

BÁO CÁO MINI PROJECT

Môn học: Lập trình JavaScript cơ bản

Tên dự án: Quản Lý Kho Hàng với Phương Thức Mảng (Inventory Management System)

Sinh viên thực hiện: Vũ Mạnh Hùng

Lớp: CNTT3

MSV: B25DTCN266

Giảng viên hướng dẫn: Lương Quốc Tuân

Ngày thực hiện: 25/ 2/ 2026

1. Giới thiệu dự án

1.1 Mục đích

Dự án *Inventory Management System* được xây dựng nhằm rèn luyện kỹ năng lập trình JavaScript, đặc biệt là vận dụng **Higher-Order Functions (HOF)** như `filter()`, `map()`, `reduce()`, `some()`, `every()` thay thế vòng lặp truyền thống.

Mục tiêu chính:

- Hiểu cách thao tác dữ liệu bằng mảng song song (Parallel Arrays).
- Áp dụng tư duy lập trình hàm vào bài toán thực tế.
- Xây dựng chương trình menu tương tác với người dùng qua `prompt()` và `alert()`.

1.2 Phạm vi và giới hạn hệ thống

Phạm vi

- Ứng dụng chạy trên trình duyệt web.
- Ngôn ngữ sử dụng: JavaScript thuần (Vanilla JS).
- Dữ liệu được lưu bằng mảng song song: `names`, `prices`, `stocks`.

Giới hạn

- Không sử dụng Object hoặc Database.
- Giao diện chỉ sử dụng `prompt()` và `alert()`.
- Chạy trong môi trường trình duyệt và triển khai trên GitHub Pages.

2. Mô tả hệ thống

Hệ thống hoạt động theo mô hình **Menu Loop**, liên tục hiển thị menu cho đến khi người dùng chọn chức năng *Thoát*.

2.1 Các chức năng chính

1. Lọc sản phẩm cao cấp

- **Đầu vào:** Mảng `names`, `prices`

- **Xử lý:** Dùng `filter()` để tìm sản phẩm có giá > 500
- **Đầu ra:** Danh sách tên sản phẩm cao cấp

2. Kiểm định trạng thái dữ liệu

- **Kiểm tra hết hàng:** dùng `some()`
- **Kiểm tra giá sàn:** dùng `every()` để đảm bảo tất cả giá > 100
- **Kết quả:** Trả về true/false và hiển thị trạng thái

3. Phân tích giá trị vốn hóa kho

- **Công thức:** Tổng tài sản = Σ (price \times stock)
- **Phương thức:** `reduce()`
- **Kết quả:** Tổng giá trị tồn kho

4. Chiến dịch giảm giá

- **Giảm 10% cho tất cả sản phẩm**
- **Phương thức:** `map()` để cập nhật giá mới
- **Yêu cầu:** Ghi đè trực tiếp vào mảng `prices`

5. Truy vấn sản phẩm theo từ khóa

- **Đầu vào:** Từ khóa tìm kiếm
- **Xử lý:** `filter()` + `toLowerCase()`
- **Kết quả:** Hiển thị Tên - Giá - Số lượng

6. Báo cáo tình trạng tồn kho

- **Xử lý:** `map()` tạo danh sách trạng thái “Còn hàng/Hết hàng”
- **Đầu ra:** Báo cáo song song với danh sách tên

7. Thoát chương trình

- Kết thúc vòng lặp `while` và hiển thị thông báo đóng ứng dụng.

Link github:https://github.com/tunav602-pixel/B25DTCN266_CNTT3_NguyenTuanViet_SS09.git

3. Yêu cầu hệ thống

3.1 Công nghệ sử dụng

- Ngôn ngữ: JavaScript ES6+
- Các phương thức mảng bắt buộc:

- o filter()

- o map()

- o reduce()

- o some()

- o every()

- o forEach()

3.2 Cấu trúc chương trình

- while(true) để hiển thị menu
- switch-case để điều hướng chức năng

4. Thiết kế giao diện (UI)

Menu chính

1. Lọc sản phẩm cao cấp (>500)

2. Kiểm định trạng thái dữ liệu

3. Phân tích giá trị vốn hóa
4. Giảm giá toàn bộ sản phẩm 10%
5. Tìm kiếm sản phẩm theo từ khóa
6. Báo cáo tình trạng tồn kho
7. Thoát chương trình

Giao diện sử dụng hộp thoại `prompt()` để nhập lựa chọn và `alert()` để hiển thị kết quả.

5. Triển khai hệ thống

5.1 Môi trường triển khai

- GitHub Pages

5.2 Quy trình triển khai

1. Tạo repository trên GitHub
2. Đẩy mã nguồn lên repository
3. Bật GitHub Pages trong Settings

- Lấy URL sản phẩm để nộp bài

6. Tiêu chí đánh giá

6.1 Tính đúng đắn thuật toán

- Sử dụng đúng mục đích từng phương thức HOF
- Đảm bảo đồng bộ chỉ số giữa các mảng song song

6.2 Tiêu chí chấm điểm

Tiêu chí	Tỷ lệ
Logic thuật toán	50%
Trình bày mã nguồn	30%
Triển khai sản phẩm	20%

7. Kết luận

Dự án *Inventory Management System* giúp sinh viên hiểu rõ cách xử lý dữ liệu bằng phương pháp lập trình hàm trong JavaScript. Việc sử dụng mảng song song và Higher-Order Functions giúp mã nguồn ngắn gọn, dễ đọc và tối ưu hơn so với vòng lặp truyền thống.

Qua dự án, sinh viên đã nâng cao kỹ năng lập trình, tư duy thuật toán và quy trình triển khai sản phẩm lên môi trường thực tế.

Tài liệu tham khảo

- JavaScript MDN Documentation
- Giáo trình JavaScript cơ bản
- Tài liệu giảng dạy của giảng viên Lương Quốc Quân