50%;  
box-shadow: inset 0 2px 4px rgba(0, 0, 0, 0.3);  
}  
  
.red {  
background-color: var(--red);  
opacity: 0.3;  
}  
  
.yellow {  
background-color: var(--yellow);  
opacity: 0.3;  
}  
  
.green {  
background-color: var(--green);  
opacity: 0.3;  
}  
  
.controls {  
display: flex;  
flex-direction: column;  
gap: 1rem;  
align-items: center;  
}  
  
.button-group {  
display: flex;  
gap: 0.75rem;  
}  
  
.button {  
background-color: var(--button-bg);  
color: var(--button-text);  
border: none;  
padding: 0.75rem 1.5rem;  
border-radius: 0.75rem;  
cursor: pointer;  
font-weight: 500;  
transition: transform 0.2s ease, box-shadow 0.2s ease;  
text-transform: uppercase;  
letter-spacing: 0.5px;
```

```
}

.button:hover {
    transform: translateY(-2px);
    box-shadow: 0 4px 12px rgba(0, 0, 0, 0.2);
}

.button:active {
    transform: translateY(0);
}

.active {
    opacity: 1 !important;
    box-shadow: 0 0 50px currentColor !important;
}

</style>
</head>
<body>
    <div class="traffic-light">
        <div class="signal" id="red"></div>
        <div class="signal" id="yellow"></div>
        <div class="signal" id="green"></div>
    </div>

    <div class="controls">
        <button class="button" id="random">Random Signal</button>
        <div class="button-group">
            <button class="button" data-color="red">Red</button>
            <button class="button" data-color="yellow">Yellow</button>
            <button class="button" data-color="green">Green</button>
        </div>
    </div>

<script>
    const signals = {
        red: document.getElementById('red'),
        yellow: document.getElementById('yellow'),
        green: document.getElementById('green')
    };

    function resetSignals() {
        Object.values(signals).forEach(signal => {
            signal.style.opacity = '0.3';
            signal.classList.remove('active');
        });
    }

    function activateSignal(color) {
        resetSignals();
        signals[color].classList.add('active');
    }

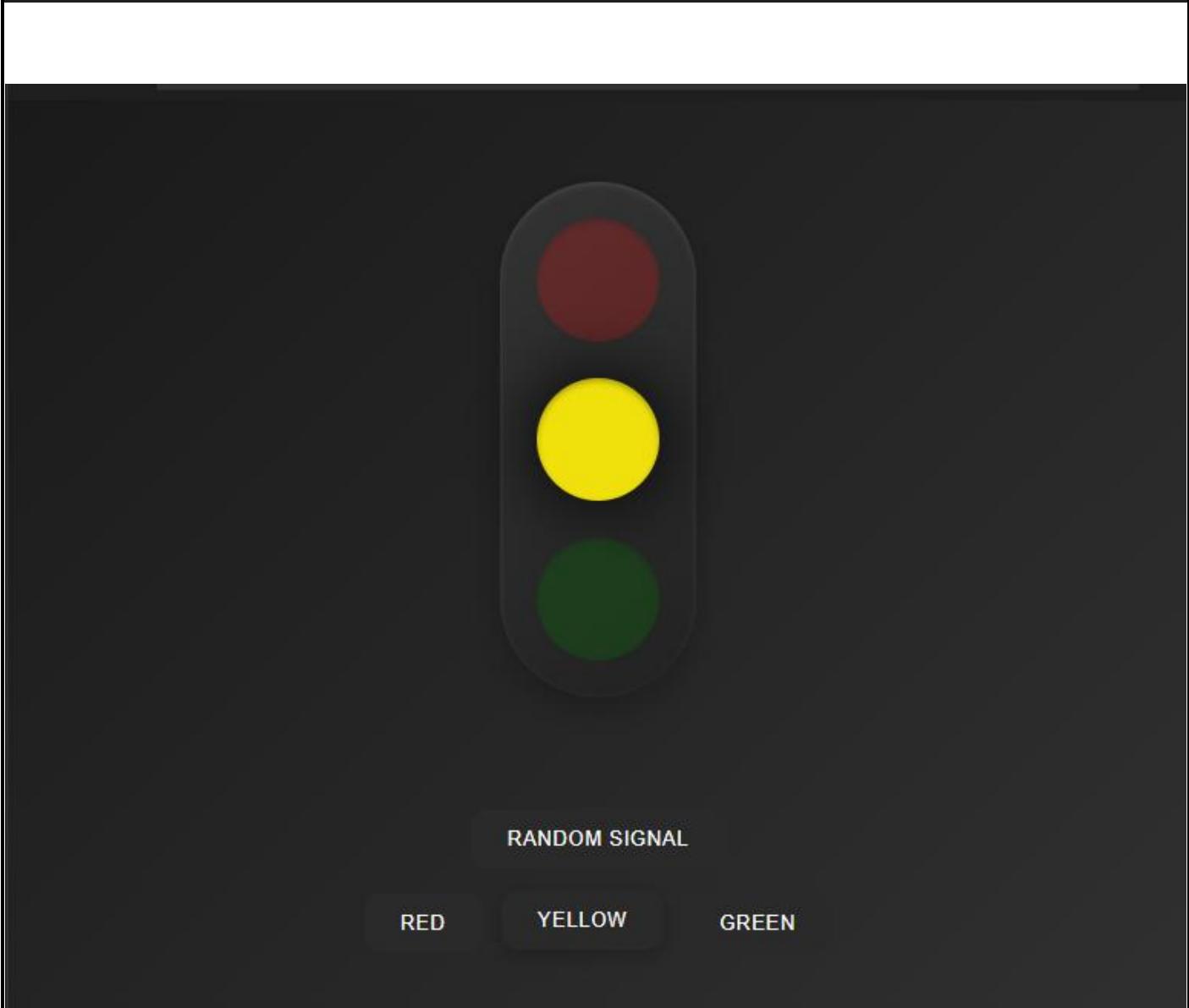
    // Event delegation for color buttons
    document.querySelector('.button-group').addEventListener('click', (e) => {
```

```

        if (e.target.matches('[data-color]')) {
            activateSignal(e.target.dataset.color);
        }
    });

// Random signal button
document.getElementById('random').addEventListener('click', () => {
    const colors = Object.keys(signals);
    const randomColor = colors[Math.floor(Math.random() * colors.length)];
    activateSignal(randomColor);
});
</script>
</body>
</html>

```



## 2. Random Quote

<https://23bai1214-divyanshu-exercise-5-t-2.netlify.app/>

```
<!-- 2. Random Quote -->
```

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Quote Generator</title>
  <style>
    :root {
      --primary: #2a2d4c;
      --text: #333;
      --accent: #6366f1;
      --background-start: #667eea;
      --background-end: #764ba2;
      --card: #ffffff;
    }

    * {
      margin: 0;
      padding: 0;
      box-sizing: border-box;
    }

    body {
      min-height: 100vh;
      display: grid;
      place-items: center;
      font-family: system-ui, -apple-system, sans-serif;
      background: linear-gradient(135deg, var(--background-start), var(--background-end));
      padding: 1rem;
    }

    .container {
      max-width: 600px;
      width: 100%;
      padding: 2rem;
      background: var(--card);
      border-radius: 1rem;
      box-shadow: 0 10px 25px rgba(0, 0, 0, 0.1);
      display: flex;
      flex-direction: column;
      gap: 1.5rem;
    }

    .quote-box {
      position: relative;
      padding: 1rem;
    }

    .quote-text {
      font-size: 1.5rem;
      color: var(--primary);
      line-height: 1.5;
      margin-bottom: 1rem;
    }
  </style>

```

```
    position: relative;
    padding: 0 1rem;
    transition: opacity 0.3s ease;
}

.quote-text::before,
.quote-text::after {
    content: '';
    font-size: 4rem;
    position: absolute;
    opacity: 0.1;
    font-family: Georgia, serif;
}

.quote-text::before {
    top: -1rem;
    left: -1rem;
}

.quote-text::after {
    bottom: -2rem;
    right: -1rem;
}

.quote-author {
    font-size: 1.1rem;
    color: var(--accent);
    text-align: right;
    font-style: italic;
    transition: opacity 0.3s ease;
}

.button {
    background-color: var(--accent);
    color: white;
    border: none;
    padding: 1rem 2rem;
    border-radius: 0.5rem;
    font-size: 1rem;
    cursor: pointer;
    transition: transform 0.2s ease, box-shadow 0.2s ease;
    font-weight: 500;
    align-self: center;
}

.button:hover {
    transform: translateY(-2px);
    box-shadow: 0 4px 12px rgba(99, 102, 241, 0.3);
}

.button:active {
    transform: translateY(0);
}
```

```
@media (max-width: 480px) {
    .container {
        padding: 1.5rem;
    }

    .quote-text {
        font-size: 1.25rem;
    }
}
</style>
</head>
<body>
    <div class="container">
        <div class="quote-box">
            <p class="quote-text" id="quote"></p>
            <p class="quote-author" id="author"></p>
        </div>
        <button class="button" id="new-quote">New Quote</button>
    </div>

    <script>
        const quotes = [
            { text: "The only way to do great work is to love what you do.", author: "Steve Jobs" },
            { text: "In three words I can sum up everything I've learned about life: it goes on.", author: "Robert Frost" },
            { text: "Success is not final, failure is not fatal: it is the courage to continue that counts.", author: "Winston Churchill" },
            { text: "The future belongs to those who believe in the beauty of their dreams.", author: "Eleanor Roosevelt" },
            { text: "Be yourself; everyone else is already taken.", author: "Oscar Wilde" },
            { text: "Two things are infinite: the universe and human stupidity; and I'm not sure about the universe.", author: "Albert Einstein" },
            { text: "Be the change that you wish to see in the world.", author: "Mahatma Gandhi" },
            { text: "The only impossible journey is the one you never begin.", author: "Tony Robbins" },
            { text: "To be yourself in a world that is constantly trying to make you something else is the greatest accomplishment.", author: "Ralph Waldo Emerson" },
            { text: "Everything you can imagine is real.", author: "Pablo Picasso" }
        ];

        const quoteText = document.getElementById('quote');
        const quoteAuthor = document.getElementById('author');
        const newQuoteButton = document.getElementById('new-quote');

        function getRandomQuote() {
            return quotes[Math.floor(Math.random() * quotes.length)];
        }

        function displayQuote() {
            const { text, author } = getRandomQuote();
            quoteText.style.opacity = 0;
```

```
        quoteAuthor.style.opacity = 0;
        setTimeout(() => {
            quoteText.textContent = text;
            quoteAuthor.textContent = `— ${author}`;
            quoteText.style.opacity = 1;
            quoteAuthor.style.opacity = 1;
        }, 300);
    }

    displayQuote();
    newQuoteButton.addEventListener('click', displayQuote);
</script>
</body>
</html>
```

Two things are infinite: the universe and human stupidity; and I'm not sure about the universe.

— Albert Einstein

New Quote

### 3. To-Do-List

<https://23bai1214-divyanshu-exercise-5-t-3.netlify.app/>

```
<!-- 3. To-Do-List -->

<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>To-Do List</title>
    <style>
```

```
:root {
    --primary-bg: #4A90E2;
    --container-bg: #ffffff;
    --text-color: #333;
    --button-bg: #ff4081;
    --button-text: #fff;
    --task-bg: #f1f1f1;
    --task-done: #bbb;
}

* {
    margin: 0;
    padding: 0;
    box-sizing: border-box;
    font-family: Arial, sans-serif;
}

body {
    display: flex;
    justify-content: center;
    align-items: center;
    height: 100vh;
    background: linear-gradient(135deg, var(--primary-bg), #D4418E);
    padding: 1rem;
}

.container {
    background: var(--container-bg);
    padding: 2rem;
    border-radius: 1rem;
    box-shadow: 0 10px 25px rgba(0, 0, 0, 0.1);
    width: 350px;
    text-align: center;
}

h2 {
    margin-bottom: 1rem;
    color: var(--text-color);
}

input {
    width: 80%;
    padding: 10px;
    margin-bottom: 1rem;
    border: 1px solid #ccc;
    border-radius: 5px;
    outline: none;
}

button {
    padding: 10px 15px;
    background: var(--button-bg);
    color: var(--button-text);
    border: none;
}
```

```
        cursor: pointer;
        border-radius: 5px;
        transition: background 0.3s;
    }

    button:hover {
        background: #e60073;
    }

    ul {
        list-style: none;
        padding: 0;
    }

    li {
        display: flex;
        justify-content: space-between;
        padding: 10px;
        background: var(--task-bg);
        margin-top: 5px;
        border-radius: 5px;
        cursor: pointer;
        transition: background 0.3s;
    }

    li:hover {
        background: #e0e0e0;
    }

    .done {
        text-decoration: line-through;
        color: var(--task-done);
    }

    .delete-btn {
        background: transparent;
        border: none;
        color: red;
        font-size: 1rem;
        cursor: pointer;
    }

```

</style>

```
</head>
<body>
    <div class="container">
        <h2>To-Do List</h2>
        <input type="text" id="task" placeholder="Add a new task...">
        <button onclick="addTask()">Add</button>
        <ul id="taskList"></ul>
    </div>
    <script>
        function addTask() {
            let taskInput = document.getElementById("task");
            let taskText = taskInput.value.trim();

```

```

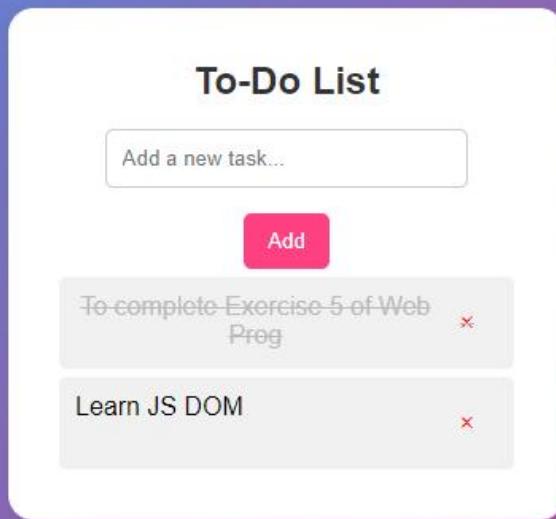
if (taskText === "") return;

let li = document.createElement("li");
li.innerHTML = `${taskText} <button class='delete-btn' onclick='removeTask(this)'>&times;</button>`;
li.addEventListener("click", function() {
    this.classList.toggle("done");
});

document.getElementById("taskList").appendChild(li);
taskInput.value = "";
}

function removeTask(button) {
    button.parentElement.remove();
}
</script>
</body>
</html>

```



## 4. Drawing Shape

<https://23bai1214-divyanshu-exercise-5-t-4.netlify.app/>

```

<!-- 4. Drawing Shapes -->

<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Canvas Shape Drawer</title>

```

```
<style>
  * {
    margin: 0;
    padding: 0;
    box-sizing: border-box;
  }

  body {
    font-family: Arial, sans-serif;
    background: linear-gradient(to right, #FF9A8B, #FF6A88, #FF99AC);
    height: 100vh;
    display: flex;
    flex-direction: column;
    justify-content: center;
    align-items: center;
    text-align: center;
  }

  h1 {
    color: white;
    margin-bottom: 15px;
  }

  .options {
    margin-bottom: 15px;
  }

  button {
    margin: 5px;
    padding: 12px 20px;
    font-size: 16px;
    border: none;
    outline: none;
    background: #4A90E2;
    color: white;
    border-radius: 8px;
    cursor: pointer;
    transition: 0.3s ease-in-out;
  }

  button:hover {
    background: #357ABD;
  }

  canvas {
    background-color: white;
    border: 3px solid black;
    border-radius: 10px;
    box-shadow: 0px 4px 10px rgba(0, 0, 0, 0.2);
  }
</style>
</head>
<body>
```

```
<h1>Draw Random Shapes on Canvas</h1>
<div class="options">
  <button onclick="drawRectangle()">Rectangle</button>
  <button onclick="drawCircle()">Circle</button>
  <button onclick="drawTriangle()">Triangle</button>
  <button onclick="clearCanvas()">Clear</button>
</div>
<canvas id="myCanvas" width="500" height="500"></canvas>

<script>
  const canvas = document.getElementById('myCanvas');
  const ctx = canvas.getContext('2d');

  function getRandomInt(min, max) {
    return Math.floor(Math.random() * (max - min + 1)) + min;
  }

  function getRandomColor() {
    return `hsl(${Math.random() * 360}, 70%, 60%)`;
  }

  function drawRectangle() {
    const x = getRandomInt(0, canvas.width - 150);
    const y = getRandomInt(0, canvas.height - 100);
    ctx.fillStyle = getRandomColor();
    ctx.fillRect(x, y, 150, 100);
  }

  function drawCircle() {
    const radius = 50;
    const x = getRandomInt(radius, canvas.width - radius);
    const y = getRandomInt(radius, canvas.height - radius);
    ctx.fillStyle = getRandomColor();
    ctx.beginPath();
    ctx.arc(x, y, radius, 0, Math.PI * 2);
    ctx.fill();
  }

  function drawTriangle() {
    const x1 = getRandomInt(50, canvas.width - 50);
    const y1 = getRandomInt(50, canvas.height - 50);
    const x2 = x1 - 50;
    const y2 = y1 + 100;
    const x3 = x1 + 50;
    const y3 = y1 + 100;
    ctx.fillStyle = getRandomColor();
    ctx.beginPath();
    ctx.moveTo(x1, y1);
    ctx.lineTo(x2, y2);
    ctx.lineTo(x3, y3);
    ctx.closePath();
    ctx.fill();
  }
</script>
```

```
function clearCanvas() {
    ctx.clearRect(0, 0, canvas.width, canvas.height);
}
</script>

</body>
</html>
```

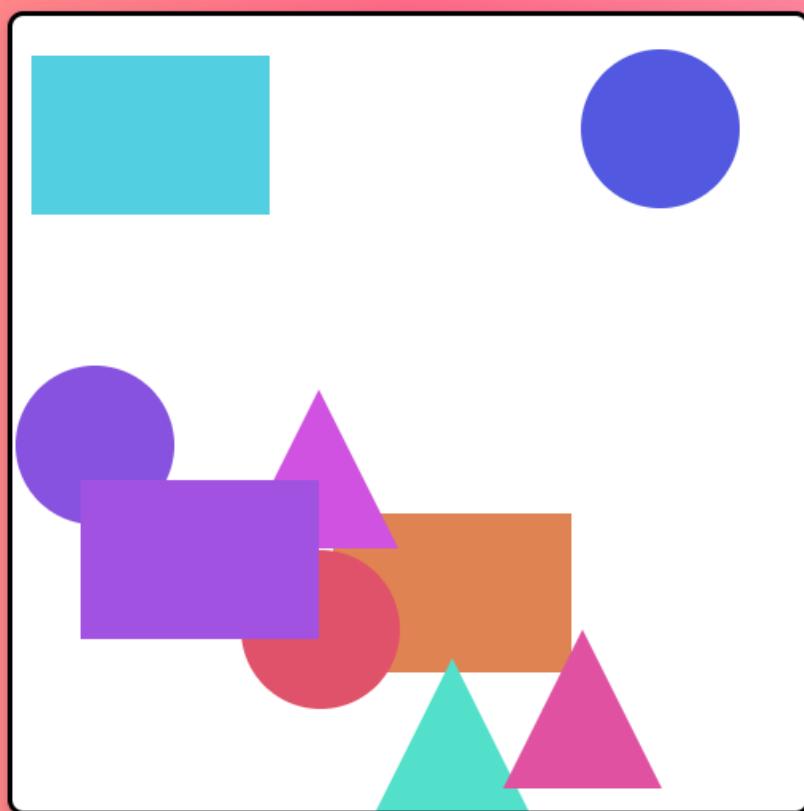
## Draw Random Shapes on Canvas

Rectangle

Circle

Triangle

Clear



## 5. Guess The Number

<https://23bai1214-divyanshu-exercise-5-t-5.netlify.app/>

```
<!-- 5. Guess The Number -->

<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Guess the Number</title>
```

```
<style>
:root {
    --primary-bg: #4A90E2;
    --container-bg: #ffffff;
    --text-color: #333;
    --button-bg: #ff4081;
    --button-text: #fff;
}

* {
    margin: 0;
    padding: 0;
    box-sizing: border-box;
    font-family: Arial, sans-serif;
}

body {
    display: flex;
    justify-content: center;
    align-items: center;
    height: 100vh;
    background: linear-gradient(135deg, var(--primary-bg), #D4418E);
    padding: 1rem;
}

.container {
    background: var(--container-bg);
    padding: 2rem;
    border-radius: 1rem;
    box-shadow: 0 10px 25px rgba(0, 0, 0, 0.1);
    width: 350px;
    text-align: center;
}

h2 {
    margin-bottom: 1rem;
    color: var(--text-color);
}

input {
    width: 80%;
    padding: 10px;
    margin-bottom: 1rem;
    border: 1px solid #ccc;
    border-radius: 5px;
    outline: none;
}

button {
    padding: 10px 15px;
    background: var(--button-bg);
    color: var(--button-text);
    border: none;
    cursor: pointer;
```

```
        border-radius: 5px;
        transition: background 0.3s;
    }

    button:hover {
        background: #e60073;
    }

    .message {
        margin-top: 1rem;
        font-size: 1.1rem;
        color: var(--text-color);
    }

```

</style>

```
</head>
<body>
    <div class="container">
        <h2>Guess the Number</h2>
        <p>Guess a number between 1 and 100</p>
        <input type="number" id="guess" placeholder="Enter your guess...">
        <button onclick="checkGuess()">Submit</button>
        <p class="message" id="message"></p>
        <p>Attempts: <span id="attempts">0</span></p>
        <button onclick="resetGame()">Restart</button>
    </div>

    <script>
        let randomNumber = Math.floor(Math.random() * 100) + 1;
        let attempts = 0;

        function checkGuess() {
            let userGuess = parseInt(document.getElementById("guess").value);
            let message = document.getElementById("message");
            let attemptCounter = document.getElementById("attempts");

            if (isNaN(userGuess) || userGuess < 1 || userGuess > 100) {
                message.textContent = "Please enter a valid number between 1 and 100.";
                return;
            }

            attempts++;
            attemptCounter.textContent = attempts;

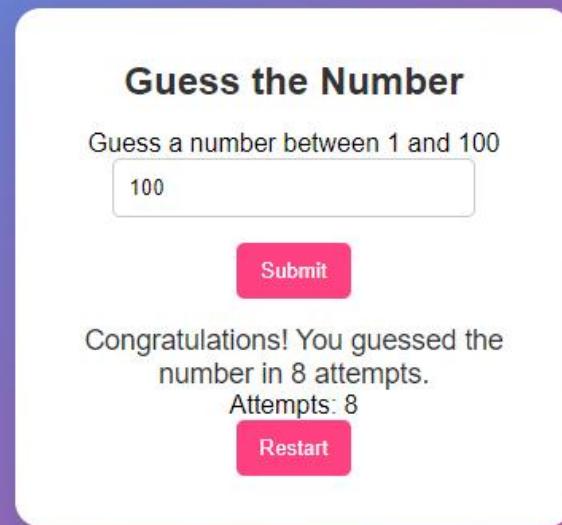
            if (userGuess === randomNumber) {
                message.textContent = `Congratulations! You guessed the number in ${attempts} attempts.`;
            } else if (userGuess < randomNumber) {
                message.textContent = "Too low! Try again.";
            } else {
                message.textContent = "Too high! Try again.";
            }
        }

        function resetGame() {
    
```

```

        randomNumber = Math.floor(Math.random() * 100) + 1;
        attempts = 0;
        document.getElementById("attempts").textContent = attempts;
        document.getElementById("message").textContent = "";
        document.getElementById("guess").value = "";
    }
</script>
</body>
</html>

```



## 6. Random Emoji Generator

<https://23bai1214-divyanshu-exercise-5-t-6.netlify.app/>

```

<!-- 6. Random Emoji Generator -->

<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Random Emoji Generator</title>
    <style>
        :root {
            --primary-bg: #4A90E2;
            --container-bg: #ffffff;
            --text-color: #333;
            --button-bg: #ff4081;
            --button-text: #fff;
        }

        *

```

```
        margin: 0;
        padding: 0;
        box-sizing: border-box;
        font-family: Arial, sans-serif;
    }

body {
    display: flex;
    justify-content: center;
    align-items: center;
    height: 100vh;
    background: linear-gradient(135deg, var(--primary-bg), #D4418E);
    padding: 1rem;
}

.container {
    background: var(--container-bg);
    padding: 2rem;
    border-radius: 1rem;
    box-shadow: 0 10px 25px rgba(0, 0, 0, 0.1);
    width: 350px;
    text-align: center;
}

h2 {
    margin-bottom: 1rem;
    color: var(--text-color);
}

.emoji-display {
    font-size: 4rem;
    margin: 1rem 0;
}

button {
    padding: 10px 15px;
    background: var(--button-bg);
    color: var(--button-text);
    border: none;
    cursor: pointer;
    border-radius: 5px;
    transition: background 0.3s;
    font-size: 1rem;
}

button:hover {
    background: #e60073;
}

</style>
</head>
<body>
    <div class="container">
        <h2>Random Emoji Generator</h2>
        <div class="emoji-display" id="emoji">☺</div>
    </div>
</body>
```

