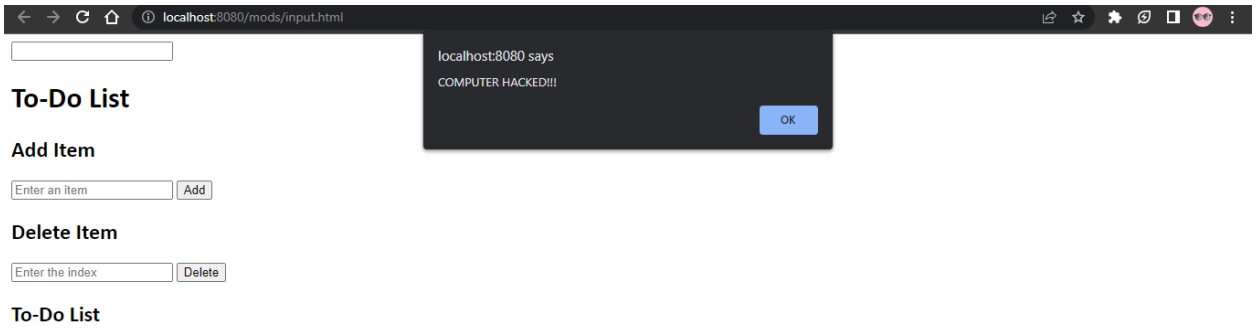
Code with malicious script injection:

```
<!DOCTYPE html>
<html>
<input type="text" autofocus onfocus="alert('COMPUTER HACKED!!!')"/>
<head>
  <title>To-Do List</title>
  <style>
    .completed {
      text-decoration: line-through;
      color: red;
    }
  </style>
</head>
<body>
  <h1>To-Do List</h1>
  <h2>Add Item</h2>
  <input type="text" id="itemInput" placeholder="Enter an item">
  <button onclick="addItem()">Add</button>
  <h2>Delete Item</h2>
  <input type="number" id="indexInput" placeholder="Enter the index">
  <button onclick="deleteItem()">Delete</button>
  <h2>To-Do List</h2>
  <ul id="todoList"></ul>
  <script>
    var items = [];
    function addItem() {
      var itemInput = document.getElementById("itemInput");
      var item = itemInput.value;
      if (item.trim() !== "") {
        items.push({ text: item, completed: false });
        itemInput.value = "";
        displayList();
      }
    }
    function deleteItem() {
      var indexInput = document.getElementById("userInput");
      var index = parseInt(indexInput.value) - 1;
      if (index >= 0 && index < items.length) {
        items[index].completed = true;
        indexInput.value = "";
        displayList();
```

```
        }
    }
    function displayList() {
        var todoList = document.getElementById("todoList");
        todoList.innerHTML = "";
        if (items.length === 0) {
            var emptyMessage = document.createElement("li");
            emptyMessage.textContent = "Your To-Do list is empty! Go relax and have fun!!!";
            todoList.appendChild(emptyMessage);
        } else {
            for (var i = 0; i < items.length; i++) {
                var listItem = document.createElement("li");
                listItem.textContent = (i + 1) + ". ";
                var itemText = document.createElement("span");
                if (items[i].completed) {
                    itemText.classList.add("completed");
                    itemText.textContent = items[i].text + " (Completed)";
                } else {
                    itemText.textContent = items[i].text;
                }
                listItem.appendChild(itemText);
                todoList.appendChild(listItem);
            }
        }
    }
  </script>
</body>
</html>
```

Output:

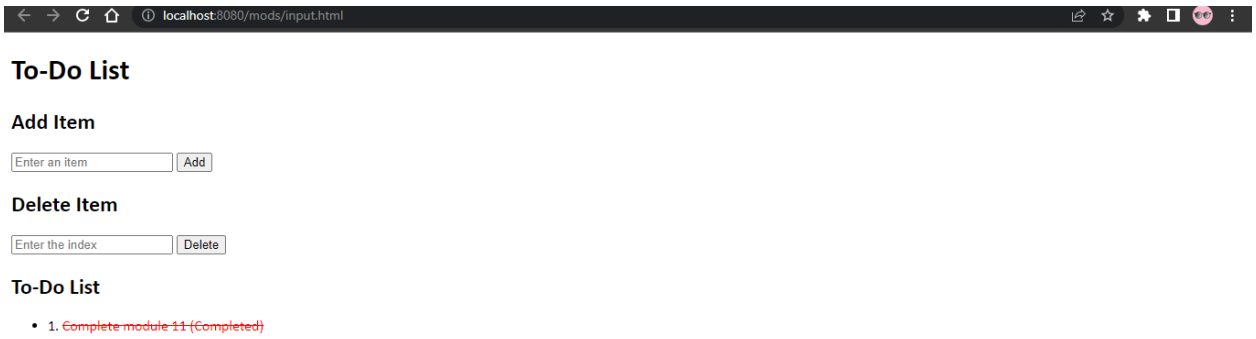


Code without malicious script injection:

```
<!DOCTYPE html>
<html>
<head>
  <title>To-Do List</title>
  <style>
    .completed {
      text-decoration: line-through;
      color: red;
    }
  </style>
</head>
<body>
  <h1>To-Do List</h1>
  <h2>Add Item</h2>
  <input type="text" id="itemInput" placeholder="Enter an item">
  <button onclick="addItem()">Add</button>
  <h2>Delete Item</h2>
  <input type="number" id="indexInput" placeholder="Enter the index">
  <button onclick="deleteItem()">Delete</button>
  <h2>To-Do List</h2>
  <ul id="todoList"></ul>
  <script>
    var items = [];
    function addItem() {
      var itemInput = document.getElementById("itemInput");
```

```
        var item = itemInput.value;
        if (item.trim() !== "") {
            items.push({ text: item, completed: false });
            itemInput.value = "";
            displayList();
        }
    }
    function deleteItem() {
        var indexInput = document.getElementById("indexInput");
        var index = parseInt(indexInput.value) - 1;
        if (index >= 0 && index < items.length) {
            items[index].completed = true;
            indexInput.value = "";
            displayList();
        }
    }
    function displayList() {
        var todoList = document.getElementById("todoList");
        todoList.innerHTML = "";
        if (items.length === 0) {
            var emptyMessage = document.createElement("li");
            emptyMessage.textContent = "Your To-Do list is empty! Go relax and have fun!!!";
            todoList.appendChild(emptyMessage);
        } else {
            for (var i = 0; i < items.length; i++) {
                var listItem = document.createElement("li");
                listItem.textContent = (i + 1) + ". ";
                var itemText = document.createElement("span");
                if (items[i].completed) {
                    itemText.classList.add("completed");
                    itemText.textContent = items[i].text + " (Completed)";
                } else {
                    itemText.textContent = items[i].text;
                }
                listItem.appendChild(itemText);
                todoList.appendChild(listItem);
            }
        }
    }
  </script>
</body>
</html>
```

Output:



On line 3 of the HTML code, I inserted a malicious script injection using an on-focus event. This propagation of vulnerability is also known as cross-site scripting (XSS). In this instance, I am specifically targeting a stored XSS vulnerability, which occurs when an application receives data from an untrusted source and includes it in a hazardous fashion in successive HTTP responses. XSS will naturally be detected and the inserted script will be executed. Because I placed the injection at the beginning of the code, the user cannot interact with the website. A hacker may misuse XSS by executing malicious scripts on another user's browser. Instead of directly attacking the target, the perpetrator exploits a vulnerability in a website that the victim visits and persuades it to disseminate the malicious script. To circumvent this, no autofocus attributes should be added to markup elements. In addition, the application must validate all input data, ensuring that only the permitted data is accepted, and that all variable output on a page is encoded prior to being presented to the user.