THE STATE UNIVERSITY OF ZANZIBAR
DEPARTMENT OF COMPUTER SCIENCE AND INFORMATION TECHNOLOGY.

NAME	REG NO
ASHURA SALUM KHAMIS	BCS/18/23/011/TZ

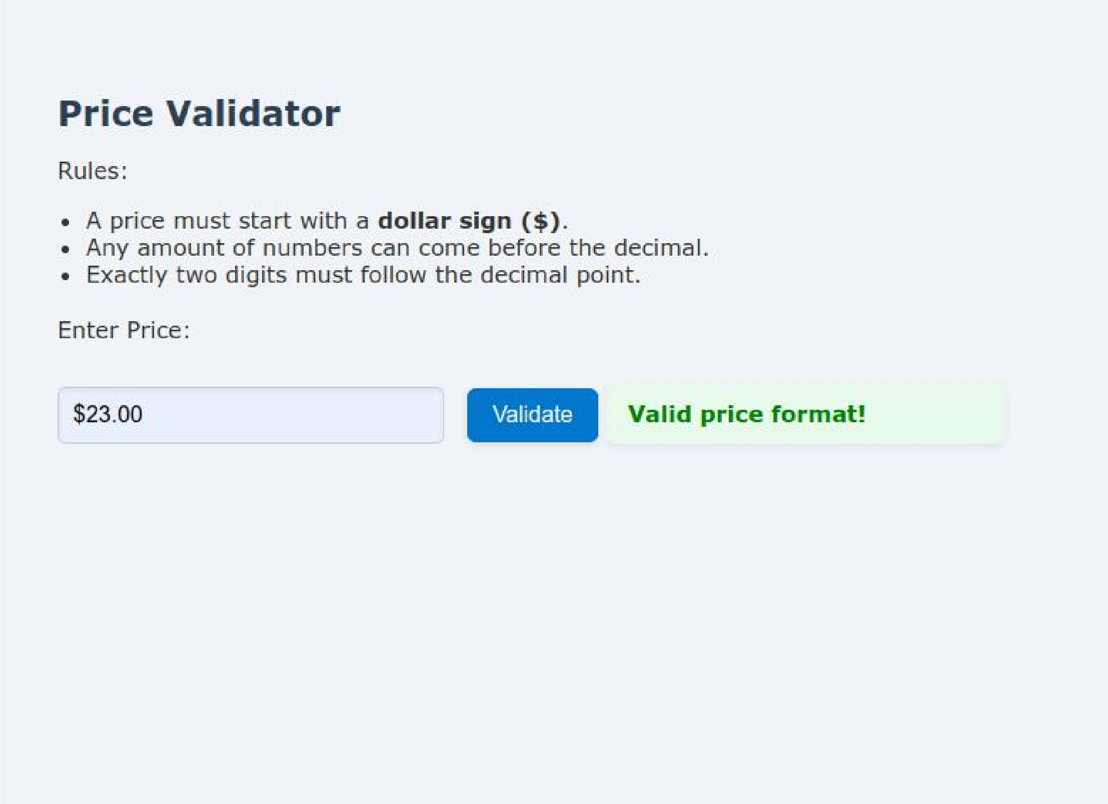
QUESTION 1:

Expression :Regex = /^\$\d*\.\d{2}$/;

Tested program

```
<!DOCTYPE html>
<html>
<head>
  <title>Price Validator</title>
  <style>
    body {
      font-family: 'Segoe UI', Tahoma, Geneva, Verdana, sans-serif;
      padding: 40px;
      background-color: #f0f4f8;
      color: #333;
    }

    h2 {
      color: #2c3e50;
      margin-bottom: 10px;
    }

    p {
      margin-bottom: 5px;
    }

    ul {
      margin-bottom: 20px;
      padding-left: 20px;
    }

    label {
      font-size: 16px;
      display: block;
      margin-bottom: 8px;
    }

    input[type="text"] {
      padding: 10px;
      width: 250px;
      font-size: 16px;
      border: 1px solid #ccc;
      border-radius: 6px;
      box-shadow: inset 0 2px 4px rgba(0, 0, 0, 0.05);
      transition: border-color 0.3s ease;
    }

    input[type="text"]:focus {
      border-color: #0077cc;
      outline: none;
    }

    button {
      padding: 10px 18px;
      font-size: 16px;
      margin-left: 10px;
      background-color: #0077cc;
      color: white;
      border: none;
      border-radius: 6px;
      cursor: pointer;
      box-shadow: 0 2px 5px rgba(0, 0, 0, 0.1);
      transition: background 0.3s ease;
    }

    button:hover {
      background-color: #005fa3;
    }

    #result {
      margin-top: 20px;
      font-weight: bold;
      padding: 12px 15px;
      border-radius: 6px;
      display: inline-block;
      min-width: 250px;
      box-shadow: 0 2px 6px rgba(0, 0, 0, 0.08);
    }
  </style>
</head>
<body>

  <h2> Price Validator</h2>
  <p>Rules:</p>
  <ul>
    <li>A price must start with a <strong>dollar sign ($)</strong>.</li>
    <li>Any amount of numbers can come before the decimal.</li>
    <li>Exactly two digits must follow the decimal point.</li>
  </ul>

  <label for="priceInput">Enter Price:</label>
  <input type="text" id="priceInput" placeholder="$00.00">
  <button onclick="validatePrice()">Validate</button>

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

  <script>
    function validatePrice() {
      const input = document.getElementById('priceInput').value;
      const result = document.getElementById('result');

           const regex = /^\$\d*\.\d{2}$/;

      if (regex.test(input)) {
        result.textContent = "Valid price format!";
        result.style.color = "green";
        result.style.backgroundColor = "#e6f9ea";
      } else {
        result.textContent = " Invalid price format!";
        result.style.color = "red";
        result.style.backgroundColor = "#ffe6e6";
      }
    }
    </script>

</body>
</html>
```

Valid out put



Price Validator

Rules:

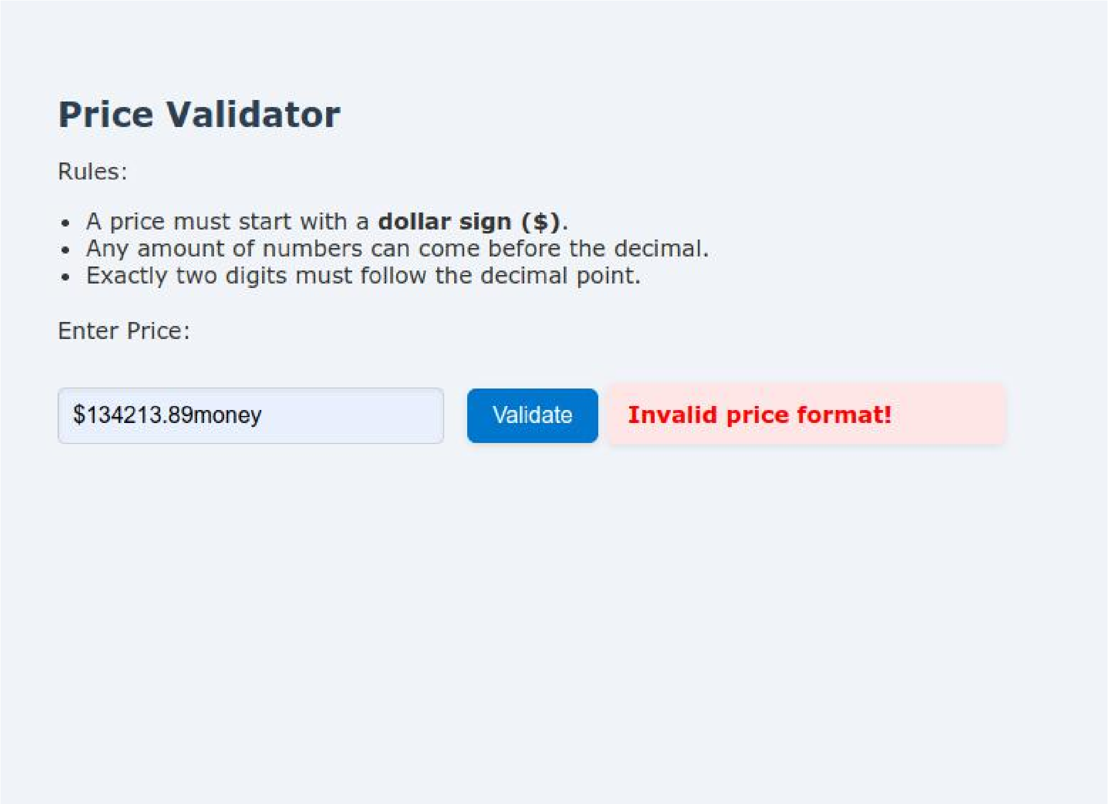
- A price must start with a **dollar sign ($)**.
- Any amount of numbers can come before the decimal.
- Exactly two digits must follow the decimal point.

Enter Price:

$23.00

Valid price format!

Invalid output



Price Validator

Rules:

- A price must start with a **dollar sign ($)**.
- Any amount of numbers can come before the decimal.
- Exactly two digits must follow the decimal point.

Enter Price:

$134213.89money

Invalid price format!

Question 2:

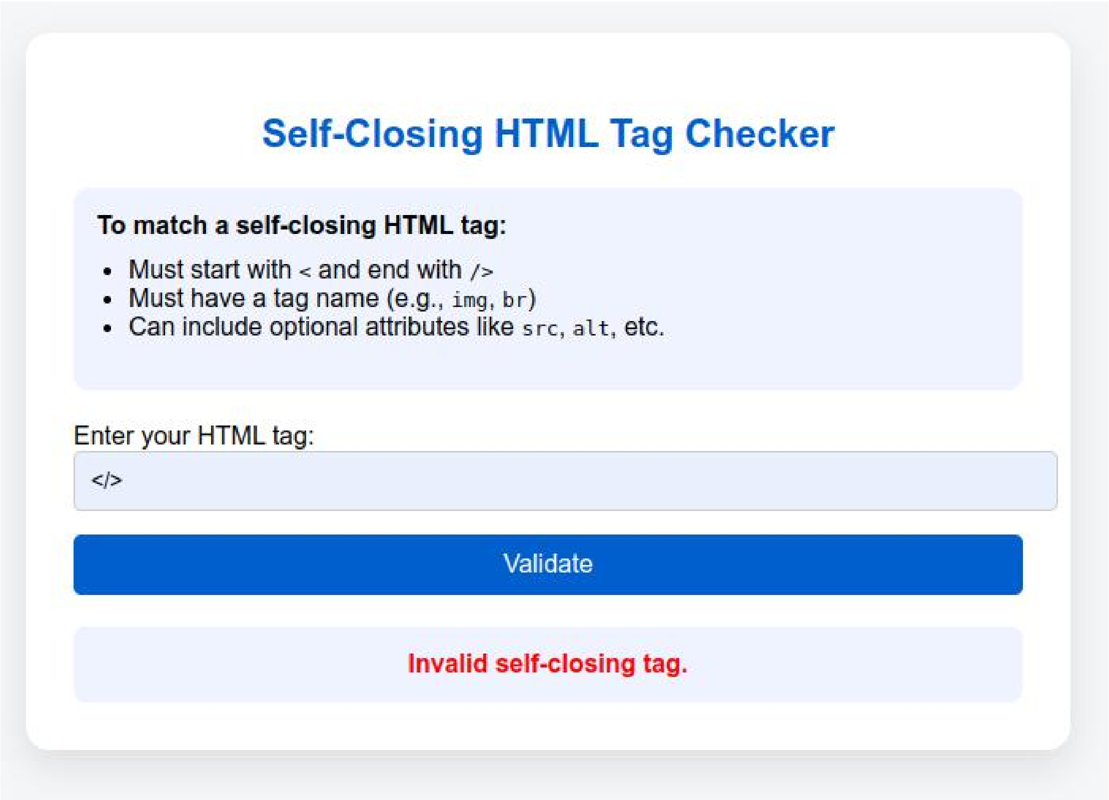
Expression:
regex = /<\s*[a-zA-Z][a-zA-Z0-9]*\s*(\s+[a-zA-Z_:][a-zA-Z0-9_\-:.]*\s*=\s*"[^"]*"\s*)*\s*\/>/g;

Tested program

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>Self-Closing HTML Tag Checker</title>
  <style>
    body {
      font-family: Arial, sans-serif;
      margin: 40px;
      background-color: #f4f6f8;
      padding: 40px;
      color: black;
    }

    .container {
      max-width: 600px;
      margin: auto;
      background: #ffffff;
      border-radius: 15px;
      box-shadow: 0 10px 25px rgba(0, 0, 0, 0.1);
      padding: 30px;
    }

    h2 {
      color: #005fcc;
      text-align: center;
      margin-bottom: 20px;
    }

    .explanation {
      background-color: #eef3ff;
      padding: 15px;
      border-radius: 10px;
      margin-bottom: 20px;
    }

    .explanation strong {
      display: block;
      margin-bottom: 10px;
      font-size: 16px;
    }

    ul {
      margin-top: 0;
      padding-left: 20px;
    }

    input[type="text"] {
      width: 100%;
      padding: 10px;
      font-size: 14px;
      margin-bottom: 15px;
      border-radius: 5px;
      border: 1px solid #ccc;
    }

    button {
      padding: 10px 20px;
      font-size: 16px;
      background-color: #005fcc;
      color: white;
      border: none;
      border-radius: 5px;
      cursor: pointer;
      width: 100%;
    }

    button:hover {
      background-color: #004bb5;
    }

    .output {
      background-color: #eef3ff;
      margin-top: 20px;
      padding: 15px;
      border-radius: 8px;
      text-align: center;
      font-weight: bold;
    }
```

```
  </style>
</head>
<body>

  <div class="container">
    <h2>Self-Closing HTML Tag Checker</h2>

    <div class="explanation">
      <strong>To match a self-closing HTML tag:</strong>
      <ul>
        <li>Must start with <code>&lt;</code> and end with <code>/&gt;</code></li>
        <li>Must have a tag name (e.g., <code>img</code>, <code>br</code>)</li>
        <li>Can include optional attributes like <code>src</code>, <code>alt</code>,
etc.</li>
      </ul>
    </div>

    <label for="htmlInput">Enter your HTML tag:</label>
    <input type="text" id="htmlInput" placeholder='<img src="image.jpg" />'>

    <button onclick="checkTags()">Validate</button>

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

  <script>
    function checkTags() {
      const input = document.getElementById("htmlInput").value.trim();
      const result = document.getElementById("result");
      result.innerHTML = "";

      const regex =
/^<\s*[a-zA-Z][a-zA-Z0-9]*\s*(\s+[a-zA-Z_:][a-zA-Z0-9_\-:.]*\s*=\s*"[^"]*")*\s*\/>$/;

      if (regex.test(input)) {
        result.innerHTML = `<span style="color:green;"> Valid self-closing tag!</span>`;
      } else {
        result.innerHTML = `<span style="color:red;"> Invalid self-closing tag.</span>`;
      }
    }
  </script>
</body>
</html>
```

Valid output



Self-Closing HTML Tag Checker

To match a self-closing HTML tag:

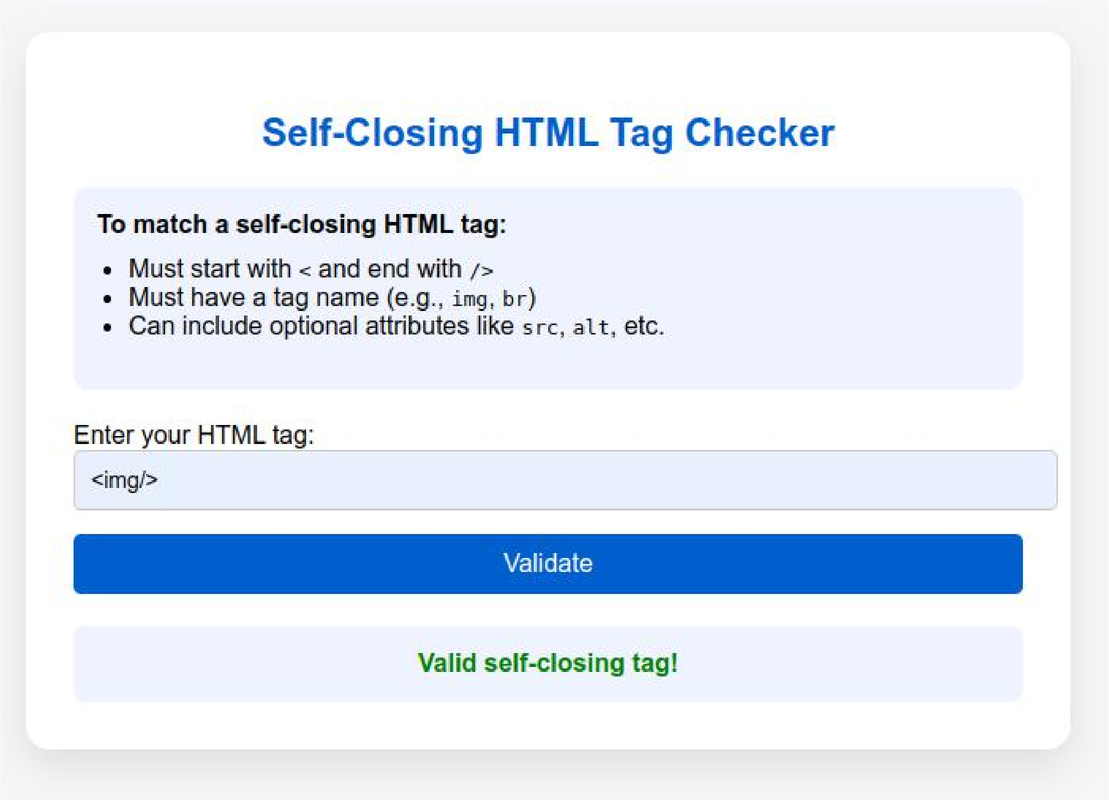
- Must start with `<` and end with `/>`
- Must have a tag name (e.g., `img`, `br`)
- Can include optional attributes like `src`, `alt`, etc.

Enter your HTML tag:

```
</>
```

Invalid self-closing tag.

Invalid output



Self-Closing HTML Tag Checker

To match a self-closing HTML tag:

- Must start with `<` and end with `/>`
- Must have a tag name (e.g., `img`, `br`)
- Can include optional attributes like `src`, `alt`, etc.

Enter your HTML tag:

```
<img/>
```

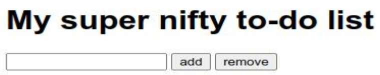
Valid self-closing tag!

Question 3

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>My Super Nifty To-Do List</title>
  <style>
    body {
      font-family: sans-serif;
      margin: 30px;
    }
    li.white {
      background-color: white;
    }
    li.yellow {
      background-color: yellow;
    }
  </style>
</head>
<body>

  <h1>My super nifty to-do list</h1>
  <ul id="list"></ul>

  <div>
    <input type="text" id="item" />
    <button id="add">add</button>
    <button id="remove">remove</button>
  </div>

  <script>
    const list = document.getElementById("list");
    const input = document.getElementById("item");
    const addBtn = document.getElementById("add");
    const removeBtn = document.getElementById("remove");

    function updateColors() {
      const items = list.getElementsByTagName("li");
      for (let i = 0; i < items.length; i++) {
        items[i].className = (i % 2 === 0) ? "white" : "yellow";
      }
    }

    addBtn.onclick = function () {
      const value = input.value.trim();
      if (value) {
        const li = document.createElement("li");
        li.textContent = value;
        list.appendChild(li);
        updateColors();
        input.value = "";
      }
    };
    removeBtn.onclick = function () {
      const value = input.value.trim().toLowerCase();
      if (value) {
        const items = list.getElementsByTagName("li");
        for (let i = 0; i < items.length; i++) {
          if (items[i].textContent.toLowerCase() === value) {
```

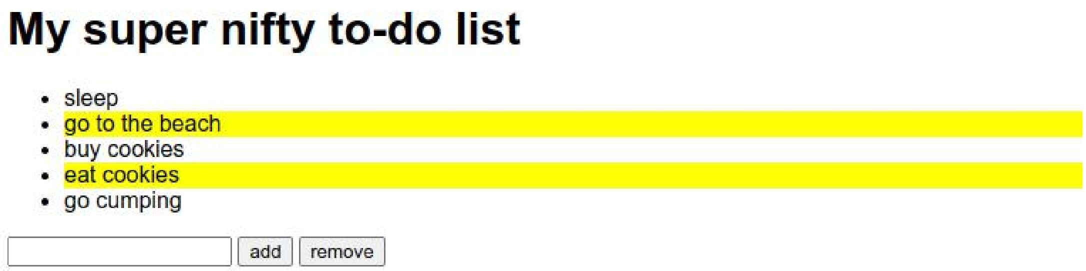
```
                        list.removeChild(items[i]);
                        break;
                    }
                }
                updateColors();
                input.value = "";
            }
        };
    </script>

</body>
</html>
```

Before anything happen



My super nifty to-do list

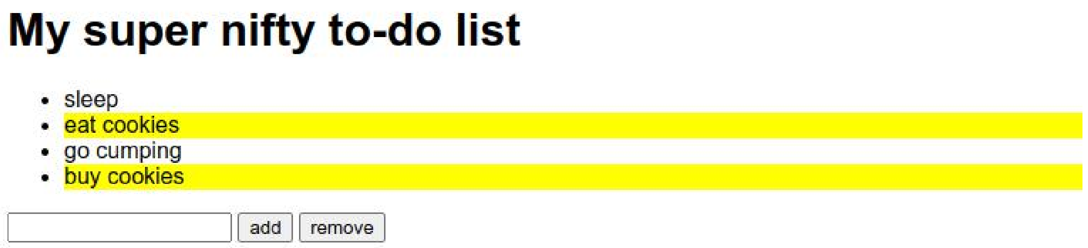
After 5 iterm added none remove



My super nifty to-do list

- sleep
- go to the beach
- buy cookies
- eat cookies
- go cumping

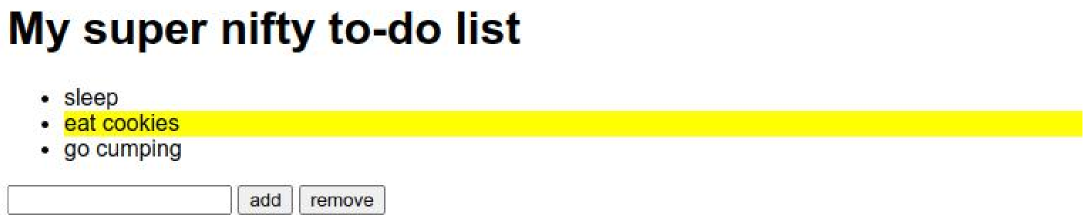
After remove “go to the beach”



My super nifty to-do list

- sleep
- eat cookies
- go cumping
- buy cookies

After remove “buy cookies”



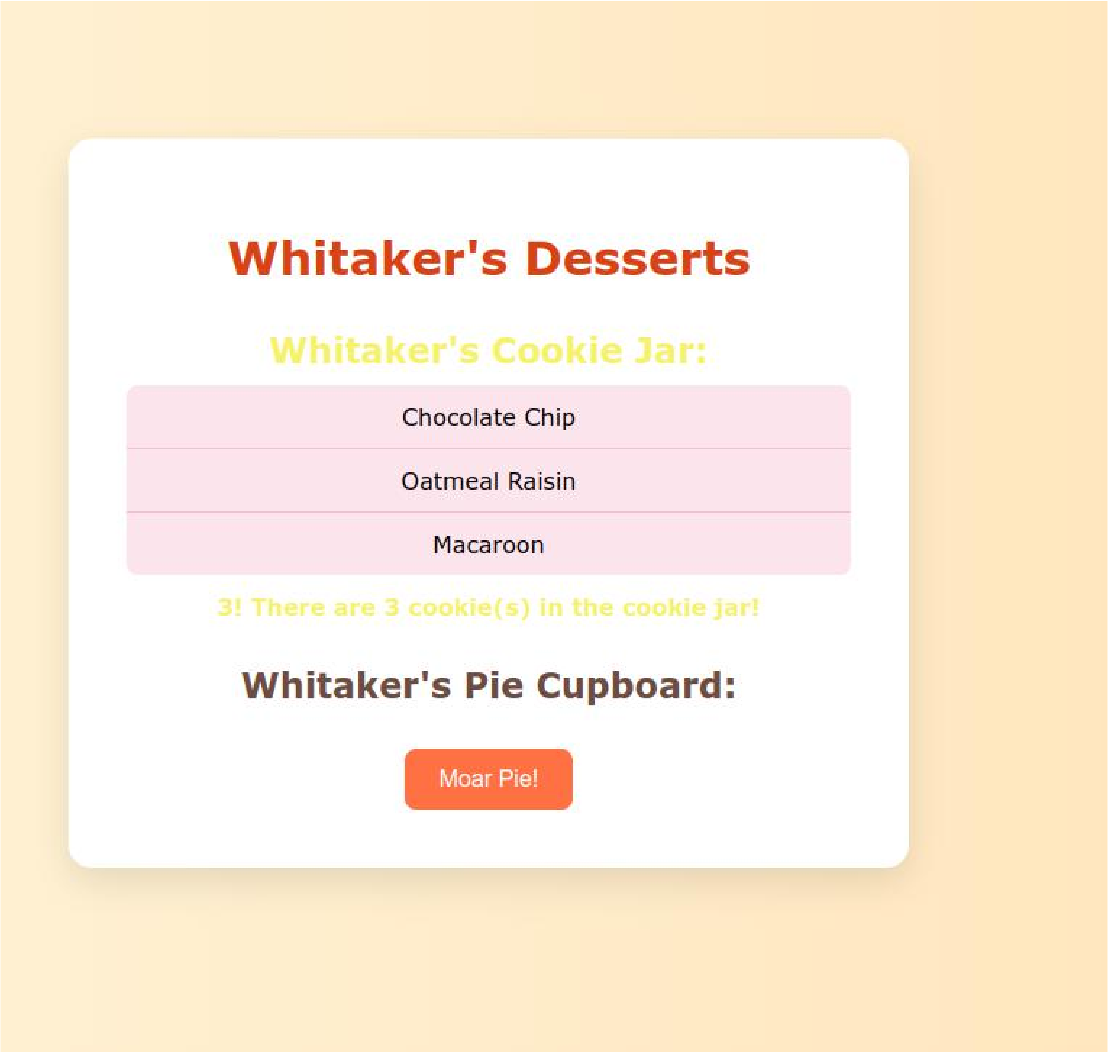
My super nifty to-do list

- sleep
- eat cookies
- go cumping

Question 4:

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>Whitaker's Desserts</title>
  <style>
    body {
      font-family: 'Segoe UI', Tahoma, Geneva, Verdana, sans-serif;
      background: linear-gradient(to right, #fff8e1, #ffe0b2);
      display: flex;
      justify-content: center;
      align-items: center;
      min-height: 100vh;
      margin: 0;
    }

    .container {
      background: #ffffff;
      padding: 40px;
      border-radius: 16px;
      box-shadow: 0 10px 25px rgba(0, 0, 0, 0.1);
      max-width: 500px;
      width: 100%;
      text-align: center;
    }

    h1 {
      color: #d84315;
      margin-bottom: 20px;
    }

    h2 {
      color: #6d4c41;
      margin-top: 30px;
      margin-bottom: 10px;
    }

    ul {
      list-style: none;
      padding: 0;
      margin: 0;
      background-color: #fce4ec;
      border-radius: 8px;
      overflow: hidden;
    }

    li {
      padding: 12px 16px;
      border-bottom: 1px solid #f8bbd0;
    }

    li:last-child {
      border-bottom: none;
    }

    #cookie-count {
      margin-top: 12px;
      font-weight: bold;
      font-size: 16px;
      color: #f7f16d;
    }

    #moar-pie {
      margin-top: 20px;
      padding: 12px 24px;
      background-color: #ff7043;
      color: white;
      border: none;
      border-radius: 8px;
      cursor: pointer;
      font-size: 16px;
      transition: background-color 0.3s ease;
    }

    #moar-pie:hover {
      background-color: #d84315;
    }
  </style>
</head>
<body>
  <div class="container">
    <h1>Whitaker's Desserts</h1>

    <h2 id="cookie-header">Whitaker's Cookie Jar:</h2>
    <ul id="cookie-jar">
      <li class="cookie">Chocolate Chip</li>
      <li class="cookie">Oatmeal Raisin</li>
      <li class="cookie">Macaroon</li>
    </ul>
    <p id="cookie-count"></p>

    <h2>Whitaker's Pie Cupboard:</h2>
    <ul id="pie-cupboard"></ul>
    <button id="moar-pie">Moar Pie!</button>
  </div>

  <script>
    document.addEventListener('DOMContentLoaded', () => {
      const cookieHeader = document.getElementById('cookie-header');
      const cookieCount = document.getElementById('cookie-count');
      const cookieJar = document.getElementById('cookie-jar');

      // Question (a)
      cookieHeader.style.color = '#f7f16d';

      // question (b)
      function updateCookieCount() {
        const cookies = cookieJar.getElementsByClassName('cookie');
        const count = cookies.length;
        cookieCount.textContent = `${count}! There are ${count} cookie(s) in the cookie jar!`;
        cookieCount.style.color = '#f7f16d';
      }

      updateCookieCount();

      // question(c)
      setInterval(() => {
        const cookies = cookieJar.getElementsByClassName('cookie');
        if (cookies.length > 0) {
                cookieJar.removeChild(cookies[cookies.length - 1]);
                updateCookieCount();
            }
        }, 30000);
    });
  </script>
</body>
</html>
```

Out put



Whitaker's Desserts

Whitaker's Cookie Jar:

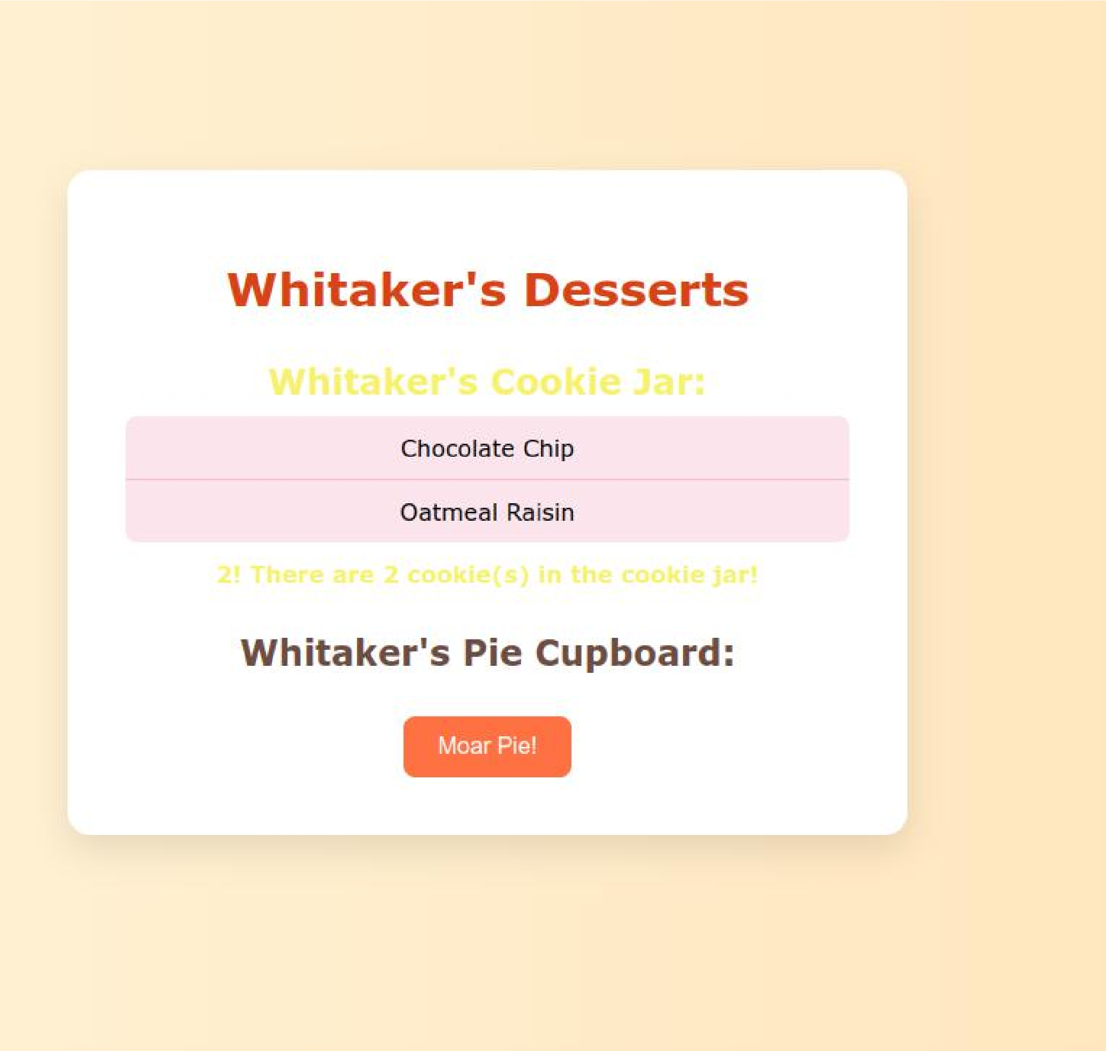
Chocolate Chip

Oatmeal Raisin

Macaroon

3! There are 3 cookie(s) in the cookie jar!

Whitaker's Pie Cupboard:



Whitaker's Desserts

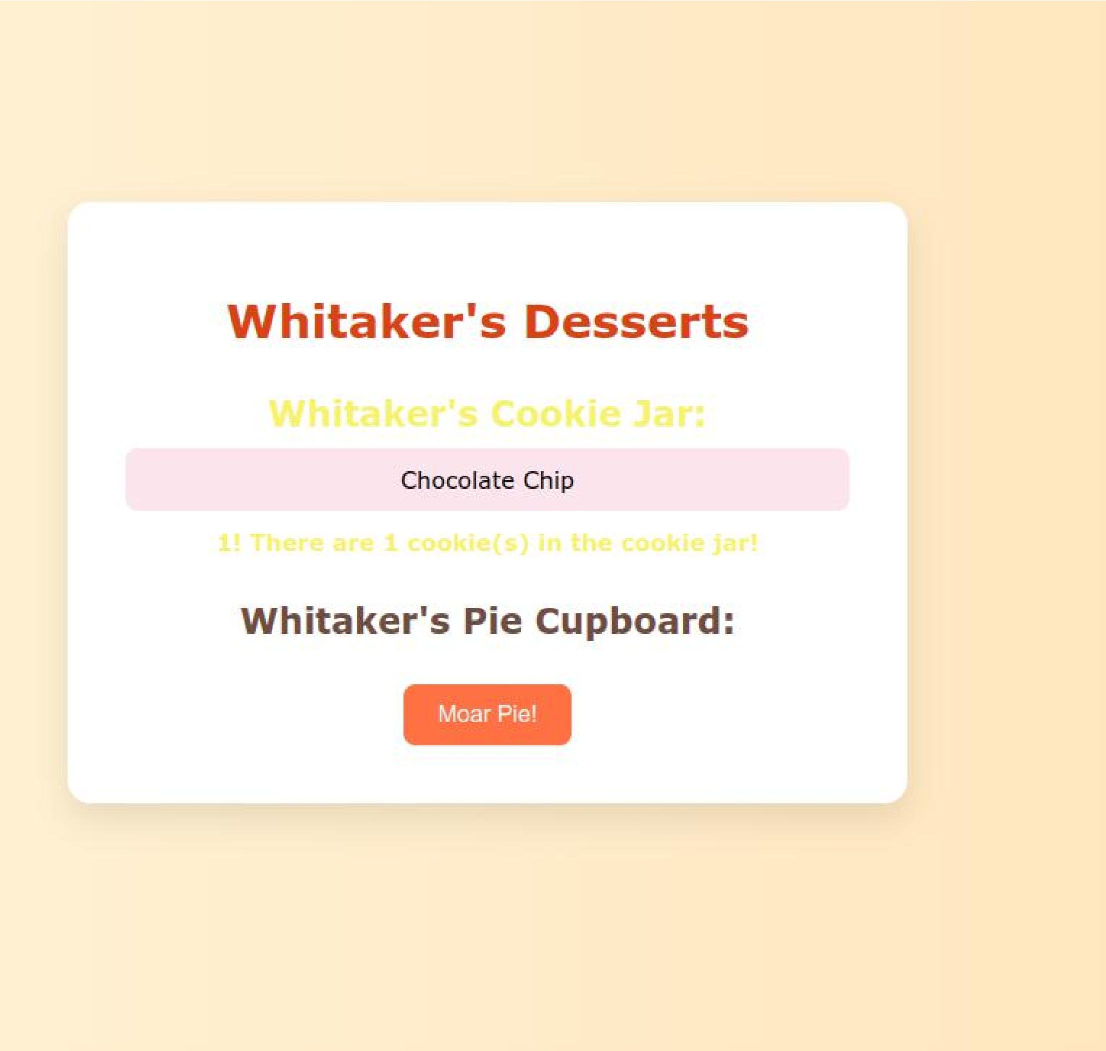
Whitaker's Cookie Jar:

Chocolate Chip

Oatmeal Raisin

2! There are 2 cookie(s) in the cookie jar!

Whitaker's Pie Cupboard:



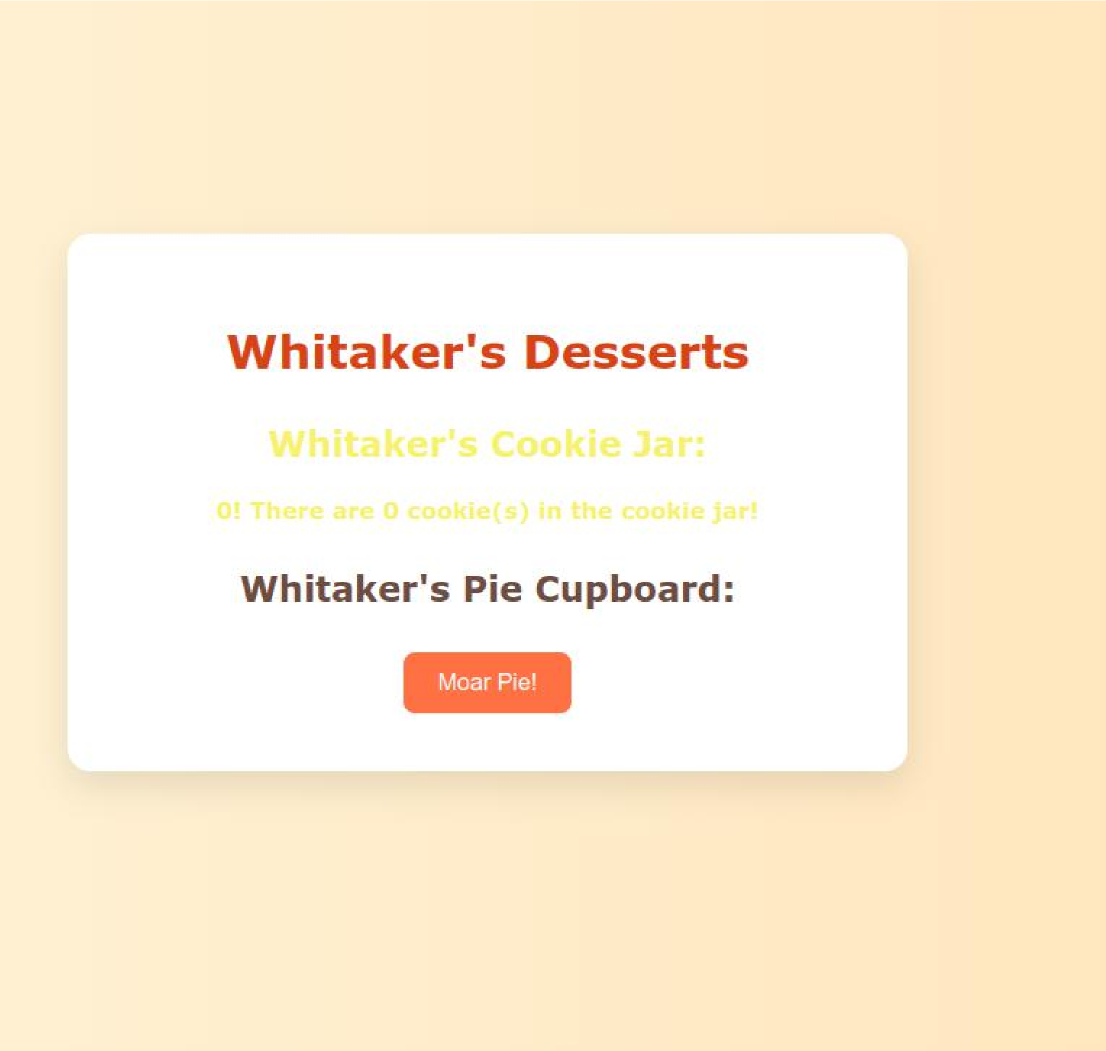
Whitaker's Desserts

Whitaker's Cookie Jar:

Chocolate Chip

1! There are 1 cookie(s) in the cookie jar!

Whitaker's Pie Cupboard:



Whitaker's Desserts

Whitaker's Cookie Jar:

0! There are 0 cookie(s) in the cookie jar!

Whitaker's Pie Cupboard: