**Exercise 1: Sum of Array Elements**

**Task:** Calculate the sum of all elements in an array.

**Exercise 2: Find the Maximum Number in an Array**

**Task:** Find the largest number in an array.

**Exercise 3: Remove Duplicates from an Array**

**Task:** Remove duplicate values from an array.

**Exercise 4: Reverse an Array**

**Task:** Reverse the elements of an array.

**Exercise 5: Find All Even Numbers in an Array**

**Task:** Find all even numbers in an array.

**Exercise 6: Merge Two Arrays**

**Task:** Merge two arrays into one array without duplicates.

**Exercise 7: Find the Index of an Element**

**Task:** Find the index of a specific element in an array.

**Exercise 8: Sum of Array Elements**

**Task:** Write a function that takes an array of numbers and returns the sum of all the elements.

**Exercise 9: Find the Maximum Number in an Array**

**Task:** Write a function that returns the largest number from an array of numbers.

**Exercise 10: Remove Duplicates from an Array**

**Task:** Write a function that removes duplicate values from an array and returns a new array with unique values.

**Exercise 11: Reverse an Array**

**Task:** Write a function that reverses the elements of an array.

**Exercise 12: Find All Even Numbers in an Array**

**Task:** Write a function that returns an array containing only the even numbers from the input array.

**Exercise 13: Merge Two Arrays**

**Task:** Write a function that takes two arrays and merges them into one array without duplicates.

**Exercise 14: Find the Index of an Element**

**Task:** Write a function that returns the index of a specific element in an array. If the element is not found, return -1.

**Exercise 15: Dynamic List Creation**

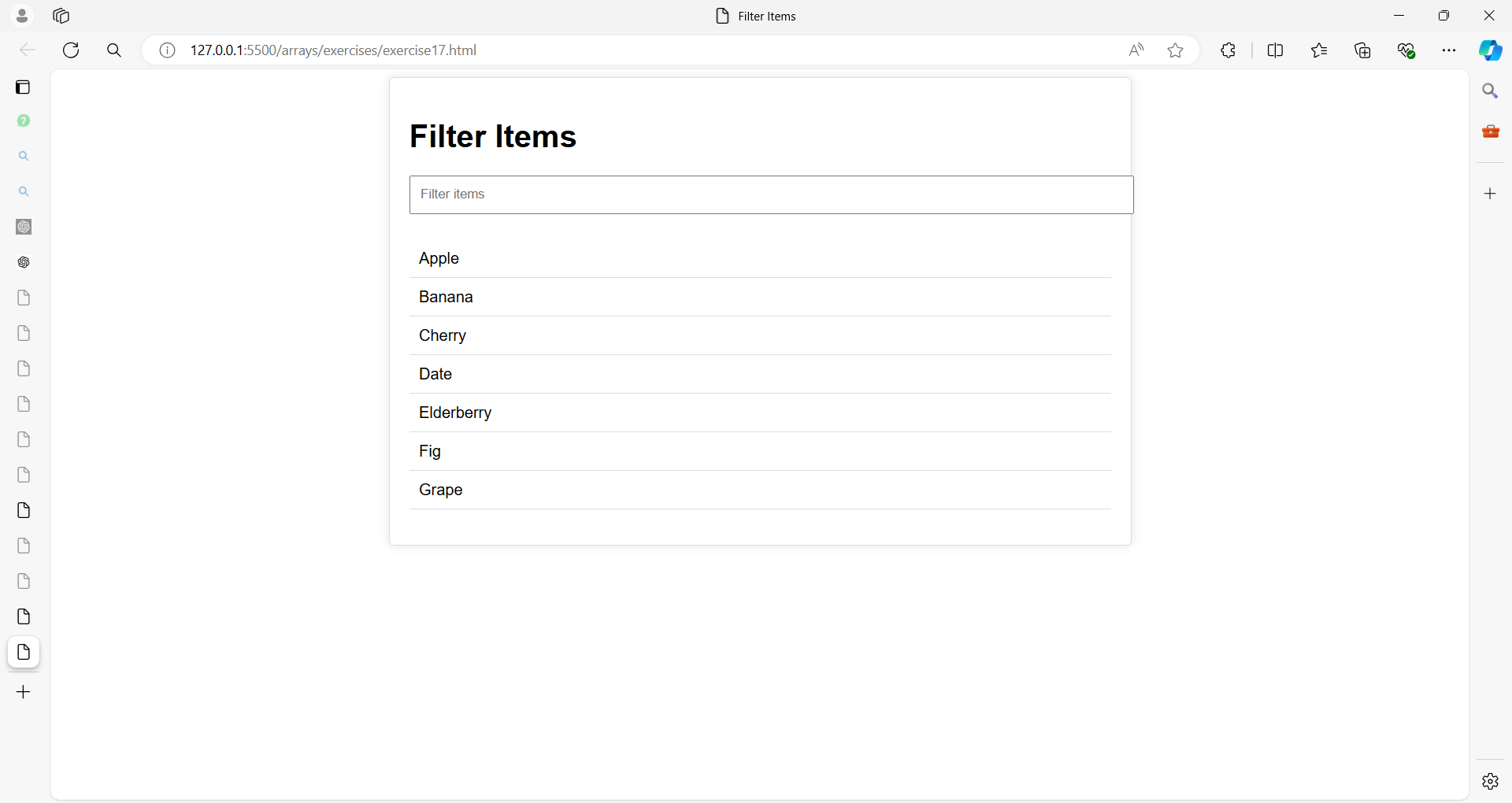
**Task:** Create a form that allows users to add items to a list. Each item should be displayed in a styled list below the form.

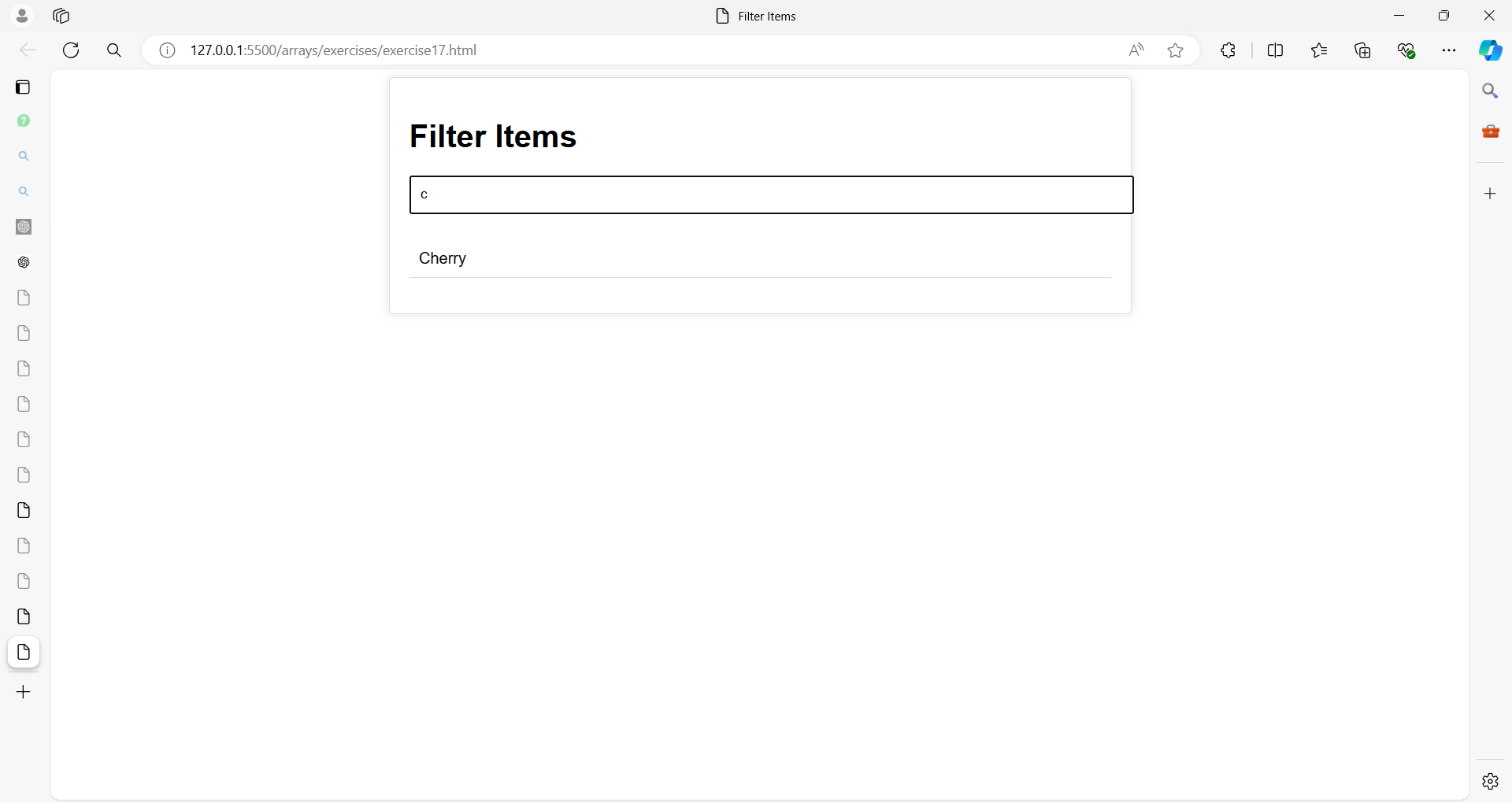
**Exercise 16: Form Validation with Array Data**

**Task:** Create a form to input user details (name and email). Validate the form to ensure no duplicate names are entered. Display the list of unique names in a styled list.

**Exercise 17: Filter and Display Array Data Based on Form Input**

**Task:** Create a form with an input field for filtering items. The list of items should be filtered based on the input and displayed in a styled list.





**Exercise 18: Product Management Application**

**Objective:** Build a Product Management Application where users can:

* Add products with various attributes (name, category, price, quantity).
* Edit existing products.
* Delete products.
* Filter products by name or category.
* Validate inputs before adding or updating products.

**Requirements:**

1. **Form Elements:**
   * Product Name
   * Category (Dropdown)
   * Price
   * Quantity
2. **Buttons:**
   * Add Product
   * Edit Product
   * Delete Product
3. **Product List Display:**
   * Display the list of products in a table.
4. **Filter:**
   * Filter products by name or category using an input field.
5. **Validation:**
   * Ensure that all fields are filled in correctly before adding or editing a product.
6. **CSS Styling:**
   * Apply modern styling for a clean and user-friendly interface.

Here are the values for the products:

iPhone 14, Electronics, 999, 50

Nike Air Max, Clothing, 120, 100

Samsung Galaxy S22, Electronics, 899, 75

Levi's 501 Jeans, Clothing, 89, 200

Apple MacBook Pro, Electronics, 1299, 30

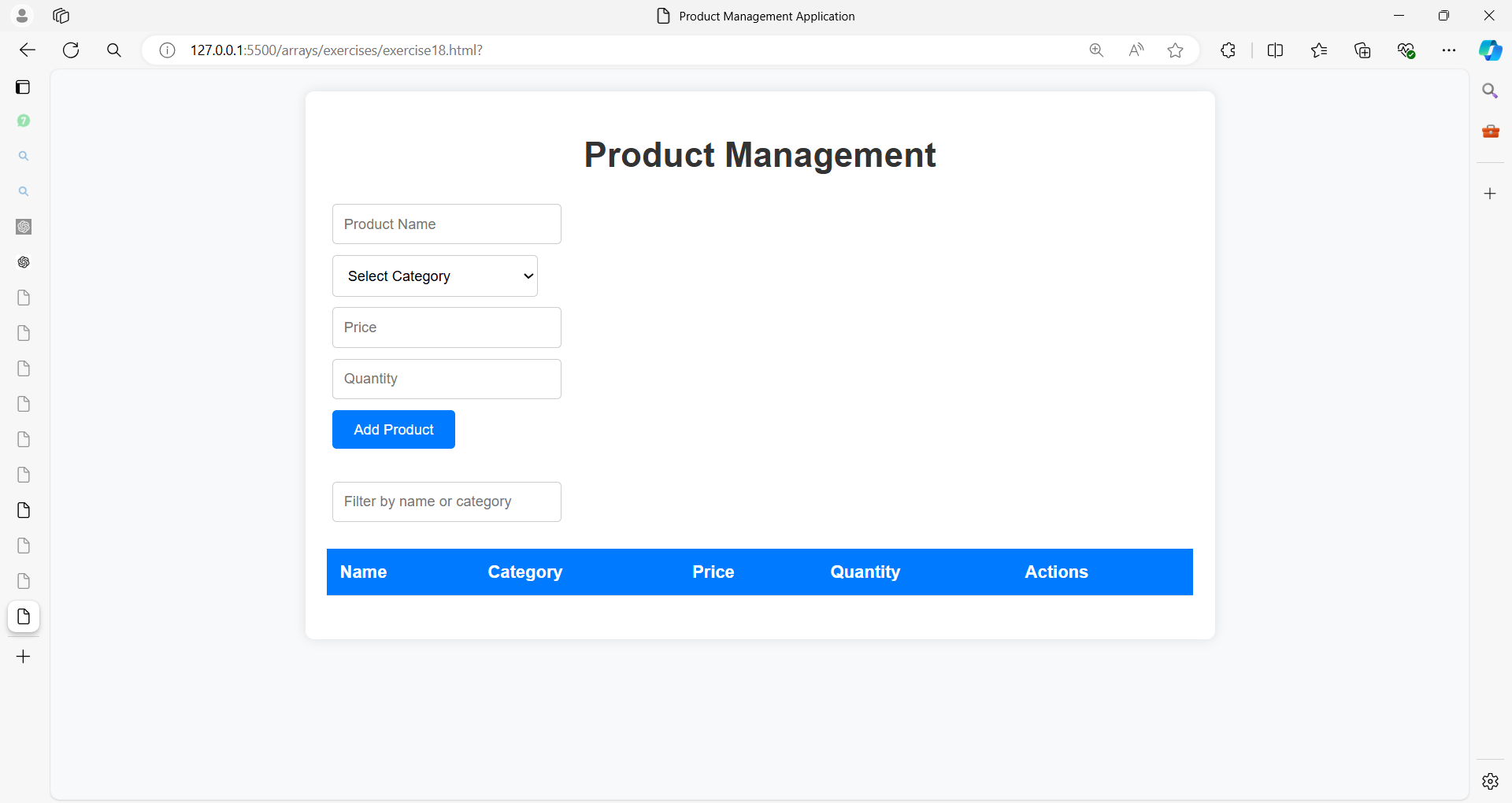
Organic Almonds (1 lb), Groceries, 15, 150

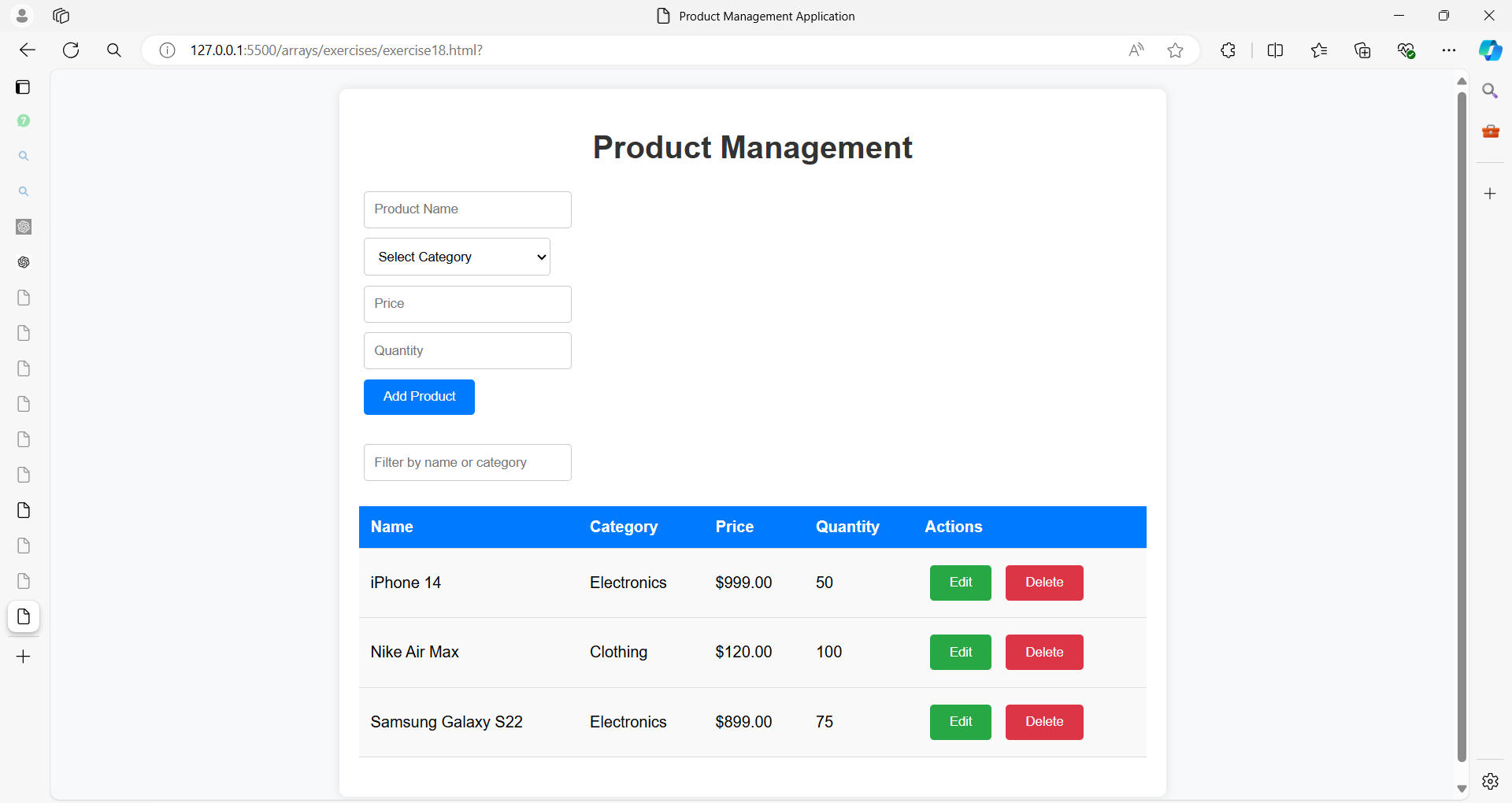
Sony WH-1000XM4 Headphones, Electronics, 349, 40

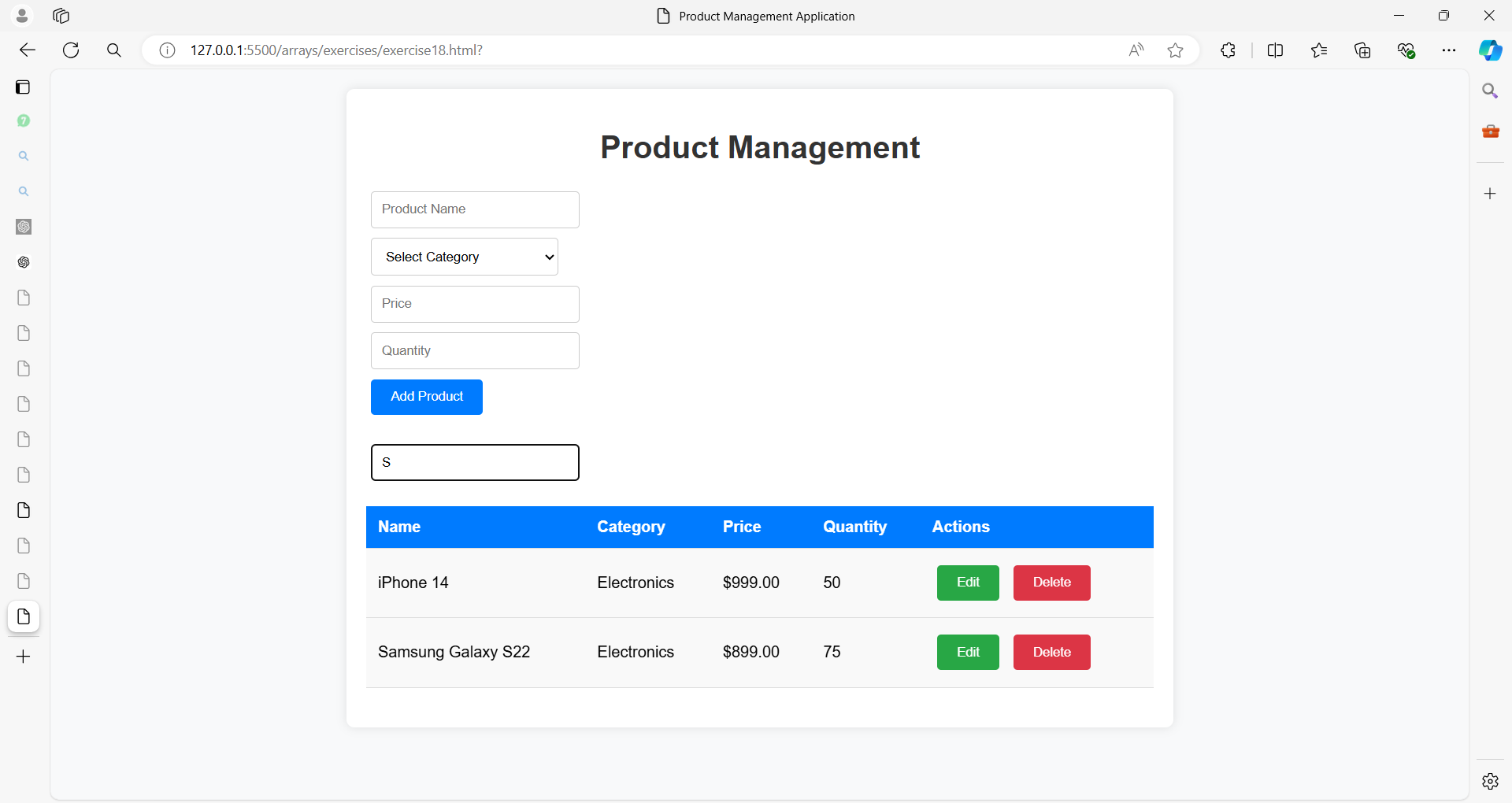
Patagonia Down Jacket, Clothing, 279, 60

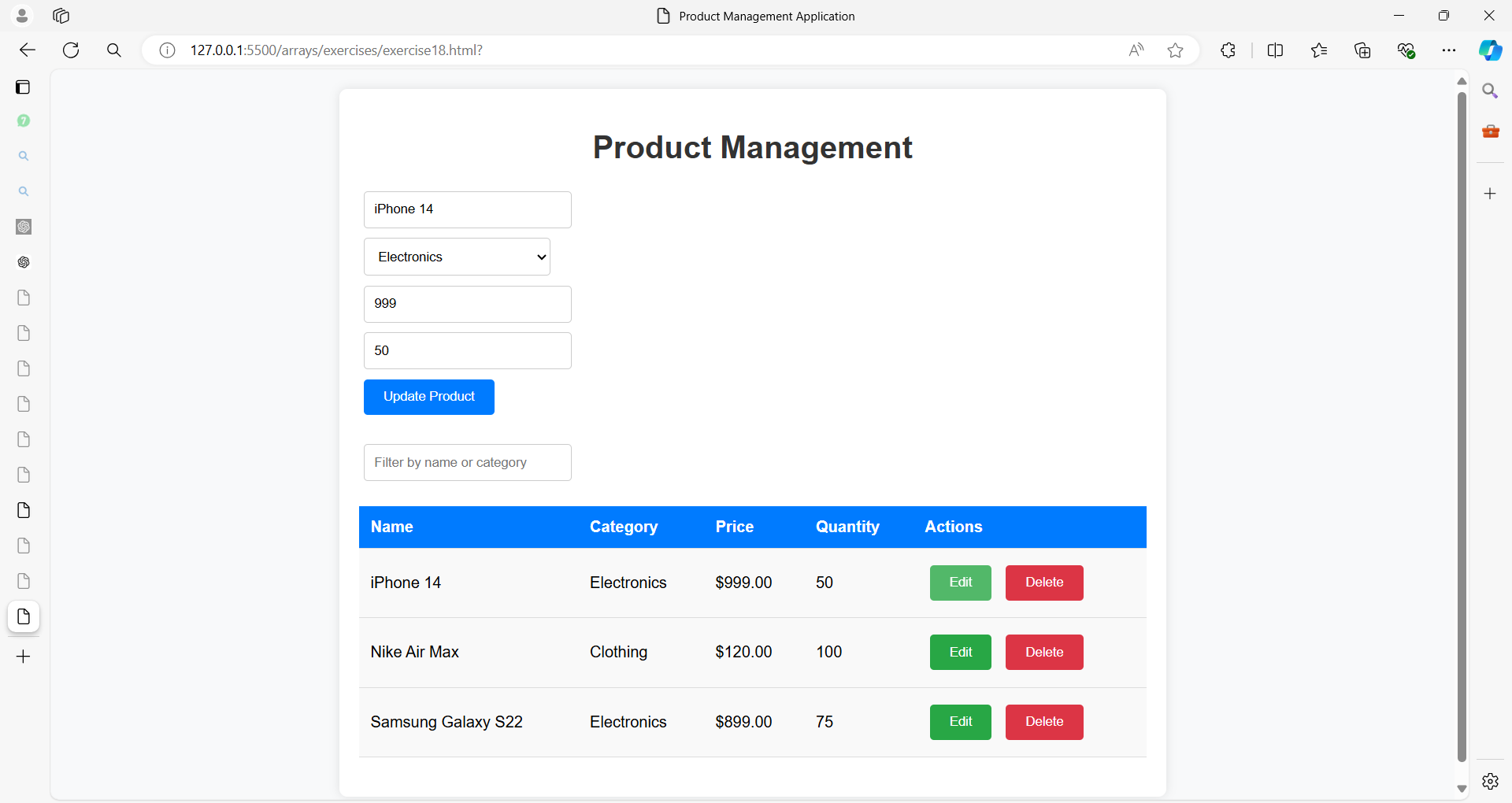
Organic Quinoa (2 lb), Groceries, 10, 250

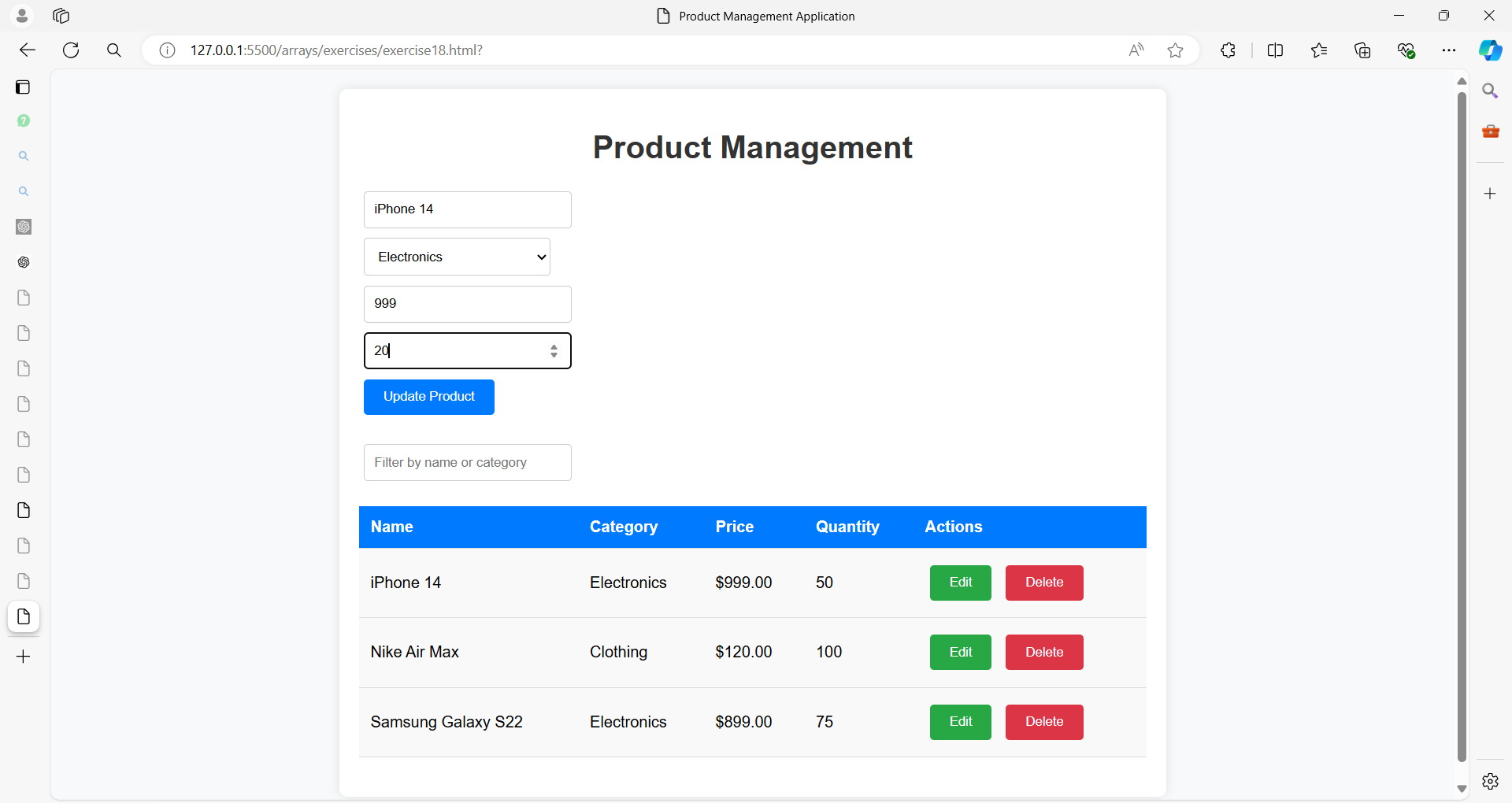
ASUS ROG Gaming Laptop, Electronics, 1499, 25

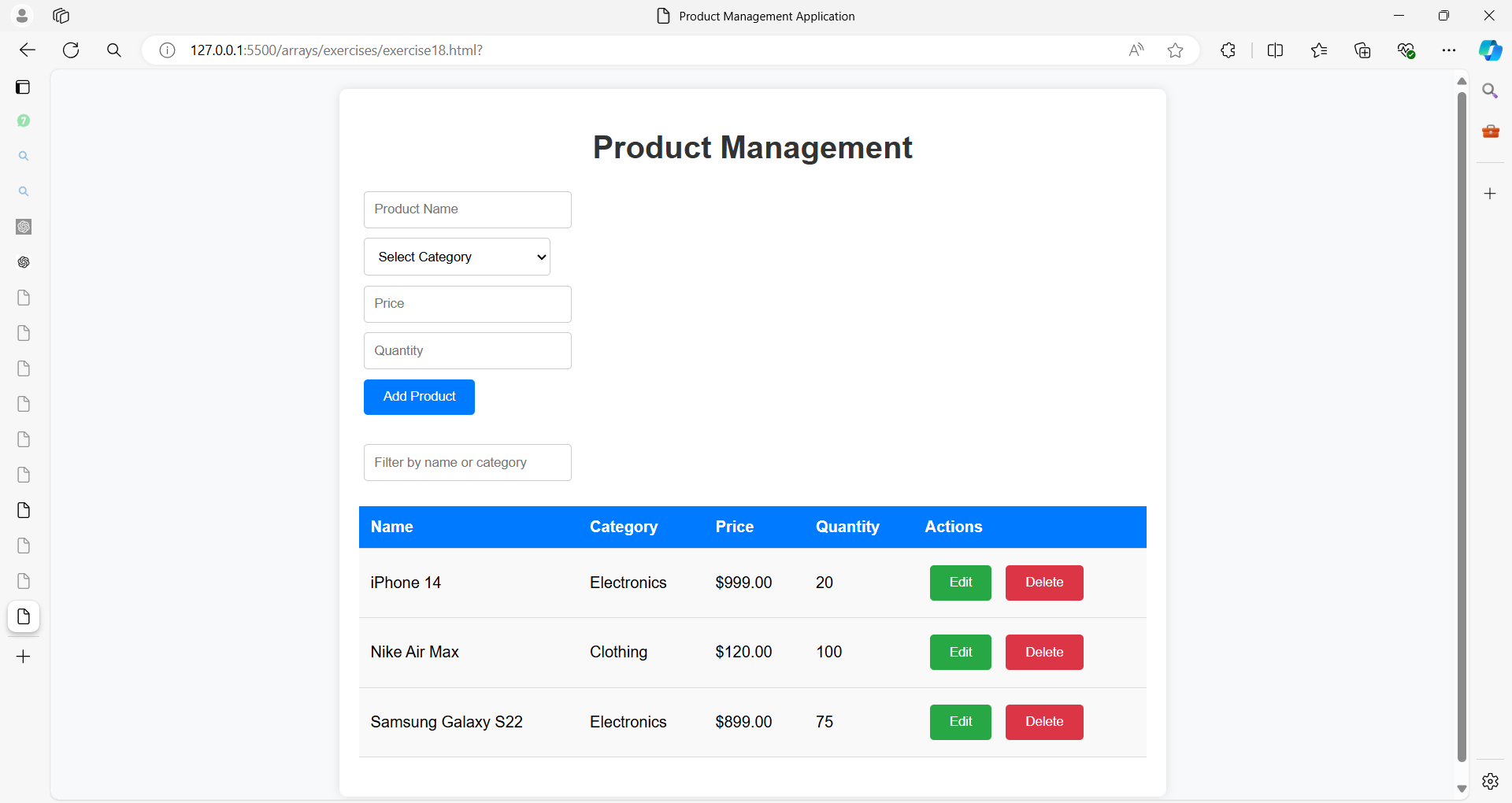


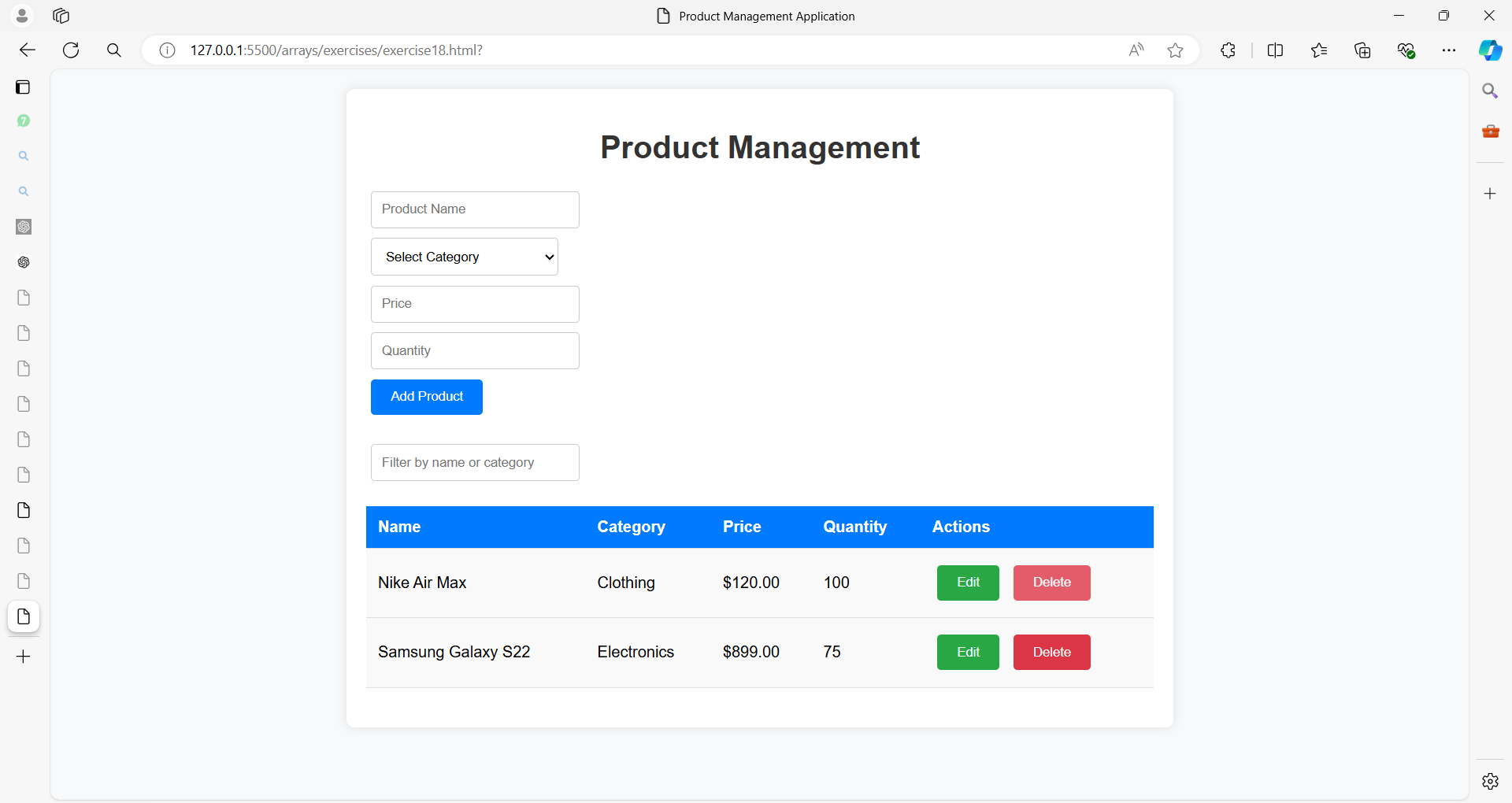












**Exercise 19: Library Management System**

**Objective:**  
Build a Library Management System using HTML, CSS3, and JavaScript. The system will allow users to manage a collection of books, including adding, viewing, editing, deleting, and filtering books.

**Exercise 20: Student Management System**

**Objective:**  
Build a Student Management System using HTML, CSS3, and JavaScript. The system will allow users to manage a list of students, including adding, viewing, editing, deleting, filtering, sorting by various fields, and calculating statistics like the average grade, highest grade, and lowest grade.

**Exercise 21: Task Management System**

**Objective:**  
Build a Task Management System using HTML, CSS3, and JavaScript. The system will allow users to manage tasks by adding, viewing, editing, deleting, filtering, sorting, and marking tasks as completed. Users will also be able to categorize tasks, prioritize them, and set due dates.

**Exercise 22: Project Management System**

**Objective:**

Create a Project Management System using HTML, CSS3, and JavaScript. The system will allow users to manage a list of projects, including adding, editing, deleting, filtering, sorting by deadline or priority, and marking projects as completed. Each project can have a title, description, deadline, priority, category, and completion status.