Problem Statement: 3 (JavaScript Objects/Array of Objects with JSON Data)

Design and develop a functional inventory management system for a Dell E-commerce application using a JavaScript array of objects populated from an external JSON file to manage a catalog of Dell products. The system should serve as a foundational backend component for a website, ensuring product information is accurate and up-to-date.

Core Information Requirements

The system must handle the following information for each product:

- **Product ID: A unique identifier for the product (e.g., a number or string).**
- **Product Name: The name of the product (e.g., 'Dell XPS 15').**
- **Inventory Status: A boolean value indicating whether a product is in stock (true) or not (false).**
- **Price: The price of the product.**
- **Category: A string representing the product category (e.g., 'Laptop', 'Monitor', 'Accessory').**

Functional Requirements

The inventory management system must perform the following actions:

- **Display Inventory: The user should be able to view a complete list of all products in the inventory, displaying all properties (ID, Name, Status, Price, and Category) for each product. The inventory should be displayed by default when the application loads.**
- **Check Availability: The user can input a product ID or name, and the system will check its availability (in stock or out of stock) and report the current status. The availability message should appear temporarily and then disappear.**

Technical Specifications

- **Data Source: The product catalog will be stored in an external JSON file (products.json) containing at least 50 product objects.**
- **Data Structure: The core of the inventory system must be implemented using a JavaScript array of objects, which will be populated dynamically by reading the**

products.json file. Each object in the array will represent a Dell product with the properties specified above.

- User Interface: An interface built with HTML and CSS to allow user interaction.
- Programming Language: JavaScript for all application logic, including fetching and parsing the JSON data.

Testing Requirements

The following test cases must be performed to ensure the system functions correctly:

Display Inventory Test

- Test Case 1.1: Load the application with an initial list of products read from products.json.
    - Expected Result: All initial products and their properties are displayed correctly in a table.

Check Stock Test

- Test Case 2.1: Check for a product that is in the inventory.
    - Expected Result: A message confirming the product is in stock is displayed temporarily.
- Test Case 2.2: Check for a product that is not in the inventory.
    - Expected Result: A message indicating the product is not found or is out of stock is displayed temporarily.

Learning Objectives

Upon completion of this project, students will be able to:

- Core Concepts: Understand and apply the fundamental concepts of JavaScript arrays and objects, including creating, modifying, and iterating over an array of objects.
- JSON Handling: Learn to fetch and parse external JSON data into JavaScript objects.
- JavaScript Functions: Write and use JavaScript functions (non-arrow functions) to encapsulate logic and organize code.

- **Event Handling: Use event handling to make the web application interactive, responding to user actions like button clicks and form submissions.**
- **CRUD Operations (Read): Implement basic Read operations on an array of objects data structure.**
- **HTML/CSS/JS Integration: Integrate HTML for structure, CSS for styling, and JavaScript for dynamic functionality to create a working web application.**
- **Problem-Solving: Break down a larger problem (inventory management) into smaller, manageable tasks and implement solutions for each.**
- **Logic and Conditionals: Utilize conditional statements (if/else) to handle different scenarios, such as checking for existing items or handling invalid inputs.**