

BỘ GIÁO DỤC VÀ ĐÀO TẠO
TRƯỜNG ĐẠI HỌC SƯ PHẠM KỸ THUẬT HƯNG YÊN



ĐỒ ÁN 2
THIẾT KẾ WEBSITE QUẢN LÝ QUÁN CÀ PHÊ

CHUYÊN NGÀNH: KỸ THUẬT PHẦN MỀM

SINH VIÊN: Cao Duy Mạnh

MÃ SINH VIÊN: 12523099

MÃ LỚP: 12523T.1

HƯỚNG DẪN: Th.S ĐỖ THỊ THU TRANG

HƯNG YÊN - 2025

NHẬN XÉT

Nhận xét của giảng viên hướng dẫn:

[illegible]

GIẢNG VIÊN HƯỚNG DẪN

(Ký và ghi rõ họ tên)

LỜI CAM ĐOAN

Em xin cam đoan đồ án “Thiết kế website quản lý quán cà phê” là kết quả thực hiện của bản thân em dưới sự hướng dẫn của cô Đỗ Thị Thu Trang.

Những phần sử dụng tài liệu tham khảo trong đồ án đã được nêu rõ trong phần tài liệu tham khảo. Các kết quả trình bày trong đồ án và chương trình xây dựng được hoàn toàn là kết quả do bản thân em thực hiện.

Nếu vi phạm lời cam đoan này, em xin chịu hoàn toàn trách nhiệm trước khoa và nhà trường.

Hung Yên, ngày 15 tháng 12 năm 2025

Sinh viên

Mạnh

Cao Duy Mạnh

LỜI CẢM ƠN

Để có thể hoàn thành đồ án này, lời đầu tiên em xin phép gửi lời cảm ơn tới bộ môn Công nghệ phần mềm, Khoa Công nghệ thông tin – Trường Đại học Sư phạm Kỹ thuật Hưng Yên đã tạo điều kiện thuận lợi cho em thực hiện đồ án môn học này.

Đặc biệt em xin chân thành cảm ơn cô Đỗ Thị Thu Trang đã rất tận tình hướng dẫn, chỉ bảo em trong suốt thời gian thực hiện đồ án vừa qua.

Em cũng xin chân thành cảm ơn tất cả các Thầy, các Cô trong Trường đã tận tình giảng dạy, trang bị cho em những kiến thức cần thiết, quý báu để giúp em thực hiện được đồ án này.

Mặc dù em đã có cố gắng, nhưng với trình độ còn hạn chế, trong quá trình thực hiện đề tài không tránh khỏi những thiếu sót. Em hi vọng sẽ nhận được những ý kiến nhận xét, góp ý của các Thầy giáo, Cô giáo về những kết quả triển khai trong đồ án.

Em xin trân trọng cảm ơn!

MỤC LỤC

NHẬN XÉT	2
LỜI CAM ĐOAN.....	3
LỜI CẢM ƠN	4
MỤC LỤC	5
DANH MỤC CÁC THUẬT NGỮ	7
DANH MỤC CÁC BẢNG	8
DANH MỤC CÁC HÌNH VẼ, ĐỒ THỊ	9
CHƯƠNG 1. . TỔNG QUAN VỀ ĐỀ TÀI	11
1.1. Lý do chọn đề tài	11
1.2. Mục tiêu của đề tài.....	11
1.2.1. Mục tiêu tổng quát.....	11
1.2.2. Mục tiêu cụ thể	11
1.3. Giới hạn và phạm vi của đề tài	12
1.3.1. Đối tượng nghiên cứu	12
1.3.2. Phạm vi nghiên cứu	12
1.4. Nội dung thực hiện.....	13
1.5. Phương pháp tiếp cận.....	14
CHƯƠNG 2. . CƠ SỞ LÝ THUYẾT	16
2.1. Quy trình phát triển phần mềm.....	16
2.1.1. Waterfall model – Mô hình thác nước.....	18
2.1.2. V - Shaped Model – Mô hình chữ V	19
2.1.3. Spiral Model – Mô hình xoắn ốc	20
2.2. Thiết kế giao diện web với HTML, CSS, JS.....	21
2.2.1. HTML (Hypertext Markup Language).....	21
2.2.2. CSS (Cascading Style Sheet Language).....	25
2.2.3. JavaScript	33
2.3. Lập trình phía font-end.....	38
CHƯƠNG 3. . PHÂN TÍCH VÀ THIẾT KẾ HỆ THỐNG	44
3.1. Phát biểu bài toán.....	44

3.2. Đặc tả yêu cầu phần mềm	44
3.2.1. Các yêu cầu chức năng	44
3.2.2. Biểu đồ lớp thực thể.....	51
3.2.3. Biểu đồ phân tích ca sử dụng	55
3.2.4. Các yêu cầu phi chức năng	61
3.3. Thiết kế giao diện	63
3.3.1. Thiết kế giao diện cho phân hệ quản trị và nhân viên	63
CHƯƠNG 4. . TRIỂN KHAI WEBSITE	68
4.1. Triển khai các chức năng cho phân hệ quản trị nội dung	68
4.1.1. Trang danh sách bàn	68
4.1.2. Trang đăng nhập	74
4.1.3. Trang thống kê doanh thu	76
4.2. Kiểm thử và triển khai ứng dụng	81
4.2.1. Kiểm thử.....	81
4.2.2. Đóng gói ứng dụng	86
4.2.3. Triển khai ứng dụng.....	88
KẾT LUẬN	91
TÀI LIỆU THAM KHẢO	92

DANH MỤC CÁC THUẬT NGỮ

STT	Từ viết tắt	Cụm từ tiếng anh	Diễn giải
1	CSDL	Database	Cơ sở dữ liệu
2	SQL	Structured Query Language	Ngôn ngữ truy vấn có cấu trúc
3	UC	Use case	Cách người dùng tương tác với hệ thống
4	HTML	HyperText Markup Language	Ngôn ngữ đánh dấu siêu văn bản
5	CSS	Cascading Style Sheets	Ngôn ngữ định dạng và tạo kiểu cho trang web
6	JavaScript	JavaScript	Ngôn ngữ lập trình phổ biến nhất cho web, tạo tính tương tác, động trên trang

DANH MỤC CÁC BẢNG

BẢNG 2. 1: CÁC THẺ CƠ BẢN TRONG HTML	23
BẢNG 2. 2: THUỘC TÍNH VỀ VĂN BẢN VÀ CHỮ	28
BẢNG 2. 3: THUỘC TÍNH VỀ MÀU SẮC VÀ NỀN.....	29
BẢNG 2. 4: THUỘC TÍNH VỀ KÍCH THƯỚC VÀ BỐ CỤC	29
BẢNG 2. 5: THUỘC TÍNH VỀ HIỂN THỊ VÀ VỊ TRÍ.....	30
BẢNG 2. 6: THUỘC TÍNH VỀ DANH SÁCH VÀ BẢNG	31
BẢNG 2. 7: CSS NÂNG CAO (HIỆU ỨNG & ANIMATION)	31
BẢNG 2. 8: CÁC KHÁI NIỆM VÀ ĐẶC ĐIỂM CHÍNH CỦA JAVASCRIPT.....	35
BẢNG 4. 1: BẢNG TEST CASE CHỨC NĂNG ĐĂNG NHẬP	81
BẢNG 4. 2: BẢNG TEST CASE CHỨC NĂNG QUẢN LÝ BÀN.....	82
BẢNG 4. 3: BẢNG TEST CASE CHỨC NĂNG QUẢN LÝ DANH MỤC VÀ MÓN	82
BẢNG 4. 4: BẢNG TEST CASE CHỨC NĂNG TẠO HÓA ĐƠN BÁN	83
BẢNG 4. 5: BẢNG TEST CASE CHỨC NĂNG TRUY XUẤT HÓA ĐƠN	84
BẢNG 4. 6: BẢNG TEST CASE CHỨC NĂNG QUẢN LÝ DOANH THU	84
BẢNG 4. 7: BẢNG TEST CASE CHỨC NĂNG THAY ĐỔI MẬT KHẨU.....	84
BẢNG 4. 8: BẢNG TEST CASE CHỨC NĂNG SAO LƯU DỮ LIỆU	85
BẢNG 4. 9: BẢNG TEST CASE CHỨC NĂNG KHÔI PHỤC DỮ LIỆU	85
BẢNG 4. 10: BẢNG ĐIỀU KIỆN PHẦN CỨNG.....	88
BẢNG 4. 11: BẢNG PHẦN MỀM VÀ CÔNG CỤ PHÁT TRIỂN.....	89

DANH MỤC CÁC HÌNH VẼ, ĐỒ THỊ

HÌNH 2. 1: MÔ HÌNH THÁC NƯỚC	18
HÌNH 2. 2: MÔ HÌNH CHỮ V	19
HÌNH 2. 3: MÔ HÌNH XOẢN ỐC	20
HÌNH 2. 4: CẤU TRÚC CỦA HTML	23
HÌNH 2. 5: VÍ DỤ VỀ CSS	25
HÌNH 2. 6: BỐ CỤC CỦA CSS.....	26
HÌNH 2. 7: CẤU TRÚC CỦA CSS	27
HÌNH 2. 8: SƠ LƯỢC VỀ BỘ CHỌN.....	28
HÌNH 2. 9: CSS INLINE	32
HÌNH 2. 10: CSS INTERNAL.....	33
HÌNH 2. 11: CSS EXTERNAL.....	33
HÌNH 2. 12: VÍ DỤ MINH HỌA.....	36
HÌNH 2. 13: SỰ KẾT HỢP CỦA HTML - CSS - JS.....	38
HÌNH 3. 1: USECASE TỔNG QUÁT HỆ THỐNG QLCF.....	45
HÌNH 3. 2: BIỂU ĐỒ LỚP THỰC THỂ	51
HÌNH 3. 3: BIỂU ĐỒ VOPC CHO CA SỬ DỤNG ĐĂNG NHẬP	55
HÌNH 3. 4: BIỂU ĐỒ VOPC CHO CA SỬ DỤNG QUẢN LÝ DANH MỤC	56
HÌNH 3. 5: BIỂU ĐỒ VOPC CHO CA SỬ DỤNG QUẢN LÝ MÓN.....	57
HÌNH 3. 6: BIỂU ĐỒ VOPC CHO CA SỬ DỤNG QUẢN LÝ BÀN	57
HÌNH 3. 7: BIỂU ĐỒ VOPC CHO CA SỬ DỤNG THỐNG KÊ DOANH THU.....	58
HÌNH 3. 8: BIỂU ĐỒ VOPC CHO CA SỬ DỤNG TẠO HÓA ĐƠN BÁN.....	59
HÌNH 3. 9: BIỂU ĐỒ VOPC CHO CA SỬ DỤNG TRUY XUẤT HÓA ĐƠN.....	59
HÌNH 3. 10: BIỂU ĐỒ VOPC CHO CA SỬ DỤNG THAY ĐỔI MẬT KHẨU	60
HÌNH 3. 11: BIỂU ĐỒ VOPC CHO CA SỬ DỤNG SAO LƯU VÀ PHỤC HỒI DỮ LIỆU	61
HÌNH 3. 12: GIAO DIỆN TRANG ĐĂNG NHẬP.....	63
HÌNH 3. 13: GIAO DIỆN TRANG DANH SÁCH BÀN	64
HÌNH 3. 14: GIAO DIỆN TRANG QUẢN LÝ DANH MỤC	64
HÌNH 3. 15: GIAO DIỆN TRANG QUẢN LÝ MÓN	65

HÌNH 3. 16: GIAO DIỆN TRANG THỐNG KÊ DOANH THU	65
HÌNH 3. 17: GIAO DIỆN TRANG QUẢN LÝ BÀN.....	66
HÌNH 3. 18: GIAO DIỆN TRANG THAY ĐỔI MẬT KHẨU.....	66
HÌNH 3. 19: GIAO DIỆN TRANG TRUY XUẤT HÓA ĐƠN BÁN.....	67
HÌNH 3. 20: GIAO DIỆN TRANG SAO LƯU VÀ KHÔI PHỤC DỮ LIỆU	67
HÌNH 4. 1: VÍ DỤ MÃ NGUỒN MINH HỌA (TRANG DANH SÁCH BÀN - HTML)	68
HÌNH 4. 2: VÍ DỤ MÃ NGUỒN MINH HỌA (TRANG DANH SÁCH BÀN – CSS).....	69
HÌNH 4. 3: VÍ DỤ MÃ NGUỒN MINH HỌA (RESPONSIVE)	70
HÌNH 4. 4: VÍ DỤ MÃ NGUỒN MINH HỌA (TRANG DANH SÁCH BÀN – JS)	72
HÌNH 4. 5: VÍ DỤ MÃ NGUỒN MINH HỌA (TRANG ĐĂNG NHẬP – HTML).....	74
HÌNH 4. 6: VÍ DỤ MÃ NGUỒN MINH HỌA (TRANG ĐĂNG NHẬP – CSS)	75
HÌNH 4. 7: VÍ DỤ MÃ NGUỒN MINH HỌA (TRANG ĐĂNG NHẬP - JS)	75
HÌNH 4. 8: VÍ DỤ MÃ NGUỒN MINH HỌA (TRANG THỐNG KÊ DOANH THU - HTML)....	76
HÌNH 4. 9: VÍ DỤ MÃ NGUỒN MINH HỌA (TRANG THỐNG KÊ DOANH THU – CSS)	77
HÌNH 4. 10: VÍ DỤ MÃ NGUỒN (TRANG THỐNG KÊ DOANH THU – XỬ LÝ THỐNG KÊ) 79	
HÌNH 4. 11: VÍ DỤ MÃ NGUỒN MINH HỌA (BIỂU ĐỒ CỘT - JS)	80
HÌNH 4. 12: VÍ DỤ MÃ NGUỒN MINH HỌA (BIỂU ĐỒ TRÒN - JS)	81
HÌNH 4. 14: CẤU TRÚC THƯ MỤC SAU KHI ĐÓNG GÓI	87
HÌNH 4. 15: CẤU TRÚC FILE .ZIP	88

CHƯƠNG 1. TỔNG QUAN VỀ ĐỀ TÀI

1.1. Lý do chọn đề tài

+ Trong bối cảnh công nghệ thông tin phát triển mạnh mẽ, việc ứng dụng các hệ thống quản lý vào hoạt động kinh doanh đang trở thành xu hướng tất yếu. Đối với các quán cà phê, việc quản lý thủ công bằng sổ sách hoặc Excel thường gây ra nhiều khó khăn như:

- Dễ xảy ra sai sót khi tính tiền hoặc ghi nhận đơn hàng,
- Khó kiểm soát doanh thu, nguyên liệu, nhân viên,
- Mất nhiều thời gian tổng hợp và thống kê.

+ Do đó, nhóm em chọn đề tài **“Thiết kế website quản lý quán cà phê”** với mong muốn xây dựng một hệ thống giúp **tự động hóa quá trình quản lý quán, giảm sai sót, tăng hiệu quả phục vụ**, đồng thời giúp **người quản lý dễ dàng theo dõi tình hình kinh doanh** mọi lúc, mọi nơi thông qua trình duyệt web.

1.2. Mục tiêu của đề tài

1.2.1. Mục tiêu tổng quát

Xây dựng một **website quản lý quán cà phê** có khả năng **quản lý hoạt động bán hàng, menu, hóa đơn, nhân viên và doanh thu**, giúp chủ quán và nhân viên thao tác nhanh chóng, chính xác và hiệu quả.

1.2.2. Mục tiêu cụ thể

- + **Thiết kế giao diện thân thiện, dễ sử dụng** cho cả nhân viên và quản lý.
- + **Xây dựng cơ sở dữ liệu** tạo hóa đơn, quản lý thông tin món, quản lý danh mục, bàn, truy xuất hóa đơn, quản lý doanh thu, sao lưu và khôi phục dữ liệu.
- + **Phát triển các chức năng chính**, bao gồm:
 - Đăng nhập, phân quyền người dùng (quản lý, nhân viên).
 - Quản lý danh mục, bàn và món trong menu.
 - Quản lý đơn hàng, tính tiền, in hóa đơn.
 - Thống kê doanh thu theo ngày, tháng, món bán chạy.
 - Sao lưu và khôi phục dữ liệu

+ **Đảm bảo tính bảo mật và toàn vẹn dữ liệu.**

1.3. Giới hạn và phạm vi của đề tài

1.3.1. Đối tượng nghiên cứu

+ Các hoạt động nghiệp vụ trong quán cà phê: phục vụ khách, ghi nhận order, tính tiền, quản lý menu, nhân viên, và báo cáo doanh thu.

+ Các công nghệ web phục vụ xây dựng hệ thống quản lý (HTML, CSS, JavaScript, PHP/ASP.NET, cơ sở dữ liệu MySQL/SQL Server).

1.3.2. Phạm vi nghiên cứu

+ **Phạm vi của đề tài:**

- Đề tài tập trung vào **xây dựng website quản lý nội bộ quán cà phê** (chủ quán và nhân viên sử dụng).
- Không đi sâu vào các chức năng nâng cao như đặt hàng online, thanh toán trực tuyến, hay tích hợp qua app di động.
- Hệ thống được xây dựng và chạy thử trên môi trường **localhost** (XAMPP hoặc Visual Studio Server).

+ **Ý nghĩa khoa học của đề tài:**

- Đề tài giúp sinh viên vận dụng kiến thức đã học về thiết kế và phát triển website, đồng thời tìm hiểu cách tổ chức và quản lý dữ liệu sản phẩm, giỏ hàng trên nền tảng web tĩnh thông qua các công nghệ HTML, CSS, JavaScript và kỹ thuật mô phỏng dữ liệu (JSON, LocalStorage).

+ **Ý nghĩa thực tiễn của đề tài:**

- Website quản lý quán cà phê mang lại nhiều lợi ích rõ rệt cho hoạt động kinh doanh:
 - **Tự động hóa quá trình quản lý**, giúp giảm bớt công việc thủ công như ghi order, tính tiền, hoặc ghi chép doanh thu.
 - **Tăng tính chính xác và minh bạch** trong việc tính hóa đơn, thống kê doanh thu, tránh thất thoát dữ liệu.

- **Dễ dàng theo dõi hoạt động kinh doanh:** chủ quán có thể xem báo cáo doanh thu, món bán chạy, hoặc tình hình làm việc của nhân viên mọi lúc, mọi nơi.
- **Tiết kiệm thời gian và chi phí vận hành,** nhờ việc tổng hợp dữ liệu tự động và hạn chế sai sót của con người.
- **Tăng trải nghiệm khách hàng,** vì nhân viên phục vụ nhanh hơn, thao tác order và thanh toán thuận tiện hơn.

+ Đối với sinh viên

- Giúp sinh viên **hiểu rõ quy trình xây dựng một ứng dụng web thực tế**, từ khâu khảo sát, phân tích đến thiết kế và lập trình.
- Tạo cơ hội **vận dụng kiến thức đã học** về lập trình web, cơ sở dữ liệu, phân tích thiết kế hệ thống thông tin.
- Rèn luyện **tư duy phân tích nghiệp vụ, khả năng thiết kế giao diện và xử lý dữ liệu** thực tế.
- Là **cơ sở thực hành hữu ích** cho các dự án lớn hơn trong tương lai, như hệ thống bán hàng trực tuyến hoặc quản lý chuỗi cửa hàng.

1.4. Nội dung thực hiện

+ Khảo sát và phân tích yêu cầu

- Tìm hiểu quy trình hoạt động của quán cà phê.
- Xác định các tác nhân, nghiệp vụ, luồng dữ liệu và các chức năng cần thiết.

+ Thiết kế hệ thống

- Thiết kế sơ đồ Use Case, ERD, và cấu trúc cơ sở dữ liệu.
- Thiết kế giao diện website (HTML, CSS, Bootstrap).

+ Xây dựng và cài đặt hệ thống

- Xây dựng chức năng đăng nhập, quản lý món, quản lý nhân viên, lập hóa đơn, thống kê.
- Kết nối website với cơ sở dữ liệu.

+ Kiểm thử và đánh giá

- Kiểm tra các chức năng của website.

- Đánh giá tính ổn định, hiệu năng và tính thân thiện của giao diện.

+ **Kết luận và hướng phát triển**

- Đánh giá kết quả đạt được.
- Đề xuất hướng mở rộng như tích hợp bán hàng online, thanh toán điện tử, hoặc app di động.

1.5. Phương pháp tiếp cận

+ Để thực hiện đề tài “Thiết kế website quản lý quán cà phê”, em lựa chọn **phương pháp tiếp cận hướng hệ thống (System Approach)** kết hợp với **quy trình phát triển phần mềm mô hình thác nước (Waterfall Model)**. Phương pháp này giúp đảm bảo việc phân tích, thiết kế và xây dựng website được tiến hành **một cách có hệ thống, logic và dễ kiểm soát**.

+ **Phương pháp thu thập thông tin**

- **Khảo sát thực tế:** Tìm hiểu quy trình vận hành của một quán cà phê, bao gồm các bước: phục vụ khách, ghi nhận order, tính tiền, và tổng hợp doanh thu.
- **Phỏng vấn / tham khảo người dùng:** Hỏi ý kiến chủ quán hoặc nhân viên phục vụ để hiểu rõ yêu cầu và khó khăn trong quá trình quản lý.
- **Tham khảo tài liệu:** Nghiên cứu các website quản lý bán hàng, các tài liệu về phân tích và thiết kế hệ thống thông tin, cũng như các hướng dẫn kỹ thuật về lập trình web.

+ **Phương pháp phân tích và thiết kế hệ thống**

- **Phân tích nghiệp vụ:** Xác định các tác nhân (nhân viên, quản lý), các chức năng chính (quản lý món, bán hàng, thống kê, v.v.) và luồng xử lý thông tin.
- **Thiết kế mô hình hệ thống:** Xây dựng **biểu đồ Use Case**, **sơ đồ luồng dữ liệu (DFD)** và **sơ đồ thực thể – liên kết (ERD)** để mô tả cấu trúc dữ liệu và mối quan hệ giữa các bảng.
- **Thiết kế giao diện người dùng:** Tạo các mẫu giao diện web (HTML, CSS, Bootstrap) đảm bảo tính thân thiện và dễ sử dụng.

+ **Phương pháp xây dựng và cài đặt**

- Sử dụng **HTML, CSS, JavaScript** để xây dựng phần giao diện.
- Dùng **ngôn ngữ lập trình** (ví dụ **PHP hoặc ASP.NET**) để xử lý dữ liệu phía máy chủ.
- Cơ sở dữ liệu sử dụng **MySQL hoặc SQL Server** để lưu trữ thông tin món, hóa đơn, nhân viên, doanh thu.
- Kết nối và kiểm thử giữa các lớp giao diện – xử lý – dữ liệu nhằm đảm bảo hệ thống hoạt động đúng yêu cầu.

+ Phương pháp kiểm thử và đánh giá

- **Kiểm thử chức năng (Functional Testing):** Đảm bảo từng chức năng (đăng nhập, quản lý món, tạo hóa đơn, thống kê) hoạt động chính xác.
- **Kiểm thử giao diện (UI Testing):** Đánh giá mức độ thân thiện, tính trực quan và tính nhất quán của các trang web.
- **Đánh giá kết quả:** So sánh giữa kết quả thực tế và yêu cầu ban đầu để xác định mức độ hoàn thiện của hệ thống.

CHƯƠNG 2. CƠ SỞ LÝ THUYẾT

2.1. Quy trình phát triển phần mềm

+ Quy trình xây dựng phần mềm bao gồm tập hợp các thao tác và kết quả tương quan sử dụng trong việc phát triển để sản xuất ra một sản phẩm phần mềm. Mỗi một phần mềm lại có đặc điểm và yêu cầu khác nhau, tuy nhiên tất cả đều cần phải trải qua các bước sau:

- **Bước 1: Phân tích yêu cầu:**

Phân tích yêu cầu là công việc bao gồm các tác vụ xác định yêu cầu cho một hệ thống mới hoặc được thay đổi dựa trên cơ sở là các nhu cầu trong quá trình sử dụng. Việc phân tích yêu cầu có ý nghĩa quan trọng đối với thành công của một dự án. Các yêu cầu phải có tính đo được, kiểm thử được, có liên quan đến các nhu cầu hoặc cơ hội doanh nghiệp đã được xác định, và phải được định nghĩa ở mức độ chi tiết đủ cho việc thiết kế hệ thống.

- **Bước 2: Thiết kế phần mềm**

Là một quá trình giải quyết vấn đề và lập kế hoạch cho một giải pháp phần mềm. Sau khi các mục đích và các đặc điểm kỹ thuật của phần mềm được giải quyết, lập trình viên sẽ thiết kế hoặc thuê người thiết kế để phát triển một kế hoạch cho giải pháp phần mềm. Nó bao gồm các thành phần cấp thấp, các vấn đề thuật toán cũng như một khung nhìn kiến trúc. Thiết kế chức năng, cơ sở dữ liệu và giao diện.

- **Bước 3: Lập trình máy tính**

Lập trình máy tính (gọi tắt là lập trình) là kỹ thuật cài đặt một hoặc nhiều thuật toán trừu tượng có liên quan với nhau bằng một hoặc nhiều ngôn ngữ lập trình để tạo ra một chương trình máy tính có các thành tố nghệ thuật, khoa học, toán học, kỹ nghệ. Các ngôn ngữ lập trình khác nhau hỗ trợ các phong cách lập trình khác nhau. Một phần của công việc lập trình là việc lựa chọn một trong những ngôn ngữ phù hợp nhất với các bài toán cần giải quyết. Các ngôn ngữ lập trình khác nhau đòi hỏi lập trình viên phải xử lý các chi tiết ở mức độ khác nhau khi cài đặt các thuật toán. Sự thống nhất trong các cách xử lý sẽ tạo thuận lợi cho việc lập trình và hiệu quả của chương trình.

- **Bước 4: Kiểm thử phần mềm**

Kiểm thử phần mềm là một cuộc kiểm tra được tiến hành để cung cấp cho các bên liên quan thông tin về chất lượng của sản phẩm hoặc dịch vụ được kiểm thử. Kiểm thử có thể cung cấp cho doanh nghiệp một quan điểm, một cách nhìn độc đáo về phần mềm để từ đó đánh giá và thấu hiểu được những rủi ro trong quá trình triển khai phần mềm. Tùy thuộc vào từng phương pháp, việc kiểm thử có thể được thực hiện bất cứ lúc nào trong quá trình phát triển phần mềm. Theo truyền thống thì các nỗ lực kiểm thử được tiến hành sau khi các yêu cầu được xác định và việc lập trình được hoàn tất trong phương pháp phát triển “Agile” thì việc kiểm thử được tiến hành liên tục trong suốt quá trình xây dựng phần mềm. Như vậy, mỗi một phương pháp kiểm thử bị chi phối theo một quy trình phát triển phần mềm nhất định.

- **Bước 5: Triển khai phần mềm**

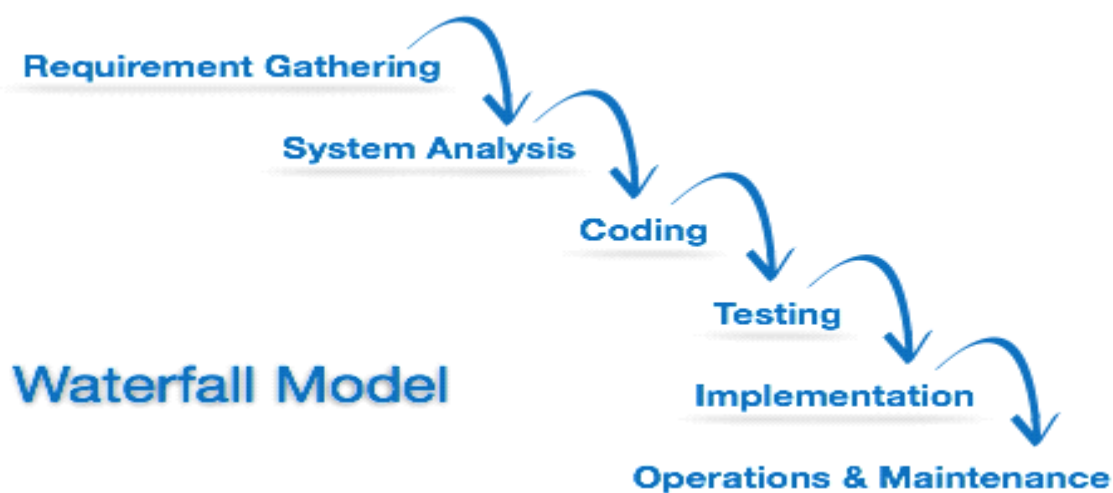
Sau khi phần mềm được kiểm thử và khắc phục những sai sót sẽ được triển khai đưa vào sử dụng trong thực tế. Đối với những phần mềm thiết kế theo thỏa thuận với khách hàng, việc triển khai đơn giản chỉ là hướng dẫn cho khách hàng cách sử dụng đạt hiệu quả cao. Với những phần mềm mang tính thông dụng, việc triển khai còn qua các chương trình giới thiệu và đưa sản phẩm ra thị trường. Trong quá trình triển khai cũng luôn đánh giá hiệu quả sử dụng của phần mềm, xem xét những nhược điểm để lên kế hoạch thiết kế phần mềm hiệu quả hơn.

- **Bước 6: Bảo trì phần mềm**

Bảo trì phần mềm bao gồm điều chỉnh các lỗi mà chưa được phát hiện trong các giai đoạn trước chu kỳ sống của phần mềm, nâng cấp tính năng sử dụng và an toàn vận hành của phần mềm. Bảo trì phần mềm có thể chiếm đến 65%-75% công sức chu kỳ sống của một phần mềm. Quá trình phát triển phần mềm bao gồm rất nhiều giai đoạn: thu thập yêu cầu, phân tích, xây dựng, kiểm tra, triển khai và bảo trì phần mềm. Nhiệm vụ của giai đoạn bảo trì phần mềm là giữ cho phần mềm được cập nhật khi môi trường thay đổi và yêu cầu người sử dụng thay đổi.

Mỗi một giai đoạn xây dựng phần mềm lại đòi hỏi các kỹ năng phân tích và ứng dụng kiến thức công nghệ khác nhau. Để xây dựng phần mềm thiết thực, mang lại hiệu quả kinh tế cao đòi hỏi con người lập trình viên phải tuân thủ các yêu cầu trong từng giai đoạn thiết kế.

2.1.1. Waterfall model – Mô hình thác nước



Hình 2. 1: Mô hình thác nước

+ Mô tả:

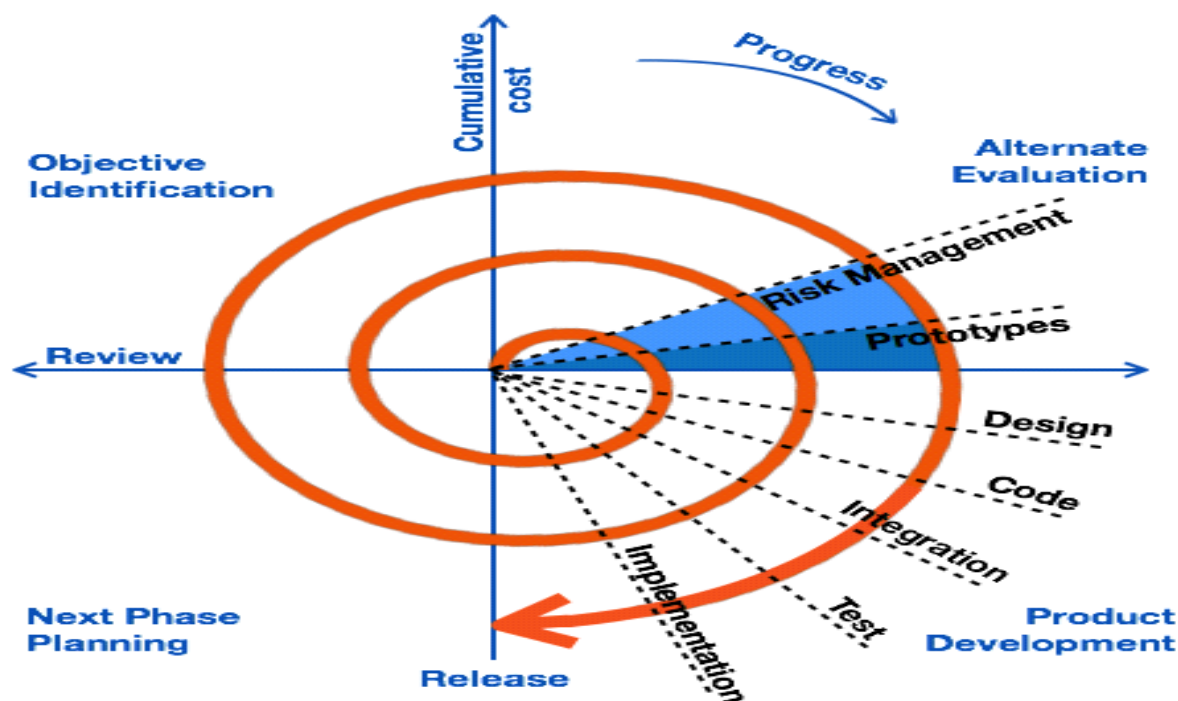
- Mô hình thác nước là mô hình áp dụng theo tính tuần tự của các giai đoạn phát triển phần mềm.
- Có nghĩa là: giai đoạn sau chỉ được phép thực hiện tiếp khi giai đoạn trước đã kết thúc.
- Không được quay lại giai đoạn trước để xử lý các thay đổi trong yêu cầu.
- Đây được coi là mô hình phát triển phần mềm đầu tiên.

+ Áp dụng: Thường được áp dụng cho các dự án không thường xuyên thay đổi về yêu cầu.

+ Đặc điểm:

- Ưu điểm:
 - Dễ sử dụng, dễ tiếp cận.
 - Các giai đoạn và hoạt động được xác định rõ ràng.
 - Xác nhận ở từng giai đoạn, đảm bảo phát hiện sớm lỗi.
- Nhược điểm:
 - Rất khó để quay lại giai đoạn nào đó khi nó đã kết thúc.
 - Ít tính linh hoạt và phạm vi điều chỉnh của nó khá là khó khăn, tốn kém.

2.1.3. Spiral Model – Mô hình xoắn ốc



Hình 2. 3: Mô hình xoắn ốc

+ Mô tả:

- Là mô hình kết hợp giữa các tính năng của mô hình prototyping và mô hình thác nước.
- Mô hình xoắn ốc được ưa chuộng cho các dự án lớn, đắt tiền và phức tạp.
- Mô hình này sử dụng nhiều những giai đoạn tương tự như mô hình thác nước, về thứ tự, plan, đánh giá rủi ro, ...

+ Áp dụng: Thường được sử dụng cho các ứng dụng lớn và các hệ thống được xây dựng theo các giai đoạn nhỏ hoặc theo các phân đoạn

+ Đặc điểm:

- Ưu điểm:
 - Estimates (i.e. budget, schedule, etc.) trở nên thực tế hơn như là một quy trình làm việc, bởi vì những vấn đề quan trọng đã được phát hiện sớm hơn.
 - Có sự tham gia sớm của developers
 - Quản lý rủi ro và phát triển hệ thống theo phase
- Nhược điểm:
 - Chi phí cao và thời gian dài để có sản phẩm cuối cùng

- Phải có kỹ năng tốt để đánh giá rủi ro và giả định.

2.2. Thiết kế giao diện web với HTML, CSS, JS

2.2.1. HTML (*Hypertext Markup Language*)

+ HTML là gì?

- HTML (viết tắt của từ Hypertext Markup Language, hay là “Ngôn ngữ Đánh dấu Siêu văn bản”) là một ngôn ngữ đánh dấu được thiết kế ra để tạo nên các trang web trên World Wide Web. Nó có thể được trợ giúp bởi các công nghệ như CSS và các ngôn ngữ kịch bản giống như JavaScript.
- Các trình duyệt web nhận tài liệu HTML từ một web server hoặc một kho lưu trữ cục bộ và render tài liệu đó thành các trang web đa phương tiện. HTML mô tả cấu trúc của một trang web về mặt ngữ nghĩa và các dấu hiệu ban đầu được bao gồm cho sự xuất hiện của tài liệu.
- Các phần tử HTML là các khối xây dựng của các trang HTML. Với cấu trúc HTML, hình ảnh và các đối tượng khác như biểu mẫu tương tác có thể được nhúng vào trang được hiển thị. HTML cung cấp một phương tiện để tạo tài liệu có cấu trúc bằng cách biểu thị ngữ nghĩa cấu trúc cho văn bản như headings, paragraphs, lists, links, quotes và các mục khác. Các phần tử HTML được phân định bằng các tags, được viết bằng dấu ngoặc nhọn. Các tags như và giới thiệu trực tiếp nội dung vào trang. Các tags khác như bao quanh và cung cấp thông tin về văn bản tài liệu và có thể bao gồm các thẻ khác làm phần tử phụ. Các trình duyệt không hiển thị các thẻ HTML, nhưng sử dụng chúng để diễn giải nội dung của trang.
- HTML có thể nhúng các chương trình được viết bằng scripting như JavaScript, điều này ảnh hưởng đến hành vi và nội dung của các trang web. Việc bao gồm CSS xác định giao diện và bố cục của nội dung. World Wide Web Consortium (W3C), trước đây là đơn vị bảo trì HTML và là người duy trì hiện tại của các tiêu chuẩn CSS, đã khuyến khích việc sử dụng CSS trên HTML trình bày rõ ràng kể từ năm 1997

+ Lịch sử phát triển HTML

- HTML được tạo ra bởi Tim Berners-Lee, một nhà vật lý học của trung tâm nghiên cứu CERN ở Thụy Sĩ. Hiện nay, HTML đã trở thành một chuẩn Internet được tổ chức W3C (World Wide Web Consortium) vận hành và phát triển. Bạn có thể tự tìm kiếm tình trạng mới nhất của HTML tại bất kỳ thời điểm nào trên Website của W3C.
- Phiên bản đầu tiên của HTML xuất hiện năm 1991, gồm 18 tag HTML. Phiên bản HTML 4.01 được xuất bản năm 1999. Sau đó, các nhà phát triển đã thay thế HTML bằng XHTML vào năm 2000.
- Đến năm 2014, HTML được nâng cấp lên chuẩn HTML5 với nhiều tag được thêm vào markup, mục đích là để xác định rõ nội dung thuộc loại là gì (ví dụ như: <article>, <header>, <footer>, ...).
- Theo Mozilla Developer Network thì HTML Element Reference hiện nay có khoảng hơn 140 tag. Tuy nhiên một vài tag trong số đó đã bị tạm ngưng (do không được hỗ trợ bởi các trình duyệt hiện hành).

+ HTML hoạt động như thế nào?

- HTML documents là files kết thúc với đuôi .html hay .htm. Bạn có thể xem chúng bằng cách sử dụng bất kỳ trình duyệt web nào (như Google Chrome, Safari, hay Mozilla Firefox). Trình duyệt đọc các files HTML này và xuất bản nội dung lên internet sao cho người đọc có thể xem được nó.
- Thông thường, trung bình một web chứa nhiều trang web HTML, ví dụ như: trang chủ, trang about, trang liên hệ, tất cả đều cần các trang HTML riêng.
- Mỗi trang HTML chứa một bộ các tag (cũng được gọi là elements), bạn có thể xem như là việc xây dựng từng khối của một trang web. Nó tạo thành cấu trúc cây thư mục bao gồm section, paragraph, heading, và những khối nội dung khác.

+ Cấu trúc của HTML

- Hầu hết các HTML elements đều có tag mở và tag đóng



Hình 2. 4: Cấu trúc của html

- Các thẻ trong HTML thường có những thẻ cơ bản như sau:

Bảng 2. 1: Các thẻ cơ bản trong html

Thẻ	Chức năng
<code><h1> ... <h6></code>	Tiêu đề (heading) từ lớn nhất (h1) đến nhỏ nhất (h6)
<code><p></code>	Đoạn văn bản (paragraph)
<code></code>	Liên kết (hyperlink)
<code></code>	Chèn hình ảnh
<code> / / </code>	Danh sách không thứ tự / có thứ tự
<code><table>, <tr>, <td>, <th></code>	Bảng dữ liệu
<code><form></code>	Biểu mẫu nhập liệu
<code><div></code>	Khối chứa (block)
<code></code>	Đoạn nội dung nhỏ trong dòng (inline)

+ Ưu và nhược điểm HTML

- HTML là một ngôn ngữ đánh dấu siêu văn bản nên nó sẽ có vai trò xây dựng cấu trúc siêu văn bản trên một website, hoặc khai báo các tập tin kỹ thuật số (media) như hình ảnh, video, nhạc. Tuy nhiên, HTML có ưu và nhược điểm của riêng nó.
- Ưu điểm:
 - Được sử dụng rộng rãi, có rất nhiều nguồn tài nguyên hỗ trợ và cộng đồng sử dụng lớn.
 - Học đơn giản và dễ hiểu.
 - Mã nguồn mở và hoàn toàn miễn phí.
 - Markup gọn gàng và đồng nhất.
 - Tiêu chuẩn thế giới được vận hành bởi World Wide Web Consortium (W3C).
 - Dễ dàng tích hợp với các ngôn ngữ backend như PHP, Python...
- Nhược điểm:
 - Được dùng chủ yếu cho web tĩnh. Đối với các tính năng động như update hay realtime thời gian thực, bạn cần sử dụng JavaScript hoặc ngôn ngữ backend bên thứ 3 như PHP.
 - Một số trình duyệt chậm hỗ trợ tính năng mới.

+ HTML đóng vai trò gì trong website

- Với những ưu và nhược điểm trên, điều đó không có nghĩa là chỉ sử dụng HTML để tạo ra một website mà HTML chỉ đóng một vai trò hình thành trên website. Một website chuẩn sẽ được hình thành bởi:
 - HTML – Xây dựng cấu trúc và định dạng các siêu văn bản.
 - CSS – Định dạng các siêu văn bản dạng thô tạo ra từ HTML thành một bố cục website, có màu sắc, ảnh nền,....

- Javascript – Tạo ra các sự kiện tương tác với hành động của người dùng (ví dụ như là chat, update nội dung, hiệu ứng slide).
- PHP – Ngôn ngữ lập trình để xử lý và trao đổi dữ liệu giữa máy chủ đến trình duyệt.
- MySQL – Hệ quản trị cơ sở dữ liệu truy vấn có cấu trúc.

2.2.2. CSS (*Cascading Style Sheet Language*)

+ CSS là gì?

- CSS (Cascading Style Sheets) là một ngôn ngữ biểu diễn (style language) giúp định dạng và làm đẹp cho các trang web được viết bằng HTML hoặc XML. Trong khi HTML tạo ra các thành phần chính của trang web như tiêu đề, đoạn văn, hình ảnh, thì CSS quyết định cách những thành phần này sẽ xuất hiện: từ màu sắc, phông chữ, đến bố cục và cách sắp xếp trên trang.

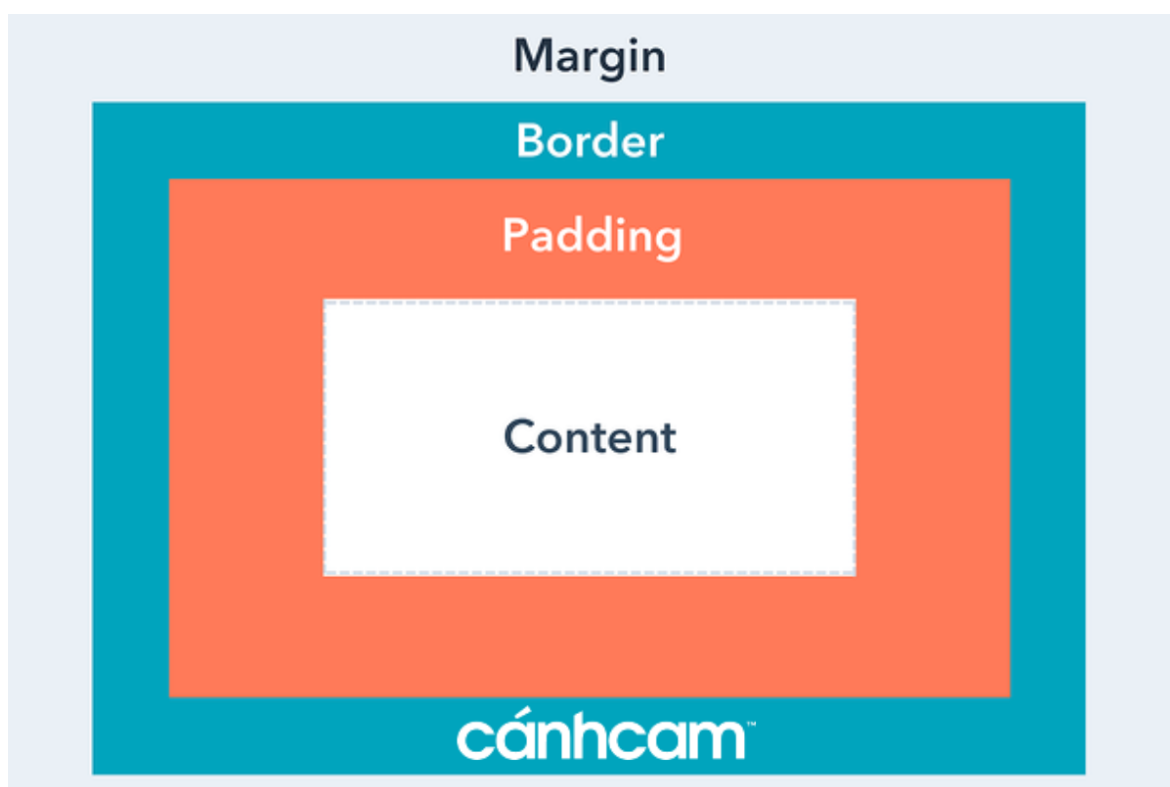


Hình 2. 5: Ví dụ về css

- Điểm đặc biệt của CSS là nó cho phép tách biệt phần nội dung và cấu trúc (được viết bằng HTML) khỏi phần trình bày, giúp dễ dàng thay đổi giao diện trang web mà không ảnh hưởng đến nội dung. Nói một cách đơn giản, HTML tạo nên khung sườn của trang web, còn CSS chịu trách nhiệm làm cho trang web trở nên bắt mắt và chuyên nghiệp hơn.

+ Bố cục của một đoạn CSS là gì?

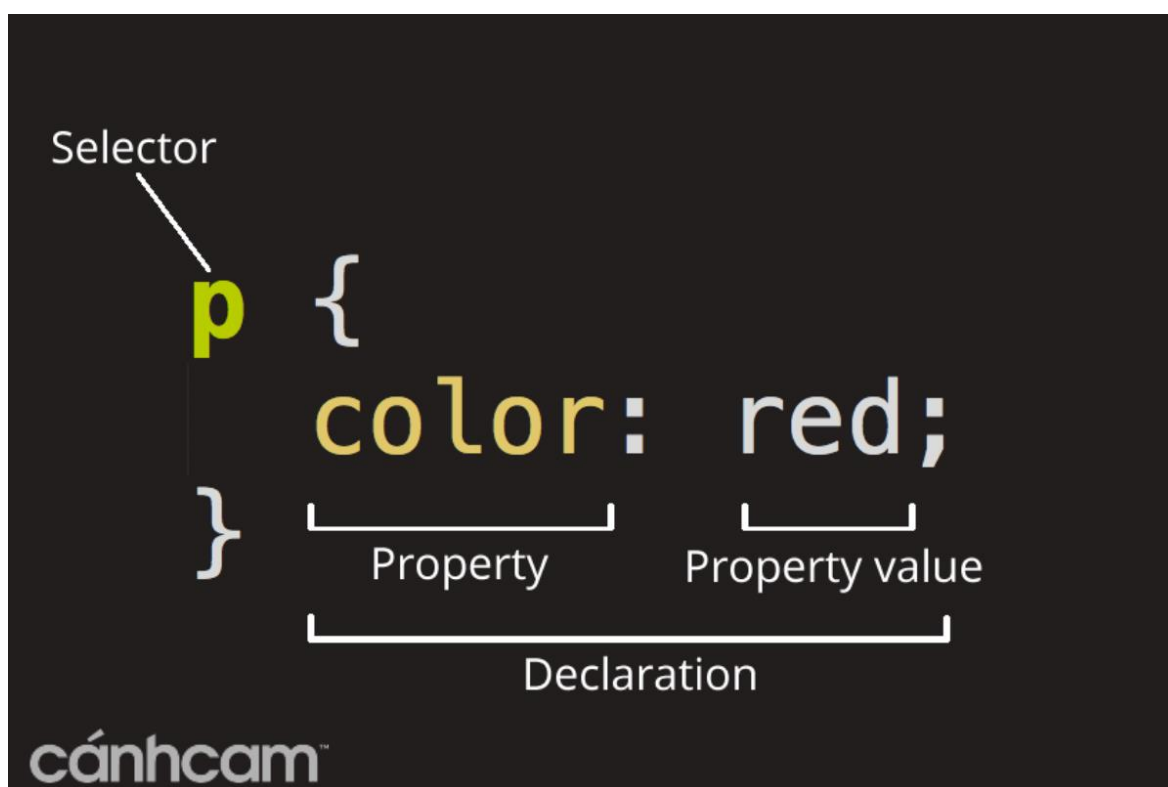
- Bố cục của css gồm những vùng sau:
 - Padding (Vùng đệm): Không gian xung quanh nội dung, chẳng hạn như không gian xung quanh một đoạn văn bản.
 - Border (Đường viền): Là đường liền nằm bên ngoài phần đệm CSS.
 - Margin (Lề): Khoảng cách xung quanh nằm ở ngoài phần tử.



Hình 2. 6: Bố cục của css

+ Cấu trúc của một đoạn CSS là gì?

- Một đoạn CSS bao gồm các phần như sau:



Hình 2. 7: Cấu trúc của css

- Cấu trúc này mang ý nghĩa khai báo bằng vùng chọn, các thuộc tính và giá trị sẽ nằm trong cặp dấu ngoặc nhọn { }. Mỗi thuộc tính sẽ luôn có một giá trị riêng. Phần giá trị và thuộc tính cách nhau bằng dấu hai chấm, và mỗi một dòng khai báo thuộc tính sẽ luôn có dấu chấm phẩy ở cuối. Một vùng chọn có thể sử dụng không giới hạn thuộc tính.

+ Trong đó:

- Selector (Bộ chọn): Selector cho phép người sử dụng có thể lựa chọn các phần tử HTML mà không làm ảnh hưởng đến xung quanh. Các Selector được áp dụng cho các trường hợp sau:
 - Tất cả phần tử theo một dạng cụ thể nào đó.
 - Thuộc tính ID và class của các phần tử.
 - Các phần tử dựa vào mối liên quan với các phần tử khác trong cây phân cấp tài liệu.

Selector	Ý Nghĩa	Ví Dụ
Universal Selector	Loại selector này sẽ tác động đến tất cả các thẻ HTML trên trang web	*{ property: value; }
Type Selector	Loại CSS này sẽ tác động đến các thẻ HTML mà chúng ta muốn tác động	h1, h2, h3 {property: value;}
Class Selector	Loại selector này sẽ tác động đến tất cả các thẻ HTML có class mà nó quy định	.note {property: value;} ; p.note {};
ID Selector	Loại selectors này sẽ tác động đến tất cả các thẻ HTML có id mà nó quy định	#introduction {property: value;}
Child Selector	Loại css này sẽ cho phép chúng ta css vào các thẻ HTML theo cấp cha con	li>a {property: value;}
Descendant Selector	Chọn bất kì phần tử con nào trong phần tử cha	p a {property: value;}
Adjacent Sibling Selector	Nhằm chọn các phần tử anh chị em	h1+p {property: value;}
General Sibling Selector	Nhằm chọn các phần tử anh chị em mặc dù nó không nhất thiết phải là phần tử đứng trước trực tiếp	h1~p {property: value;}

Hình 2. 8: Sơ lược về bộ chọn

- Declaration (Khai báo): Declaration có chức năng xác định thuộc tính của một phần tử bất kỳ trong chương trình. Khởi khai báo chứa một hoặc nhiều khai báo, phân tách với nhau bằng các dấu chấm phẩy.
- Properties (Thuộc tính): Những cách mà bạn có thể tạo kiểu cho một phần tử HTML. (Với trường hợp này thì font-size được xem là một trong những thuộc tính của phần tử p). Chính vì vậy, với CSS thì bạn chỉ cần lựa chọn thuộc tính mà chính bạn muốn tác động nhất trong bộ quy tắc của mình.
- Giá trị thuộc tính: Ở bên phải của thuộc tính sau dấu hai chấm(:), chúng ta sẽ sở hữu giá trị thuộc tính mà việc lựa chọn trong số đó phụ thuộc vào số lần xuất hiện của thuộc tính.

+ Một số thuộc tính thường dùng trong CSS

Bảng 2. 2: Thuộc tính về văn bản và chữ

Thuộc tính	Chức năng	Ví dụ
color	Màu chữ	color: red;
font-family	Font chữ	font-family: Arial, sans-serif;
font-size	Kích thước chữ	font-size: 16px;

font-style	Kiểu chữ (ngghiêng/italic)	font-style: italic;
font-weight	Độ đậm nhạt chữ	font-weight: bold;
text-align	Căn lề chữ (trái, phải, giữa, justify)	text-align: center;
text-decoration	Trang trí chữ (gạch chân, gạch ngang...)	text-decoration: underline;
line-height	Chiều cao dòng	line-height: 1.5;
letter-spacing	Khoảng cách giữa các ký tự	letter-spacing: 2px;
word-spacing	Khoảng cách giữa các từ	word-spacing: 5px;

Bảng 2. 3: Thuộc tính về màu sắc và nền

Thuộc tính	Chức năng	Ví dụ
background-color	Màu nền	background-color: yellow;
background-image	Hình nền	background-image: url('bg.jpg');
background-repeat	Lặp hình nền	background-repeat: no-repeat;
background-size	Kích thước hình nền	background-size: cover;
background-position	Vị trí hình nền	background-position: center;
opacity	Độ mờ trong suốt	opacity: 0.5;

Bảng 2. 4: Thuộc tính về kích thước và bố cục

Thuộc tính	Chức năng	Ví dụ
width	Chiều rộng	width: 200px;
height	Chiều cao	height: 100px;

max-width / min-width	Giới hạn chiều rộng	max-width: 100%;
max-height / min-height	Giới hạn chiều cao	min-height: 50px;
margin	Khoảng cách ngoài	margin: 10px;
padding	Khoảng cách trong	padding: 20px;
border	Viền	border: 1px solid black;
border-radius	Bo tròn góc	border-radius: 10px;
box-shadow	Đổ bóng	box-shadow: 0 0 10px gray;

Bảng 2. 5: Thuộc tính về hiển thị và vị trí

Thuộc tính	Chức năng	Ví dụ
display	Kiểu hiển thị (block, inline, flex, grid)	display: flex;
position	Vị trí (static, relative, absolute, fixed, sticky)	position: absolute;
top, right, bottom, left	Khoảng cách đến cạnh khi dùng position	top: 20px; left: 50px;
float	Đẩy phần tử sang trái/phải	float: right;
clear	Ngăn float chồng lấn	clear: both;
z-index	Thứ tự chồng lớp	z-index: 100;
overflow	Xử lý tràn nội dung (hidden, scroll, auto)	overflow: hidden;
visibility	Hiện/ẩn phần tử	visibility: hidden;

Bảng 2. 6: Thuộc tính về danh sách và bảng

Thuộc tính	Chức năng	Ví dụ
list-style-type	Kiểu ký hiệu danh sách	list-style-type: square;
list-style-image	Ảnh làm ký hiệu danh sách	list-style-image: url('icon.png');
table-layout	Cách bố trí bảng	table-layout: fixed;
border-collapse	Gộp/không gộp viền ô	border-collapse: collapse;
border-spacing	Khoảng cách giữa các ô	border-spacing: 5px;
caption-side	Vị trí tiêu đề bảng	caption-side: bottom;

Bảng 2. 7: CSS nâng cao (hiệu ứng & animation)

Thuộc tính	Chức năng	Ví dụ
transition	Tạo hiệu ứng chuyển đổi mượt	transition: all 0.3s ease;
transform	Biến đổi (xoay, phóng to, nghiêng...)	transform: rotate(45deg);
animation	Tạo hoạt ảnh	animation: bounce 2s infinite;
cursor	Thay đổi con trỏ chuột	cursor: pointer;

+ Cách nhúng CSS vào website

- Để CSS có thể thực thi trên website hoặc HTML Documents thì phải tiến hành nhúng CSS vào website. Khi nói đến việc áp dụng CSS cho các trang web, thường có ba 3 cách nhúng CSS vào website:
 - Inline CSS – Nhúng trực tiếp vào tài liệu HTML thông qua cặp thẻ `<style></style>`.
 - Internal CSS – Dùng thẻ `<style>` bên trong thẻ `<head>` của HTML để tạo ra nơi viết mã CSS.

- External CSS – Bằng cách sử dụng phần tử <link> để liên kết với tệp CSS bên ngoài

+ **Inline CSS**

- CSS nội tuyến được sử dụng để áp dụng một kiểu duy nhất cho một phần tử HTML.

```
<h1 style="color:blue;">A Blue Heading</h1>  
<p style="color:red;">A red paragraph.</p>
```

Hình 2. 9: CSS Inline

- CSS nội tuyến sử dụng thuộc tính style của một phần tử HTML. Mã CSS chỉ tác động lên chính phần tử đó.
- Ví dụ: Đặt màu văn bản của phần tử <h1> thành màu xanh lam- blue và màu văn bản của phần tử <p> thành màu đỏ-red

+ **Internal CSS**

- CSS nội bộ được sử dụng để xác định kiểu cho một trang HTML.
- CSS nội bộ được xác định trong phần <head> của trang HTML, trong phần tử <style>
- Ví dụ: Đặt màu văn bản của tất cả các phần tử <h1> thành màu xanh lam-blue và màu văn bản của tất cả các phần tử <p> thành màu đỏ-red. Ngoài ra, trang sẽ được hiển thị với màu nền “powderblue”


```
<!DOCTYPE html>
<html>
<head>
<style>
body {background-color: powderblue;}
h1 {color: blue;}
p {color: red;}
</style>
</head>
<body>

<h1>This is a heading</h1>
<p>This is a paragraph.</p>

</body>
</html>
```

Hình 2. 10: CSS Internal

+ External CSS

- Với External CSS: Ta đặt các thuộc tính định dạng vào bên trong tập tin CSS. External CSS được sử dụng để xác định kiểu cho nhiều trang HTML.
- External CSS có thể được viết trong bất kỳ trình soạn thảo văn bản nào. Tập không được chứa bất kỳ mã HTML nào và phải được lưu bằng phần mở rộng .css.
- Để có thể thực hiện nhúng tập tin CSS vào trang web, hãy thêm liên kết đến biểu định kiểu này trong phần <head> của mỗi trang HTML

```
<head>
  <link rel="stylesheet" type="text/css" href="style.css">
</head>
```

Hình 2. 11: CSS External

2.2.3. JavaScript

+ JavaScript là một ngôn ngữ lập trình kịch bản ra đời vào năm 1995 bởi Brendan Eich khi ông đang làm việc tại Netscape. Ban đầu, JavaScript được phát triển với mục tiêu bổ sung tính năng động cho các trang web tĩnh vốn chỉ được xây dựng bằng HTML và CSS. Với đặc điểm là một ngôn ngữ thông dịch, JavaScript có thể chạy

trực tiếp trong trình duyệt mà không cần biên dịch, từ đó tạo ra sự linh hoạt, nhanh chóng và dễ triển khai. Ngày nay, JavaScript đã trở thành ngôn ngữ phổ biến nhất trong phát triển web và được hầu hết các trình duyệt hiện đại hỗ trợ.

+ Nếu như HTML chịu trách nhiệm xây dựng cấu trúc nội dung của một trang web, CSS phụ trách phần hiển thị và định dạng giao diện, thì JavaScript đóng vai trò tạo ra các tương tác động và xử lý các hành vi của người dùng. Nhờ JavaScript, một website không còn chỉ là một tập hợp văn bản và hình ảnh tĩnh mà trở thành một ứng dụng sống động, có khả năng phản hồi trực tiếp với các hành động như nhấn chuột, nhập dữ liệu, cuộn trang hoặc thay đổi nội dung hiển thị mà không cần tải lại toàn bộ trang.

+ Điểm đặc biệt ở JavaScript là ngôn ngữ này không giới hạn trong trình duyệt mà còn được mở rộng sang nhiều lĩnh vực khác nhau. Với sự ra đời của Node.js, JavaScript đã bước ra khỏi phạm vi front-end để trở thành một công cụ lập trình server-side mạnh mẽ. Chính điều này biến JavaScript trở thành ngôn ngữ “full-stack”, cho phép các lập trình viên chỉ cần một ngôn ngữ để phát triển cả phía giao diện người dùng (client-side) và phía máy chủ (server-side). Bên cạnh đó, JavaScript còn được sử dụng trong phát triển ứng dụng di động, desktop, thậm chí cả trí tuệ nhân tạo và IoT.

+ Vai trò và đặc điểm của JavaScript

- JavaScript là yếu tố không thể thiếu trong bộ ba công nghệ cốt lõi của web: HTML – CSS – JavaScript. HTML cung cấp khung sườn nội dung, CSS làm đẹp và định dạng bố cục, còn JavaScript giúp trang web trở nên thông minh và thân thiện hơn với người dùng. Khi người dùng thao tác trên website, ví dụ nhập dữ liệu vào một biểu mẫu, JavaScript sẽ giúp kiểm tra tính hợp lệ trước khi gửi dữ liệu về máy chủ, hoặc khi người dùng cuộn trang, JavaScript có thể thay đổi hiệu ứng hiển thị để mang lại trải nghiệm trực quan hơn.
- Một số đặc điểm quan trọng của JavaScript có thể kể đến:
 - Là ngôn ngữ thông dịch (interpreted language), không cần biên dịch trước.

- Là ngôn ngữ hướng đối tượng nhưng mang tính linh hoạt, hỗ trợ cả lập trình hàm.
- Có khả năng tương tác trực tiếp với DOM (Document Object Model) để thay đổi cấu trúc và nội dung của trang.
- Có cộng đồng phát triển rộng lớn, thư viện và framework phong phú, từ đó rút ngắn thời gian phát triển ứng dụng.

Bảng 2. 8: Các khái niệm và đặc điểm chính của JavaScript

Nội dung	Mô tả
Kiểu ngôn ngữ	Ngôn ngữ kịch bản, thông dịch trực tiếp, hướng đối tượng linh hoạt.
Môi trường chạy	Trình duyệt web (Chrome, Firefox, Edge...) và máy chủ (Node.js).
Ưu điểm	Linh hoạt, chạy được ở nhiều nền tảng, có cộng đồng lớn, nhiều framework hỗ trợ.
Nhược điểm	Dễ bị khai thác bảo mật, hiệu năng thấp hơn so với ngôn ngữ biên dịch.
Ứng dụng	Web động, ứng dụng di động (React Native), ứng dụng web SPA, game online, IoT.

+ JavaScript hoạt động như thế nào?

- Nguyên lý hoạt động của JavaScript khá đơn giản nhưng lại rất hiệu quả. Khi một trang web được tải lên, trình duyệt sẽ đọc và phân tích tài liệu HTML và CSS, sau đó khi gặp đoạn mã JavaScript, nó sẽ được thực thi ngay lập tức. JavaScript có thể truy cập và thay đổi các phần tử trong HTML thông qua DOM, từ đó cho phép cập nhật nội dung, thay đổi kiểu dáng hoặc phản hồi các sự kiện từ người dùng mà không cần tải lại trang.

- Ví dụ, trong một trang web thương mại điện tử, khi người dùng nhấn nút “Thêm vào giỏ hàng”, JavaScript sẽ lắng nghe sự kiện này, cập nhật số lượng sản phẩm trong giỏ ngay trên giao diện, đồng thời có thể gửi dữ liệu tới máy chủ để lưu lại đơn hàng. Tất cả những thao tác này diễn ra trong vài mili-giây, tạo cảm giác liền mạch cho trải nghiệm người dùng.



```
html Sao chép mã

<!DOCTYPE html>
<html lang="vi">
<head>
  <meta charset="UTF-8">
  <title>Ví dụ JavaScript</title>
</head>
<body>
  <h2>Ví dụ đơn giản với JavaScript</h2>
  <p id="demo">Nội dung ban đầu</p>
  <button onclick="thayDoiNoiDung()">Nhấn vào đây</button>

  <script>
    function thayDoiNoiDung() {
      document.getElementById("demo").innerHTML = "Nội dung đã được thay đổi bằng JavaScript!";
    }
  </script>
</body>
</html>
```

Hình 2. 12: Ví dụ minh họa

- Trong ví dụ này, đoạn văn bản có id là demo sẽ thay đổi ngay lập tức khi người dùng nhấn vào nút. Đây là minh chứng rõ ràng cho khả năng thao tác trực tiếp với DOM và phản hồi hành động của người dùng mà JavaScript mang lại.

+ Các ứng dụng phổ biến của JavaScript

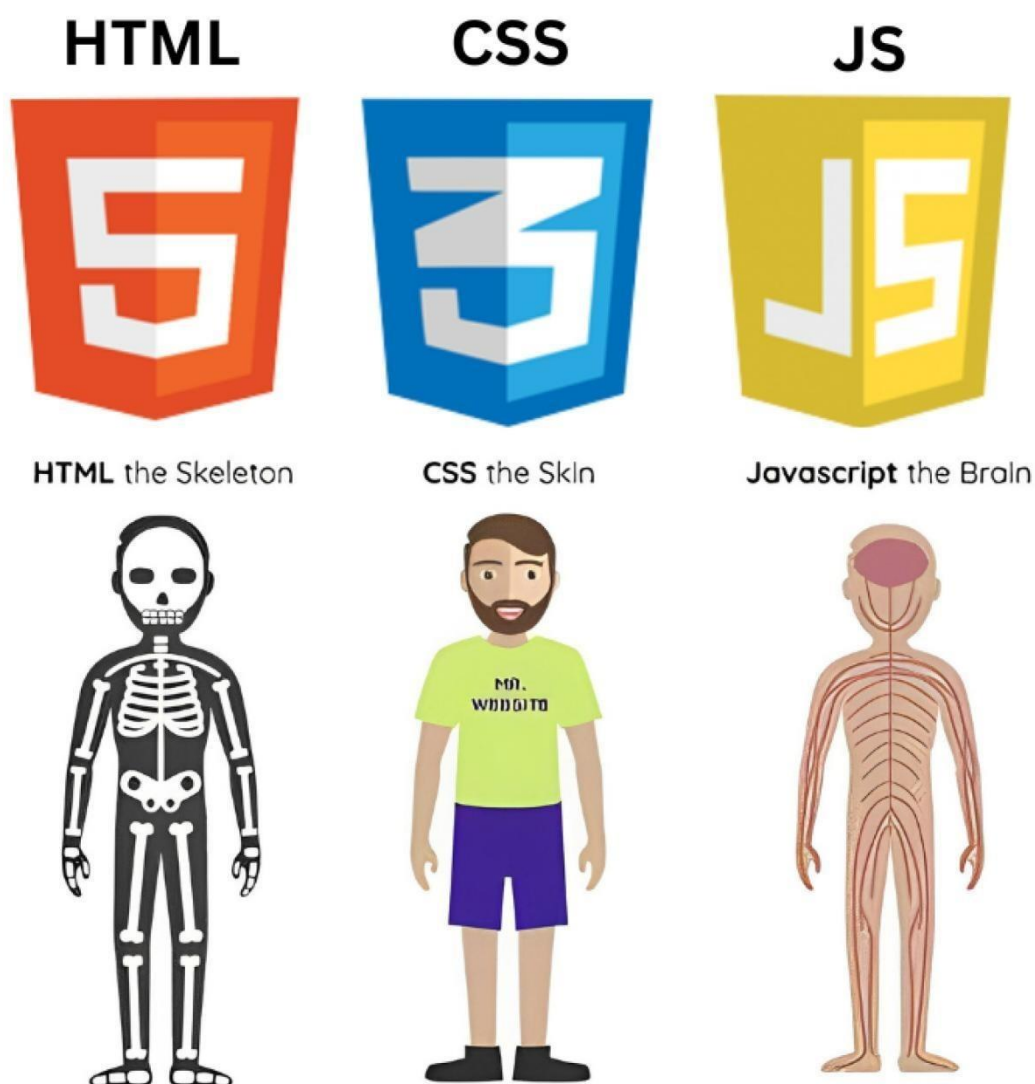
- JavaScript được ứng dụng trong rất nhiều lĩnh vực, từ những thao tác cơ bản đến các ứng dụng phức tạp. Trong phát triển web, JavaScript là nền tảng của các ứng dụng web một trang (SPA – Single Page Application) nơi mà toàn bộ trải nghiệm người dùng diễn ra trên một trang duy nhất nhờ khả năng tải dữ liệu động. Bên cạnh đó, JavaScript còn hỗ trợ công nghệ AJAX để giao tiếp bất đồng bộ với máy chủ, giúp cập nhật nội dung mà không cần tải lại toàn bộ trang.

- Trong lĩnh vực di động, các framework như React Native cho phép viết ứng dụng chạy trên cả Android và iOS bằng JavaScript. Trong khi đó, ở phía máy chủ, Node.js đã mở rộng khả năng của JavaScript, biến nó thành một ngôn ngữ full-stack, có thể xử lý cả logic phía backend lẫn giao diện người dùng. Ngoài ra, JavaScript còn được dùng trong phát triển game 2D/3D trên nền tảng web, lập trình IoT, và thậm chí cả trí tuệ nhân tạo thông qua các thư viện hỗ trợ như TensorFlow.js.

+ Mối quan hệ giữa HTML – CSS – JavaScript

- Trong phát triển web, HTML, CSS và JavaScript được xem là “bộ ba nền tảng” không thể tách rời. Mỗi công nghệ đảm nhận một vai trò riêng biệt nhưng lại gắn kết chặt chẽ với nhau để tạo nên một website hoàn chỉnh.
- HTML (HyperText Markup Language) là ngôn ngữ đánh dấu dùng để xây dựng khung sườn của trang web. HTML định nghĩa nội dung sẽ hiển thị, ví dụ như tiêu đề, đoạn văn, hình ảnh, bảng biểu, biểu mẫu... Có thể coi HTML là phần “xương sống” hay cấu trúc nền tảng của một trang web.
- CSS (Cascading Style Sheets) chịu trách nhiệm về phần định dạng và phong cách. CSS quyết định cách các phần tử HTML sẽ hiển thị như thế nào: màu sắc, kích thước chữ, bố cục, khoảng cách, hình nền, hiệu ứng giao diện... Nếu HTML là “xương sống”, thì CSS chính là “da thịt và quần áo” làm cho trang web trở nên bắt mắt, dễ nhìn và chuyên nghiệp.
- JavaScript là ngôn ngữ lập trình giúp trang web trở nên động và có khả năng tương tác với người dùng. JavaScript cho phép xử lý các sự kiện như khi người dùng nhấp chuột, nhập dữ liệu, cuộn trang hay gửi form. Ngoài ra, JavaScript có thể thay đổi nội dung HTML, thao tác với CSS, tải dữ liệu từ máy chủ mà không cần tải lại trang, từ đó tạo nên trải nghiệm liền mạch và hấp dẫn. Có thể coi JavaScript là “bộ não” giúp website hoạt động thông minh và sinh động.
- Khi ba thành phần này kết hợp lại, chúng tạo thành một hệ thống thống nhất: HTML cung cấp khung, CSS làm đẹp giao diện và JavaScript mang đến sự sống động, tương tác. Nếu thiếu một trong ba, website sẽ bị khiếm khuyết: chỉ có

HTML thì web khô khan, chỉ có HTML + CSS thì đẹp nhưng tĩnh, chỉ khi thêm JavaScript thì website mới thật sự hiện đại, thân thiện và gần gũi với người dùng.



Hình 2. 13: Sự kết hợp của HTML - CSS - JS

2.3. Lập trình phía front-end

+ **Front end** là gì?

- [Front end](#) (hay Frontend) là phần giao diện của một trang web hoặc ứng dụng web mà người dùng có thể nhìn thấy và tương tác trực tiếp trên trình duyệt của họ. Nó bao gồm tất cả các yếu tố hình ảnh, văn bản, nút bấm, menu, và cách chúng được sắp xếp, trình bày.

- Nói đơn giản, mọi thứ bạn thấy và chạm vào trên màn hình khi lướt web – từ việc đọc bài viết này, nhấp vào một liên kết, xem một video, điền vào một biểu mẫu – đều thuộc về Front end. Đây chính là “bộ mặt” của website, nơi diễn ra sự tương tác chính giữa người dùng và sản phẩm số.
- Để dễ hình dung hơn, hãy tưởng tượng website như một ngôi nhà. Front end chính là toàn bộ phần nội thất và ngoại thất bạn thấy: màu sơn tường, kiểu dáng cửa sổ, cách bài trí bàn ghế, tranh ảnh treo tường... Tất cả những gì tạo nên vẻ đẹp, sự tiện nghi và trải nghiệm khi bạn bước vào và sử dụng không gian đó.

+ Tại sao Front end lại quan trọng?

- Front end không chỉ đơn thuần là làm cho website “đẹp”. Nó đóng vai trò cực kỳ quan trọng trong sự thành công của một sản phẩm web vì những lý do chính sau:
- Tạo ấn tượng đầu tiên: Giao diện người dùng (User Interface – [UI](#)) là thứ đầu tiên “đập vào mắt” người dùng. Một Front end được thiết kế tốt, trực quan và thẩm mỹ sẽ tạo thiện cảm ban đầu, giữ chân người dùng ở lại khám phá lâu hơn. Ngược lại, giao diện lộn xộn, khó sử dụng sẽ khiến họ rời đi nhanh chóng.
- Quyết định Trải nghiệm Người dùng ([UX](#)): Front end chịu trách nhiệm chính cho việc người dùng tương tác với website có dễ dàng, mượt mà và hiệu quả hay không. Tốc độ tải trang nhanh, điều hướng rõ ràng, các nút bấm hoạt động đúng như mong đợi... tất cả đều góp phần tạo nên một UX tích cực.
- Tăng khả năng truy cập (Accessibility): Một Front end tốt cần đảm bảo rằng trang web có thể được sử dụng bởi tất cả mọi người, kể cả người khuyết tật (ví dụ: người khiếm thị sử dụng trình đọc màn hình). Điều này không chỉ mang tính nhân văn mà còn mở rộng đối tượng người dùng tiềm năng.
- Hỗ trợ mục tiêu kinh doanh: Giao diện thân thiện, quy trình rõ ràng (như thêm vào giỏ hàng, thanh toán) sẽ trực tiếp ảnh hưởng đến tỷ lệ chuyển đổi (conversion rate). Một Front end tối ưu có thể giúp tăng doanh số bán hàng, số lượt đăng ký hoặc các mục tiêu kinh doanh khác.

- Xây dựng nhận diện thương hiệu: Màu sắc, logo, font chữ và phong cách thiết kế nhất quán trên Front end giúp củng cố hình ảnh và nhận diện thương hiệu trong tâm trí người dùng. Nó tạo ra sự chuyên nghiệp và đáng tin cậy.
- Tóm lại, Front end là cầu nối trực tiếp giữa sản phẩm và người dùng. Đầu tư vào một Front end chất lượng không chỉ là đầu tư vào “vẻ bề ngoài” mà còn là đầu tư vào sự hài lòng của người dùng và sự thành công bền vững của dự án web.

+ Lập trình viên Front end làm những gì?

- Lập trình viên Front end (Front-end Developer) là những người “thổi hồn” vào các bản thiết kế tĩnh, biến chúng thành những trang web sống động, tương tác mà chúng ta sử dụng hàng ngày. Công việc của họ không chỉ đơn thuần là viết code mà bao gồm nhiều nhiệm vụ đa dạng:
- Chuyển đổi thiết kế thành [mã nguồn](#): Họ nhận các bản thiết kế giao diện (thường từ các công cụ như Figma, Sketch, Adobe XD) từ đội ngũ thiết kế UI/UX và sử dụng [HTML](#), [CSS](#), [JavaScript](#) để hiện thực hóa chúng thành giao diện web thực tế.
- Phát triển giao diện người dùng: Xây dựng các thành phần UI như menu điều hướng, nút bấm, biểu mẫu, [slider](#), modal... đảm bảo chúng hoạt động chính xác thân thiện với người dùng trên các trình duyệt và thiết bị khác nhau.
- Đảm bảo tính đáp ứng (Responsive Design): Một nhiệm vụ quan trọng là làm cho giao diện hiển thị tốt và hoạt động mượt mà trên mọi kích thước màn hình, từ máy tính để bàn lớn đến máy tính bảng và điện thoại di động.
- Tối ưu hiệu năng: Cải thiện tốc độ tải trang bằng các kỹ thuật như tối ưu hình ảnh, giảm thiểu mã CSS/JavaScript, sử dụng lazy loading... để mang lại trải nghiệm tốt nhất cho người dùng.
- Kiểm thử và sửa lỗi (Debugging): Tìm kiếm và khắc phục các lỗi hiển thị hoặc lỗi chức năng trên các trình duyệt và thiết bị khác nhau (cross-browser compatibility testing).

- Công việc của một Front-end Developer đòi hỏi sự kết hợp giữa tư duy logic, kỹ năng giải quyết vấn đề và một chút mắt thẩm mỹ để tạo ra những sản phẩm web vừa đẹp mắt vừa hiệu quả.

+ Các công nghệ và kỹ năng Front end cốt lõi

Để thực hiện những nhiệm vụ trên, Lập trình viên Front end cần nắm vững một bộ công nghệ và kỹ năng nền tảng. Dưới đây là những thành phần không thể thiếu:

- *HTML: Nền tảng cấu trúc Website*
 - HTML (HyperText Markup Language) là ngôn ngữ đánh dấu siêu văn bản, được coi là bộ khung xương của mọi trang web. Nó không phải là ngôn ngữ lập trình theo nghĩa truyền thống, mà dùng các thẻ (tags) để định nghĩa cấu trúc và ý nghĩa ngữ nghĩa cho nội dung.
 - Ví dụ, thẻ <h1> định nghĩa tiêu đề chính, thẻ <p> định nghĩa một đoạn văn bản, thẻ dùng để chèn hình ảnh, và thẻ <a> tạo ra các liên kết. Sử dụng HTML đúng cách, đặc biệt là HTML ngữ nghĩa (Semantic HTML), giúp trình duyệt và các công cụ tìm kiếm hiểu rõ hơn về nội dung trang web.
- *CSS: “Nghệ thuật” tạo kiểu và định dạng*
 - Nếu HTML là khung xương, thì CSS (Cascading Style Sheets) chính là lớp “da thịt” và “quần áo”, quyết định toàn bộ diện mạo của trang web. CSS cho phép bạn kiểm soát mọi thứ liên quan đến trình bày và thẩm mỹ.
 - Bạn có thể dùng CSS để thay đổi màu sắc văn bản, màu nền, font chữ, kích thước chữ, khoảng cách giữa các phần tử, tạo bố cục (layout) phức tạp, thêm hiệu ứng đổ bóng, bo góc và nhiều hơn nữa. Các CSS Preprocessors như Sass hoặc Less cũng thường được sử dụng để viết CSS hiệu quả và dễ bảo trì hơn.
- *JavaScript: Thêm “sự sống” và tính tương tác*
 - JavaScript (JS) là một ngôn ngữ lập trình thực thụ, đóng vai trò then chốt trong việc tạo ra các [trang web động](#) và tương tác. Nếu HTML và CSS tạo nên cấu trúc và vẻ ngoài tĩnh, thì JS thổi hồn vào đó, cho phép trang web phản hồi lại hành động của người dùng.

- Nhờ JavaScript, chúng ta có thể:
 - Tạo slideshow ảnh tự động chuyển đổi.
 - Hiện thị/ẩn các phần tử khi người dùng nhấp chuột.
 - Kiểm tra tính hợp lệ của dữ liệu người dùng nhập vào biểu mẫu (form validation) ngay lập tức.
 - Tải dữ liệu mới từ [máy chủ](#) mà không cần tải lại toàn bộ trang (kỹ thuật [AJAX](#)).
 - Tạo các hiệu ứng động, game đơn giản ngay trên trình duyệt.
- *Frameworks & Thư viện (React, Angular, Vue.js...)*
 - Khi các ứng dụng web trở nên phức tạp hơn, việc chỉ sử dụng HTML, CSS, JS thuần có thể trở nên khó quản lý. Đó là lúc các [Framework](#) và Thư viện ([Library](#)) JavaScript phát huy tác dụng. Chúng cung cấp các cấu trúc, mẫu thiết kế và thành phần dựng sẵn giúp lập trình viên xây dựng giao diện nhanh hơn, hiệu quả hơn và có tổ chức hơn.
 - [React](#) (thư viện phát triển bởi Facebook): Rất phổ biến, tập trung vào việc xây dựng giao diện người dùng theo mô hình component (thành phần).
 - [Angular](#) (framework phát triển bởi Google): Cung cấp một giải pháp toàn diện hơn, bao gồm nhiều tính năng tích hợp sẵn cho các ứng dụng lớn.
 - [Vue.js](#) (framework cộng đồng): Được đánh giá là dễ tiếp cận, linh hoạt và có hiệu năng tốt.
 - Việc lựa chọn framework/thư viện nào phụ thuộc vào yêu cầu dự án và sở thích của đội ngũ phát triển. Tuy nhiên, nền tảng vững chắc về JavaScript thuần vẫn là điều kiện tiên quyết.
- *Kỹ năng khác: Responsive Design, Debugging, Git...*
 - Ngoài bộ ba công nghệ cốt lõi và các framework, một lập trình viên Front end giỏi cần trang bị thêm nhiều kỹ năng quan trọng khác:

- [Responsive Web Design](#): Kỹ năng thiết kế và xây dựng giao diện có thể tự động điều chỉnh để hiển thị tối ưu trên mọi loại thiết bị (desktop, tablet, mobile).
- Debugging (Gỡ lỗi): Khả năng sử dụng các công cụ dành cho nhà phát triển (Developer Tools) tích hợp sẵn trong trình duyệt để tìm và sửa lỗi trong mã HTML, CSS, JS.
- Version Control (Quản lý phiên bản): Thành thạo sử dụng Git và các nền tảng như [GitHub](#), [GitLab](#) để quản lý mã nguồn, theo dõi lịch sử thay đổi và cộng tác hiệu quả trong nhóm.
- Web Performance Optimization (Tối ưu hiệu năng): Hiểu biết về các kỹ thuật giúp tăng tốc độ tải trang và cải thiện trải nghiệm người dùng (tối ưu ảnh, lazy loading, code splitting...).
- Cross-Browser Compatibility (Tương thích trình duyệt): Đảm bảo trang web hoạt động và hiển thị nhất quán trên các trình duyệt phổ biến (Chrome, Firefox, Safari, Edge...).
- Web Accessibility (Khả năng truy cập): Xây dựng trang web sao cho người khuyết tật cũng có thể sử dụng dễ dàng (tuân thủ các tiêu chuẩn như WCAG).
- Build Tools & Task Runners (Công cụ xây dựng): Biết cách sử dụng các công cụ như [Webpack](#), [Vite](#), [Gulp](#) để tự động hóa các tác vụ lặp đi lặp lại trong quá trình phát triển (ví dụ: [biên dịch](#) code, tối ưu file).

CHƯƠNG 3. PHÂN TÍCH VÀ THIẾT KẾ HỆ THỐNG

3.1. Phát biểu bài toán

+ Hiện nay, đa số các quán cà phê vừa và nhỏ vẫn quản lý hoạt động kinh doanh bằng phương pháp thủ công, như ghi order trên giấy, tính tiền bằng máy tính cầm tay, hoặc tổng hợp doanh thu cuối ngày bằng Excel.

Cách làm này dẫn đến nhiều hạn chế:

- Thông tin đơn hàng dễ bị sai sót hoặc thất lạc.
- Khó kiểm soát doanh thu, chi phí và hiệu suất làm việc của nhân viên.
- Việc cập nhật giá món, thêm món mới hoặc thống kê doanh thu tốn nhiều thời gian và không chính xác.

+ Để khắc phục những hạn chế trên, cần xây dựng **một website quản lý quán cà phê**, giúp **tự động hóa quy trình phục vụ và quản lý**, bao gồm các chức năng:

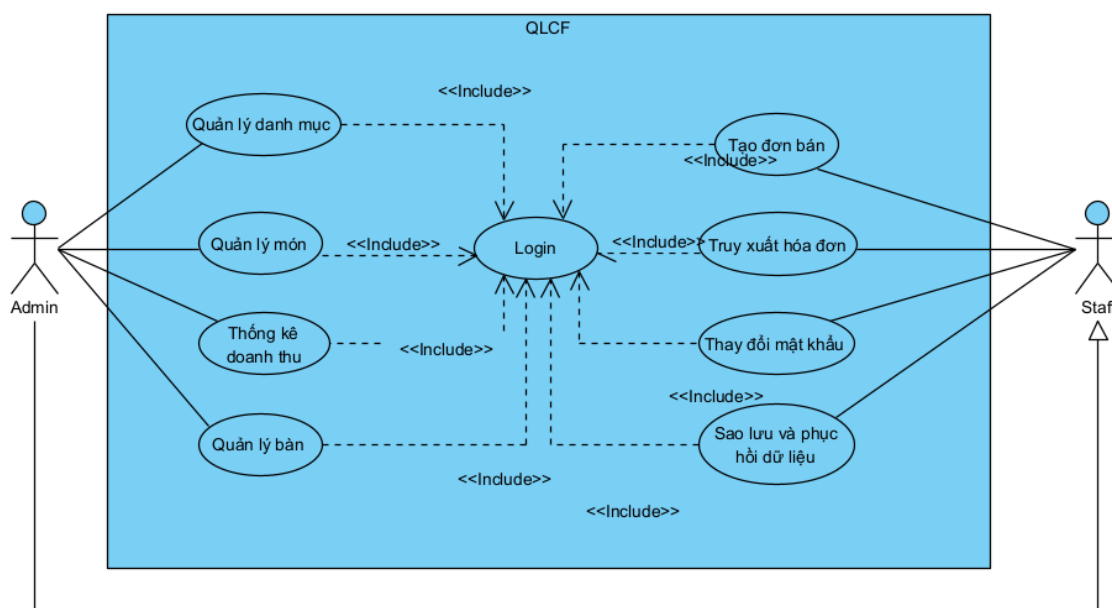
- Quản lý menu món uống và danh mục sản phẩm.
- Quản lý đơn hàng, tạo và in hóa đơn cho khách hàng.
- Theo dõi và thống kê doanh thu theo ngày, tháng, hoặc theo món.
- Phân quyền sử dụng giữa **nhân viên** và **quản lý**.

+ Hệ thống cần được thiết kế theo hướng **web-based**, hoạt động trên trình duyệt, giao diện thân thiện, dễ thao tác, phù hợp cho nhân viên phục vụ và chủ quán. Tất cả dữ liệu (món, nhân viên, hóa đơn, doanh thu) được lưu trữ tập trung trong **cơ sở dữ liệu**, giúp chủ quán dễ dàng kiểm tra, tra cứu và cập nhật.

3.2. Đặc tả yêu cầu phần mềm

3.2.1. Các yêu cầu chức năng

a) Chức năng của phân hệ quản trị và nhân viên.



Hình 3. 1: Usecase tổng quát hệ thống QLCF

1. Use Case: Đăng nhập (Login)

+ Tác nhân: Admin, Staff

+ Mô tả: Cho phép người dùng xác thực tài khoản để truy cập vào hệ thống.

+ Điều kiện đầu vào:

- Hệ thống đang chạy.
- Người dùng có tài khoản hợp lệ.

+ Điều kiện đầu ra: Đăng nhập thành công hoặc hiển thị thông báo lỗi.

+ Luồng chính:

1. Người dùng mở ứng dụng.
2. Hệ thống hiển thị giao diện đăng nhập.
3. Người dùng nhập username và password.
4. Người dùng nhấn nút 'Đăng nhập'.
5. Hệ thống kiểm tra thông tin.

6. Nếu hợp lệ → chuyển vào hệ thống theo quyền.

+ Luồng ngoại lệ:

E1: Username hoặc password bị bỏ trống → Báo lỗi.

E2: Sai mật khẩu → Báo lỗi.

E3: Tài khoản không tồn tại → Báo lỗi.

2. Use Case: Quản lý danh mục

+ Tác nhân: Admin

+ Mô tả: Cho phép admin thêm, sửa, xóa danh mục món.

+ Điều kiện đầu vào: Admin đã đăng nhập.

+ Điều kiện đầu ra: Danh mục được cập nhật vào cơ sở dữ liệu.

+ Luồng chính:

1. Admin chọn chức năng ‘Quản lý danh mục’.

2. Hệ thống hiển thị danh sách danh mục.

3. Admin thêm, sửa hoặc xóa danh mục.

4. Hệ thống lưu thay đổi.

+ Luồng ngoại lệ:

E1: Tên danh mục bị bỏ trống → Báo lỗi.

E2: Tên danh mục đã tồn tại → Báo lỗi.

E3: Danh mục đang được sử dụng → Không cho phép xóa.

3. Use Case: Quản lý món

+ Tác nhân: Admin

+ Mô tả: Admin quản lý món ăn/đồ uống.

+ Điều kiện đầu vào: Admin đã đăng nhập.

+ Điều kiện đầu ra: Món được cập nhật vào cơ sở dữ liệu.

+ Luồng chính:

1. Admin chọn ‘Quản lý món’.
2. Hệ thống hiển thị danh sách món.
3. Admin thêm/sửa/xóa món.
4. Hệ thống lưu và hiển thị kết quả.

+ Luồng ngoại lệ:

- E1: Thiếu thông tin món → Báo lỗi.
- E2: Giá nhập sai định dạng → Báo lỗi.
- E3: Món đang được bán → Không thể xóa.

4. Use Case: Quản lý bàn

+ Tác nhân: Admin

+ Mô tả: Quản lý số bàn và trạng thái bàn.

+ Điều kiện đầu vào: Admin đã đăng nhập.

+ Điều kiện đầu ra: Thông tin bàn được cập nhật.

+ Luồng chính:

1. Admin chọn ‘Quản lý bàn’.
2. Hệ thống hiển thị danh sách bàn.
3. Admin thêm, sửa, xóa bàn.
4. Hệ thống lưu dữ liệu.

+ Luồng ngoại lệ:

- E1: Bàn đang có khách → Không thể xóa.
- E2: Trùng số bàn → Báo lỗi.

5. Use Case: Thống kê doanh thu

+ Tác nhân: Admin

+ Mô tả: Thống kê doanh thu theo ngày/tháng/năm.

+ Điều kiện đầu vào: Admin đã đăng nhập.

+ Điều kiện đầu ra: Thống kê được hiển thị.

+ Luồng chính:

1. Admin chọn chức năng thống kê.

2. Nhập thời gian cần thống kê.

3. Hệ thống truy xuất dữ liệu.

4. Hiển thị báo cáo doanh thu.

+ Luồng ngoại lệ:

E1: Khoảng thời gian không hợp lệ.

E2: Không có dữ liệu → Hiển thị thông báo.

6. Use Case: Tạo đơn bán

+ Tác nhân: Admin, Staff

+ Mô tả: Nhân viên và quản lý tạo hóa đơn bán cho khách.

+ Điều kiện đầu vào: Đã đăng nhập.

+ Điều kiện đầu ra: Hóa đơn được lưu vào CSDL.

+ Luồng chính:

1. Staff chọn bàn cần phục vụ.

2. Chọn món và số lượng.

3. Hệ thống tính tổng tiền.

4. Staff xác nhận thanh toán.

5. Hệ thống lưu hóa đơn.

+ Luồng ngoại lệ:

E1: Số lượng không hợp lệ.

E2: Món hết hàng.

E3: Bàn không tồn tại.

7. Use Case: Truy xuất hóa đơn

+ Tác nhân: Admin, Staff

+ Mô tả: Xem lại các hóa đơn đã tạo.

+ Điều kiện đầu vào: Đã đăng nhập.

+ Điều kiện đầu ra: Thông tin hóa đơn được hiển thị.

+ Luồng chính:

1. Staff mở chức năng xem hóa đơn.
2. Hệ thống hiển thị danh sách hóa đơn.
3. Staff chọn hóa đơn để xem chi tiết.
4. Hệ thống hiển thị nội dung hóa đơn.

+ Luồng ngoại lệ:

E1: Hóa đơn không tồn tại.

E2: Lỗi dữ liệu.

8. Use Case: Thay đổi mật khẩu

+ Tác nhân: Admin, Staff

+ Mô tả: Quản lý và nhân viên thay đổi mật khẩu tài khoản.

+ Điều kiện đầu vào: Đã đăng nhập.

+ Điều kiện đầu ra: Mật khẩu mới được cập nhật.

+ Luồng chính:

1. Staff chọn 'Đổi mật khẩu'.
2. Nhập mật khẩu cũ.
3. Nhập mật khẩu mới và xác nhận.

4. Hệ thống cập nhật mật khẩu.

+ Luồng ngoại lệ:

E1: Mật khẩu cũ không đúng.

E2: Mật khẩu mới trùng mật khẩu cũ.

E3: Mật khẩu xác nhận không khớp.

9. Use Case: Sao lưu và phục hồi dữ liệu

+ Tác nhân: Admin, Staff

+ Mô tả: Quản trị viên sao lưu và phục hồi cơ sở dữ liệu.

+ Điều kiện đầu vào: Đã đăng nhập

+ Điều kiện đầu ra: File sao lưu được tạo hoặc dữ liệu được phục hồi.

+ Luồng chính:

1. Admin chọn chức năng sao lưu/phục hồi.

2. Chọn thư mục lưu file hoặc file phục hồi.

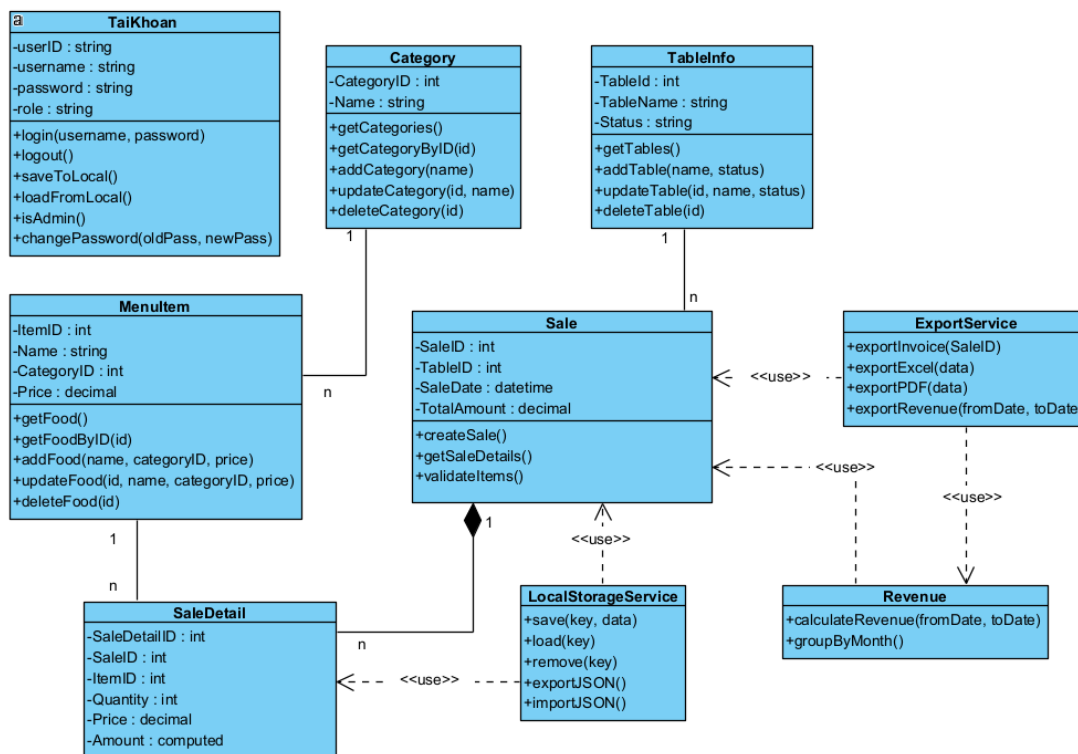
3. Hệ thống tiến hành xử lý.

4. Hiển thị kết quả.

+ Luồng ngoại lệ:

E1: File sao lưu lỗi hoặc không hợp lệ.

3.2.2. Biểu đồ lớp thực thể



Hình 3. 2: Biểu đồ lớp thực thể

1) Category 1 – n MenuItem

+ **Kiểu quan hệ:** Association (1-to-many).

+ **Biểu diễn dữ liệu (localStorage):** Mỗi MenuItem lưu thuộc tính categoryID tham chiếu đến Category.categoryID.

+ **Ý nghĩa:**

- Một Category có thể chứa nhiều MenuItem.
- Mỗi MenuItem chỉ thuộc về **một Category xác định**.

+ **Hành vi khi thao tác:**

- Thêm Category: không ảnh hưởng ngay đến các MenuItem.
- Xóa Category: cần xử lý logic ở phía frontend:
 - Không cho phép xóa nếu còn MenuItem liên quan (khuyến nghị).

- Hoặc chuyển các món sang Category mặc định (ví dụ: “Khác”).

2) TableInfo 1 – n Sale

+ **Kiểu quan hệ:** Association (1-to-many).

+ **Biểu diễn dữ liệu:** Sale.tableID tham chiếu đến TableInfo.tableID.

+ **Ý nghĩa:**

- Một bàn có thể phát sinh **nhiều hóa đơn** tại các thời điểm khác nhau.
- TableInfo.status biểu diễn **trạng thái hiện tại của bàn** (Trống / Đang phục vụ).

+ **Hành vi nghiệp vụ:**

- Khi người dùng chọn bàn và bắt đầu order:
- Tạo một Sale mới.
- Cập nhật trạng thái bàn = *Đã có người*.
- Khi thanh toán:
- Hoàn tất hóa đơn (tính tổng tiền).
- Cập nhật trạng thái bàn = *Trống*.

3) Sale 1 – n SaleDetail (Composition)

+ **Kiểu quan hệ:**

- Composition (quan hệ hợp thành).

+ **Biểu diễn dữ liệu:**

- SaleDetail.saleID tham chiếu đến Sale.saleID.

+ **Ý nghĩa:**

- SaleDetail **không thể tồn tại độc lập** nếu không có Sale.
- Một hóa đơn được cấu thành từ nhiều dòng chi tiết món.

+ **Hành vi:**

- Khi tạo hóa đơn:
- Tạo Sale.
- Tạo danh sách SaleDetail tương ứng.
- Khi xóa hoặc hoàn tất Sale:
- Toàn bộ SaleDetail liên quan được xóa khỏi localStorage.

+ **Tính toán:**

- TotalAmount = SUM (SaleDetail.price × quantity).

4) MenuItem 1 – n SaleDetail

+ **Kiểu quan hệ:** Association (1-to-many).

+ **Biểu diễn dữ liệu:** SaleDetail.itemID tham chiếu đến MenuItem.itemID.

+ **Ý nghĩa:**

- Một món có thể xuất hiện trong **nhiều hóa đơn khác nhau**.
- SaleDetail.price lưu **giá tại thời điểm bán** để đảm bảo tính toàn vẹn lịch sử.

+ **Hành vi:**

- Thay đổi giá MenuItem **không ảnh hưởng** đến các hóa đơn đã thanh toán.
- Không nên xóa MenuItem đã từng bán → chỉ nên ẩn hoặc ngừng kinh doanh (soft delete).

5) Revenue → Sale (Dependency / Uses)

+ **Kiểu quan hệ:** Dependency (Uses).

+ **Ý nghĩa:**

- RevenueService là **lớp xử lý nghiệp vụ thống kê**, chỉ **đọc dữ liệu Sale** từ localStorage.
- Không sở hữu dữ liệu và không ràng buộc cấu trúc.

+ **Chức năng:**

- Lọc hóa đơn theo khoảng thời gian.
- Tính tổng doanh thu.
- Gom nhóm doanh thu theo ngày / tháng để phục vụ biểu đồ.

6) BackupManager & ExportService

+ **Kiểu quan hệ:** Service độc lập.

+ **Ý nghĩa:**

- **LocalStorageService**

- Chịu trách nhiệm lưu trữ, đọc và quản lý toàn bộ dữ liệu của hệ thống trên trình duyệt thông qua localStorage.
- Thực hiện chức năng sao lưu và phục hồi dữ liệu bằng cách xuất / nhập dữ liệu dưới dạng file JSON.
- Không phụ thuộc vào thực thể cụ thể, đóng vai trò như một tầng lưu trữ cục bộ thay thế cho cơ sở dữ liệu phía backend.

- **ExportService**

- Thực hiện chức năng xuất dữ liệu hóa đơn và doanh thu ra các định dạng như Excel hoặc PDF.
- Chỉ đọc dữ liệu từ hệ thống (thông qua LocalStorageService hoặc dữ liệu đã được xử lý sẵn).
- Không can thiệp vào nghiệp vụ bán hàng và không ràng buộc vòng đời với các thực thể như Sale hay SaleDetail.

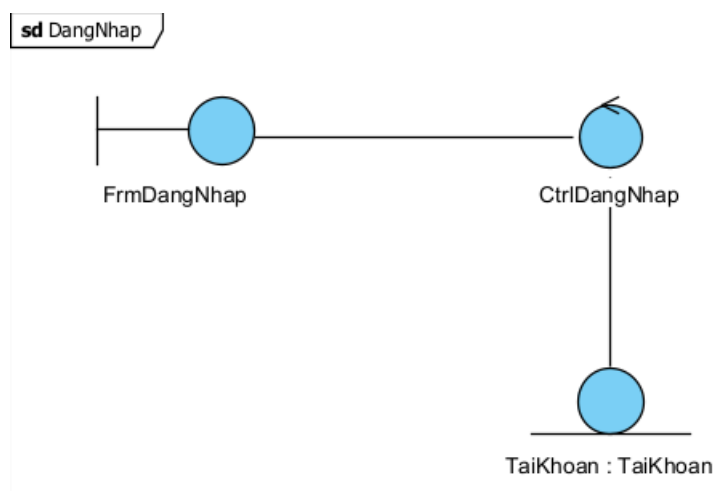
7) TaiKhoan (Account - LocalStorage)

+ Kiểu quan hệ: Independent class.

+ Ý nghĩa:

- Lưu phiên người dùng trên client.
- Không liên quan tới DB.
- Dùng để phân quyền UI.

3.2.3. Biểu đồ phân tích ca sử dụng

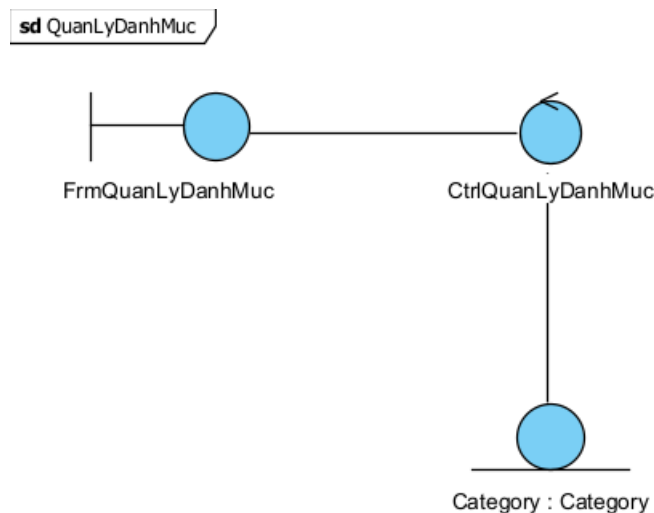


Hình 3. 3: Biểu đồ VOPC cho ca sử dụng Đăng nhập

+ Trong ca sử dụng Đăng nhập, các lớp tham gia bao gồm:

- **Boundary (frmLogin):** Là lớp giao diện cho phép người dùng nhập tên đăng nhập và mật khẩu.
- **Control (CtrlDangNhap):** Tiếp nhận yêu cầu đăng nhập từ giao diện, xử lý logic xác thực thông tin người dùng.
- **Entity (TaiKhoan):** Đại diện cho tài khoản trong localStorage, lưu trữ thông tin đăng nhập và quyền người dùng.

+ **Quy trình xử lý:** Người dùng nhập thông tin đăng nhập vào giao diện. Dữ liệu được gửi đến lớp điều khiển để kiểm tra thông tin trong cơ sở dữ liệu thông qua lớp thực thể. Kết quả xác thực được trả về giao diện để thông báo cho người dùng.

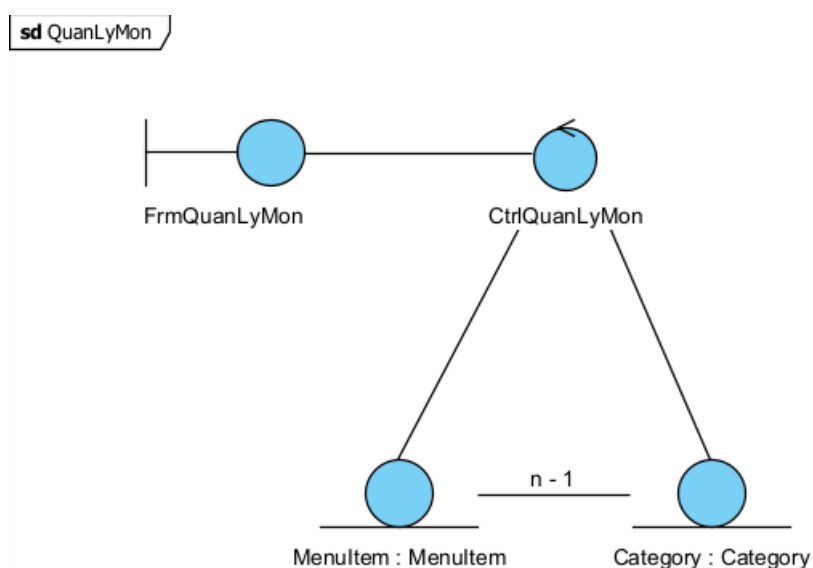


Hình 3. 4: Biểu đồ VOPC cho ca sử dụng Quản lý danh mục

+ Trong ca sử dụng Quản lý danh mục, các lớp tham gia bao gồm:

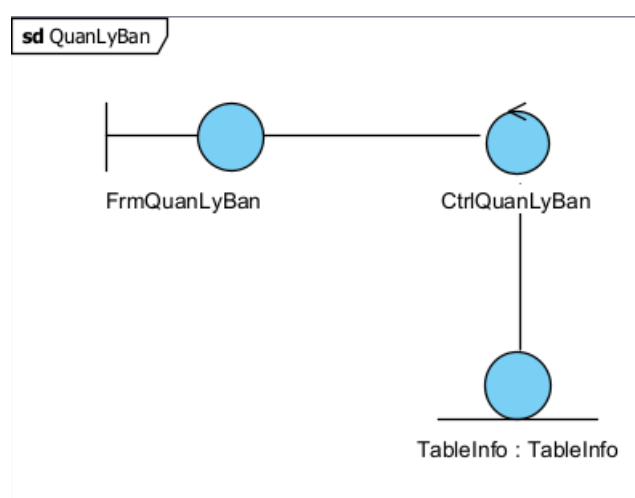
- **Boundary (frmQuanLyDanhMuc):** Cung cấp giao diện quản lý danh mục
- **Control (CtrlQuanLyDanhMuc):** Xử lý nghiệp vụ thêm, sửa, xóa và tìm kiếm.
- **Entity (Catagory):** Lưu trữ thông tin các danh mục.

+ **Quy trình xử lý:** Thông tin từ giao diện được chuyển đến lớp điều khiển, sau đó cập nhật dữ liệu thông qua các lớp thực thể tương ứng.

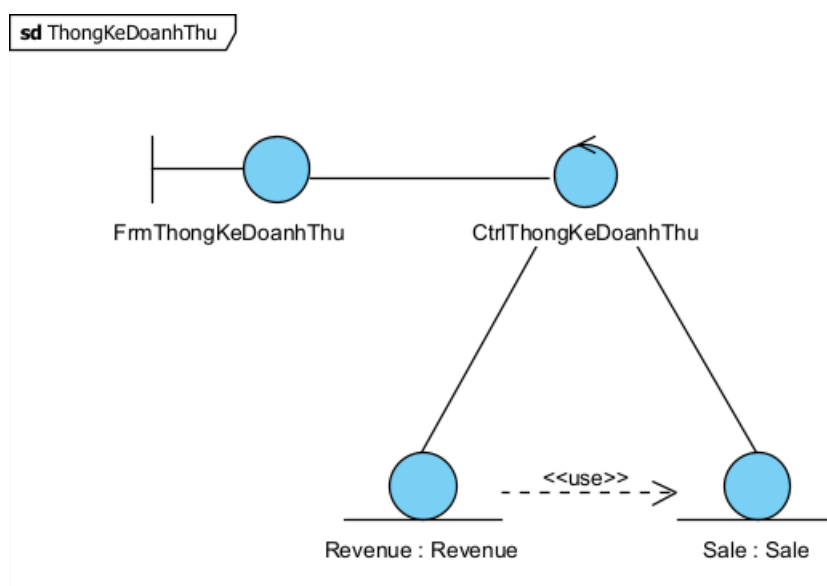


Hình 3. 5: Biểu đồ VOPC cho ca sử dụng Quản lý món

- **Boundary (frmQuanLyMon):** Cung cấp giao diện quản lý món.
- **Control (CtrlQuanLyMon):** Xử lý nghiệp vụ thêm, sửa, xóa và tìm kiếm.
- **Entity (MenuItem, Category):** Đại diện cho bảng Category, MenuItem trong cơ sở dữ liệu, lưu trữ thông tin món.



- + Trong ca sử dụng Quản lý bàn, các lớp tham gia bao gồm:
 - **Boundary (frmQuanLyBan):** Cung cấp giao diện quản lý bàn.
 - **Control (CtrlQuanLyBan):** Xử lý nghiệp vụ thêm, sửa, xóa.
 - **Entity (TableInfo):** Đại diện cho bảng TableInfo trong cơ sở dữ liệu, lưu trữ thông tin các bàn.

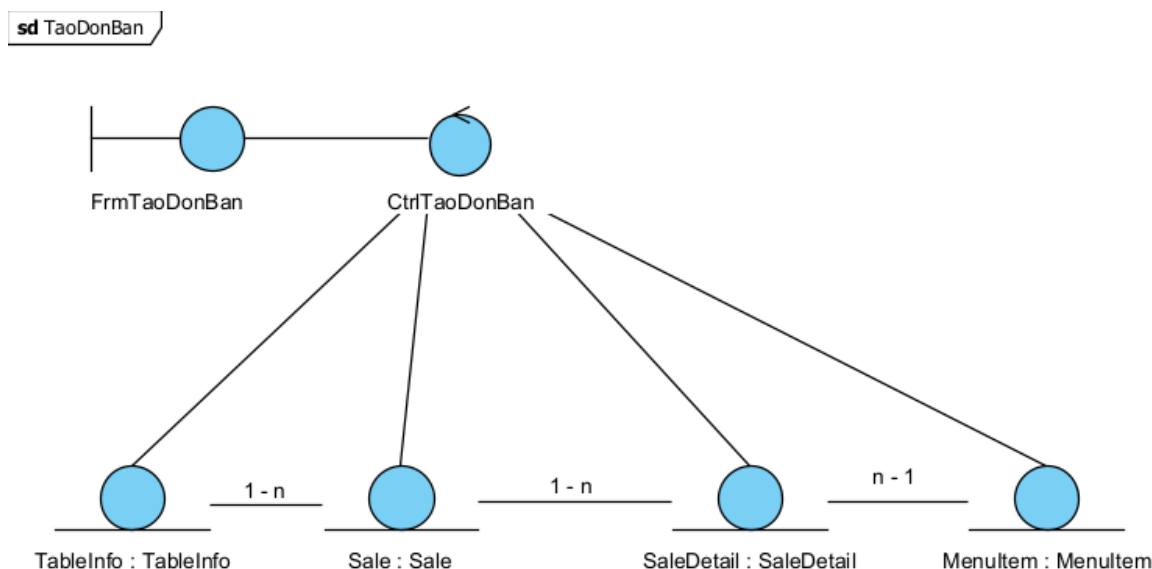


Hình 3. 7: Biểu đồ VOPC cho ca sử dụng Thống kê doanh thu

+ Trong ca sử dụng Thống kê doanh thu, các lớp tham gia bao gồm:

- **Boundary (frmThongKeDoanhThu):** Cung cấp giao diện thống kê doanh thu.
- **Control (CtrlThongKeDoanhThu):** Xử lý các yêu cầu thống kê theo thời gian.
- **Entity (Revenue, Sale):** Cung cấp dữ liệu hóa đơn để tính toán doanh thu.

+ **Quy trình xử lý:** Người dùng chọn khoảng thời gian cần thống kê, hệ thống truy vấn dữ liệu và hiển thị kết quả.

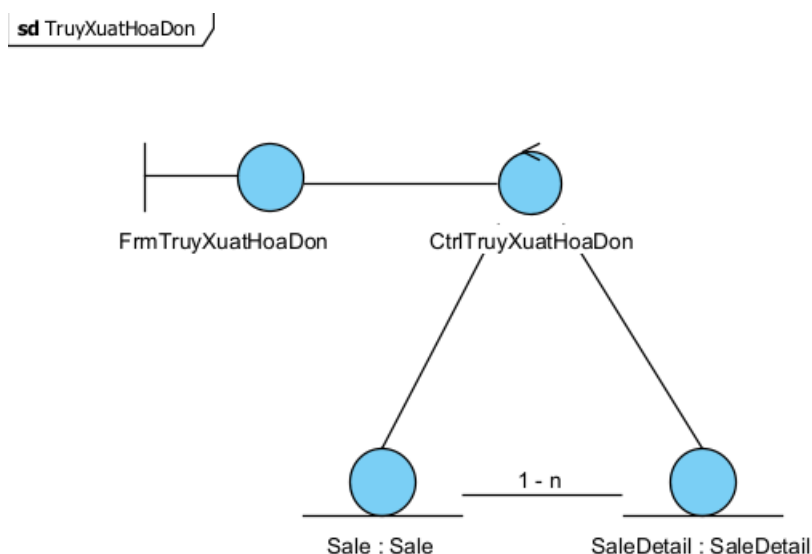


Hình 3. 8: Biểu đồ VOPC cho ca sử dụng Tạo hóa đơn bán

+ Trong ca sử dụng Tạo đơn bán, các lớp tham gia bao gồm:

- **Boundary (frmTaoDonBan):** Cung cấp giao diện tạo đơn bán.
- **Control (CtrlTaoDonBan):** Xử lý nghiệp vụ lập hóa đơn, thêm món, thanh toán, cập nhật trạng thái bàn.
- **Entity (TableInfo, Sale, SaleDetail, MenuItem):** Lưu trữ thông tin hóa đơn và chi tiết hóa đơn.

+ **Quy trình xử lý:** Khi người dùng thêm món và thanh toán, hệ thống tạo hóa đơn, lưu chi tiết hóa đơn và cập nhật trạng thái bàn tương ứng.

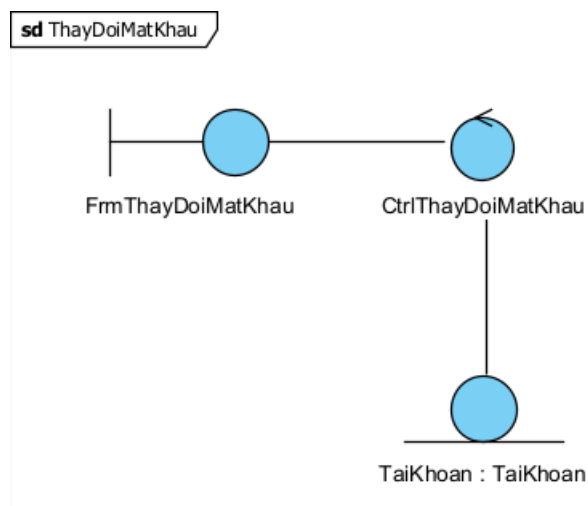


Hình 3. 9: Biểu đồ VOPC cho ca sử dụng Truy xuất hóa đơn

+ Trong ca sử dụng Truy xuất hóa đơn, các lớp tham gia bao gồm:

- **Boundary (frmTruyXuatHoaDon):** Cung cấp giao diện truy xuất hóa đơn.
- **Control (CtrlTruyXuatHoaDon):** Xử lý nghiệp vụ tìm kiếm chi tiết hóa đơn dựa vào mã hóa đơn tương ứng
- **Entity (Sale, SaleDetail):** Cung cấp dữ liệu để truy xuất chi tiết hóa đơn.

+ **Quy trình xử lý:** Khi người dùng nhập mã hóa đơn, sẽ hiển thị thông tin chi tiết hóa đơn đó ra bảng.

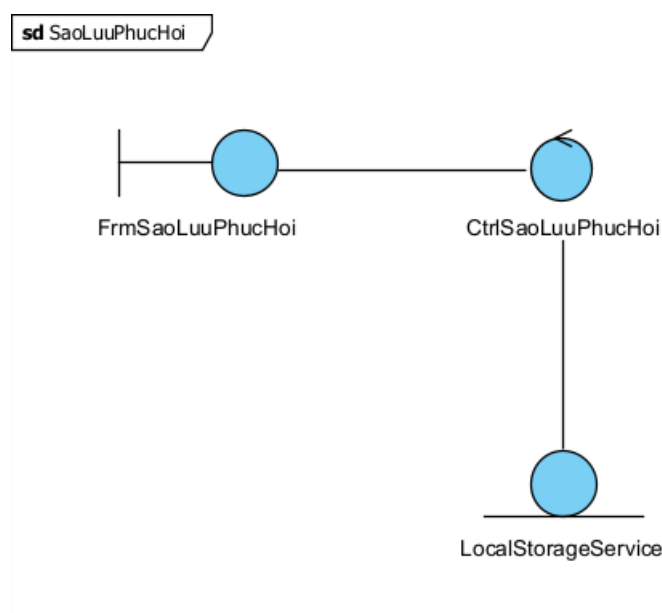


Hình 3. 10: Biểu đồ VOPC cho ca sử dụng Thay đổi mật khẩu

+ Trong ca sử dụng Thay đổi mật khẩu, các lớp tham gia bao gồm:

- **Boundary (frmThayDoiMatKhau):** Cung cấp giao diện thay đổi mật khẩu.
- **Control (CtrlThayDoimatKhau):** Xử lý nghiệp vụ cập nhật mật khẩu mới.
- **Entity (TaiKhoan):** Cung cấp dữ liệu các tài khoản trong localStorage.

+ **Quy trình xử lý:** Sau khi đăng nhập với quyền tương ứng, người dùng chọn chức năng thay đổi mật khẩu → nhập mật khẩu cũ, mật khẩu mới và xác nhận mật khẩu mới. Nếu mật khẩu cũ chính xác, mật khẩu sẽ được cập nhật. Ngược lại sẽ báo lỗi



Hình 3. 11: Biểu đồ VOPC cho ca sử dụng Sao lưu và phục hồi dữ liệu

+ Trong ca sử dụng Sao lưu và phục hồi dữ liệu, các lớp tham gia bao gồm:

- **Boundary (frmSaoLuuPhucHoi):** Cung cấp giao diện sao lưu và phục hồi dữ liệu.
- **Control (CtrlSaoLuuPhucHoi):** Xử lý nghiệp vụ sao lưu dữ liệu vào folder backup và phục hồi dữ liệu từ các file backup.
- **Entity (BackupManager):** Đại diện cho dữ liệu trong localstorage.

+ **Quy trình xử lý:** Sau khi đăng nhập, người dùng truy cập chức năng sao lưu và khôi phục. Nhấn nút sao lưu, hệ thống sẽ tiến hành lưu dữ liệu vào file (.bak) về máy cục bộ. Chọn file (.bak) cần khôi phục → Hệ thống gửi yêu cầu khôi phục

3.2.4. Các yêu cầu phi chức năng

+ **Hiệu năng (Performance)**

- Hệ thống phải xử lý các yêu cầu như đăng nhập, thêm món, thanh toán trong vòng < 3 giây.
- Website phải có khả năng đáp ứng tối thiểu 50 người dùng truy cập đồng thời mà không xảy ra treo hệ thống.
- Tốc độ tải trang trung bình không quá 5 giây đối với mạng Internet thông thường.

+ **Tính bảo mật (Security)**

- Mỗi người dùng (nhân viên, quản lý) phải đăng nhập bằng tên đăng nhập và mật khẩu riêng.
- Hệ thống phân quyền rõ ràng:
 - Quản lý có quyền thống kê, quản lý món, bàn, danh mục, hóa đơn.
 - Nhân viên chỉ có quyền bán hàng và xử lý hóa đơn.

+ Tính khả dụng (Usability)

- Giao diện được thiết kế thân thiện, dễ hiểu, có hướng dẫn sử dụng ở từng chức năng chính.
- Màu sắc, bố cục phù hợp với chủ đề quán cà phê, tạo cảm giác gần gũi cho người dùng.
- Người dùng có thể thao tác chỉ với vài cú nhấp chuột để hoàn thành tác vụ (ví dụ: thanh toán, thêm món).

+ Khả năng bảo trì và mở rộng (Maintainability & Scalability)

- Mã nguồn được tổ chức theo mô hình phân lớp (MVC hoặc 3-layer) giúp dễ dàng chỉnh sửa, mở rộng.
- Hệ thống có thể mở rộng thêm các chức năng khác như đặt hàng online, quản lý kho, nhập hàng mà không cần thay đổi cấu trúc hiện tại quá nhiều.
- CSDL thiết kế có tính linh hoạt, cho phép bổ sung bảng dữ liệu mới khi cần.

+ Tính tin cậy (Reliability)

- Dữ liệu được lưu trữ an toàn, tránh mất mát khi hệ thống gặp sự cố.
- Có chức năng sao lưu và phục hồi dữ liệu (backup/restore) định kỳ.
- Khi lỗi xảy ra, hệ thống hiển thị thông báo rõ ràng và không làm gián đoạn toàn bộ hoạt động.

+ Tính tương thích (Compatibility)

- Website có thể hoạt động tốt trên các trình duyệt phổ biến như Chrome, Edge, Firefox.
- Hệ thống tương thích với các thiết bị máy tính, máy POS, laptop, máy tính bảng.
- Hỗ trợ ngôn ngữ tiếng Việt đầy đủ và hiển thị chuẩn trên các hệ điều hành phổ biến (Windows, macOS, Linux).

+ Yêu cầu về sao lưu dữ liệu (Data Backup)

- Dữ liệu phải được sao lưu tự động mỗi ngày hoặc khi quản trị viên yêu cầu.
- File sao lưu được lưu trữ an toàn, có thể phục hồi lại toàn bộ hệ thống khi xảy ra sự cố.

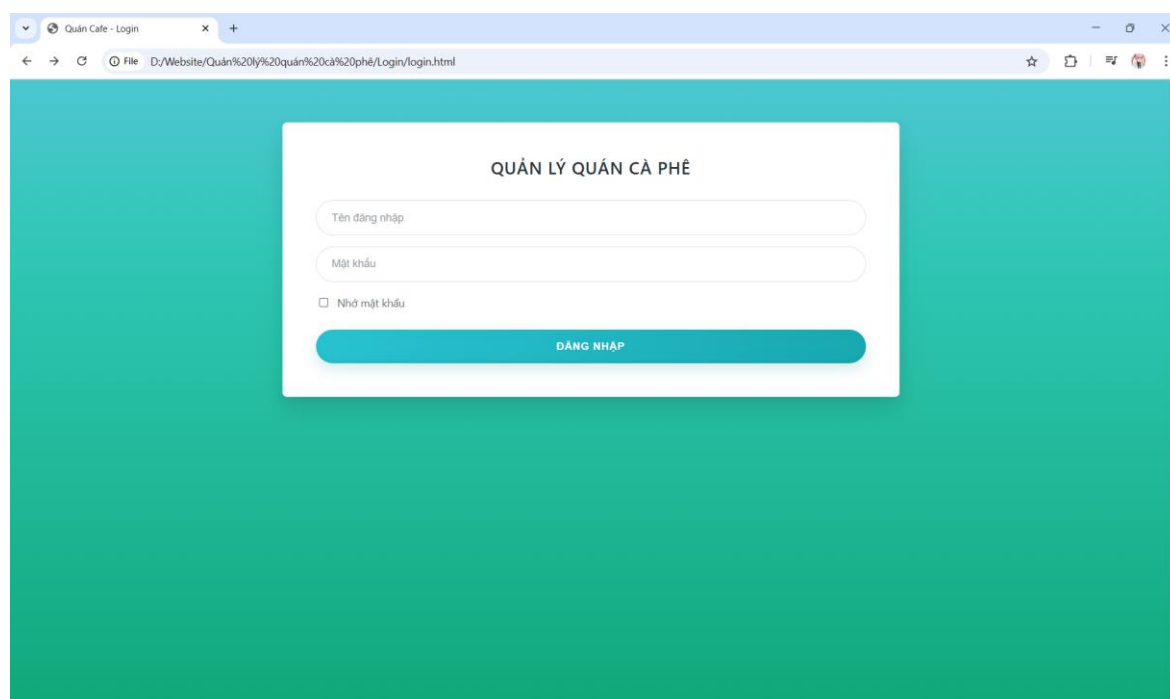
+ Tính toàn vẹn dữ liệu (Data Integrity)

- Không cho phép nhập dữ liệu thiếu hoặc trùng khóa chính (ví dụ: mã món, mã bàn, mã tài khoản).
- Các thao tác cập nhật, xóa, thêm dữ liệu phải được kiểm tra tính hợp lệ trước khi lưu vào cơ sở dữ liệu.

3.3. Thiết kế giao diện

3.3.1. Thiết kế giao diện cho phân hệ quản trị và nhân viên

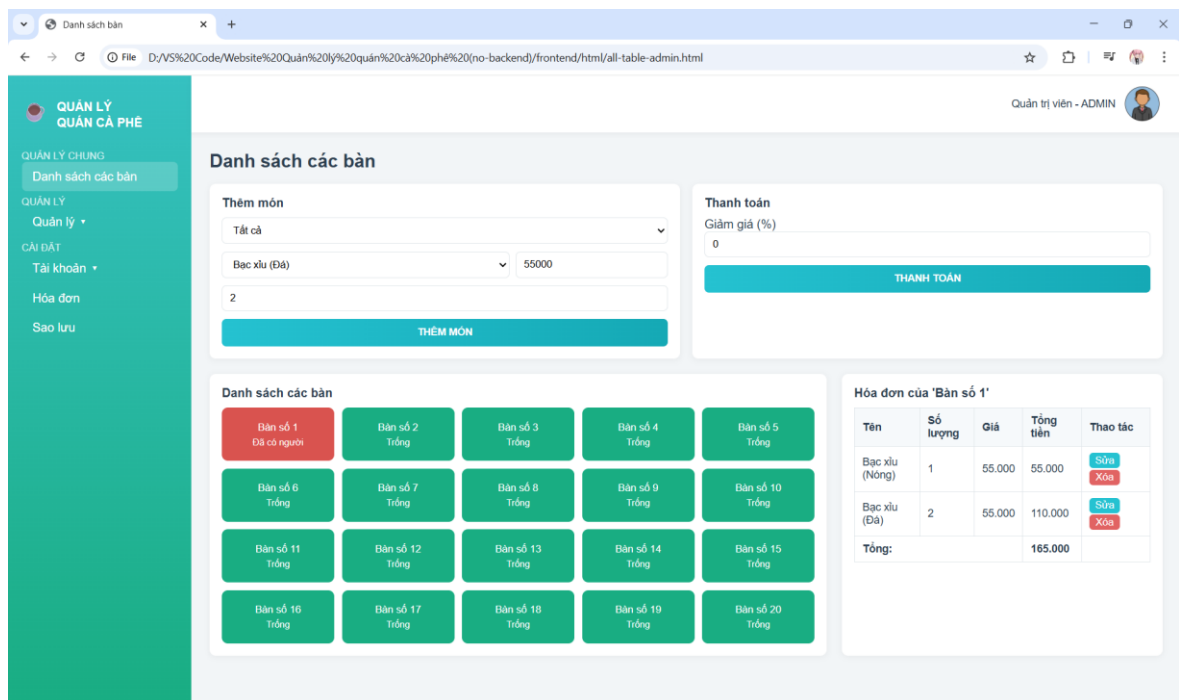
+ Giao diện trang đăng nhập



Hình 3. 12: Giao diện trang Đăng nhập

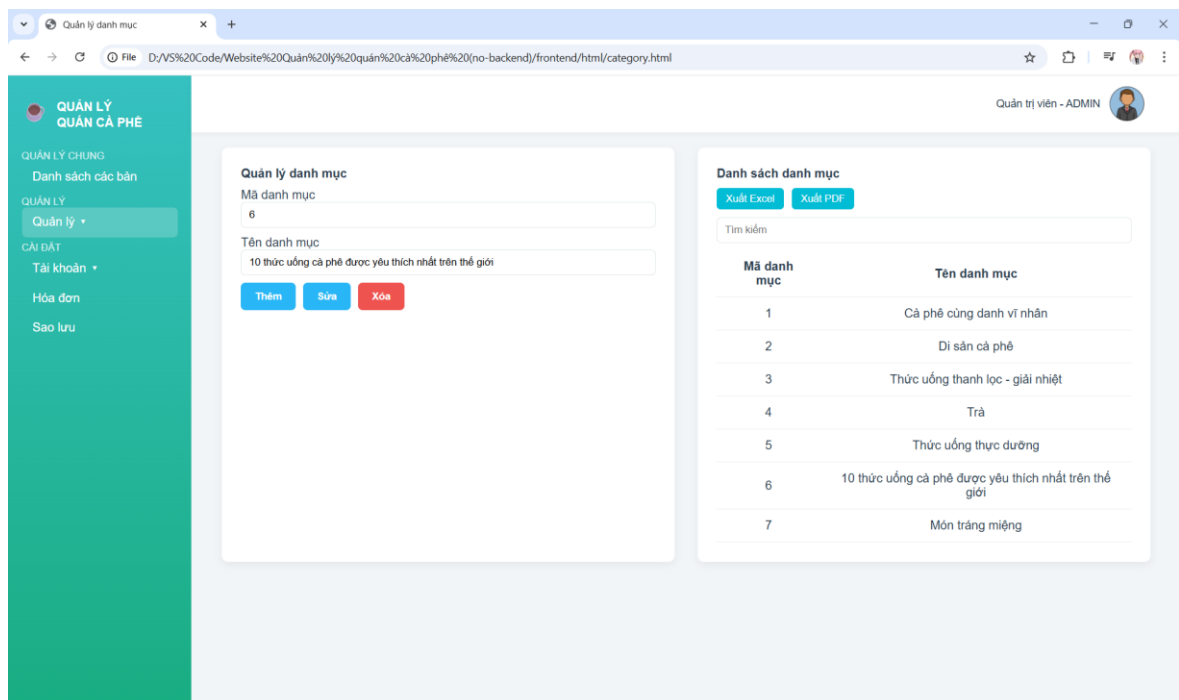
+ Giao diện trang quản lý chung (Danh sách bàn)

Đồ án 2: Thiết kế website quản lý quán cà phê



Hình 3. 13: Giao diện trang Danh sách bàn

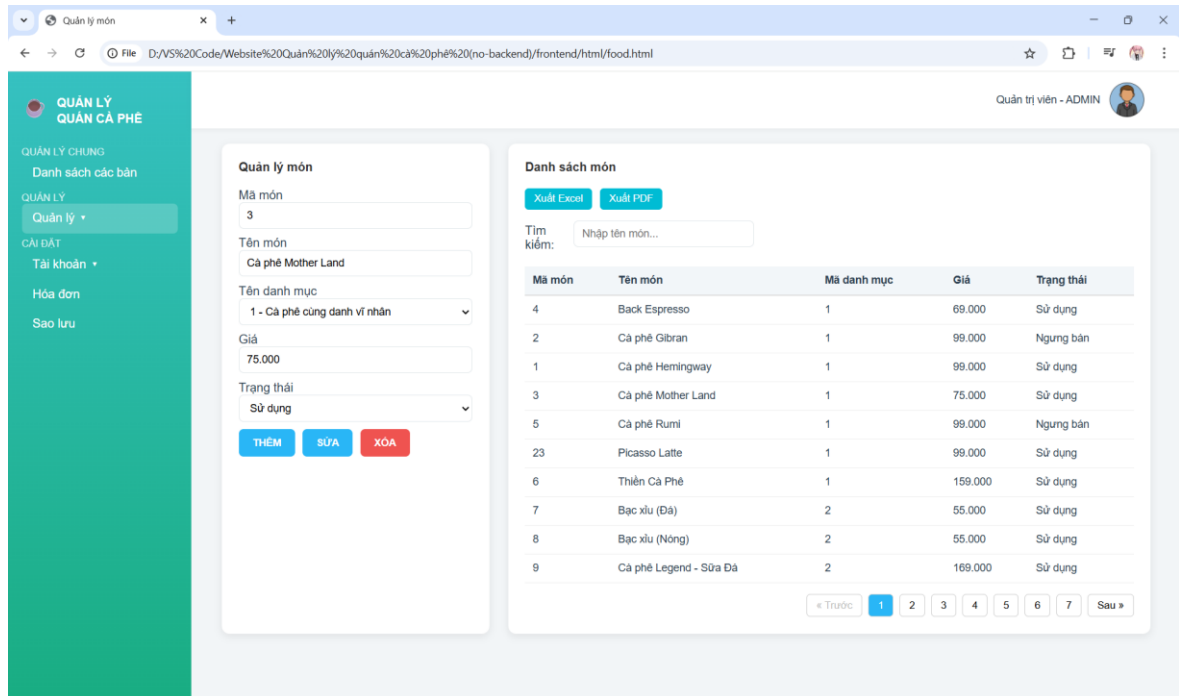
+ Giao diện trang quản lý danh mục



Hình 3. 14: Giao diện trang Quản lý danh mục

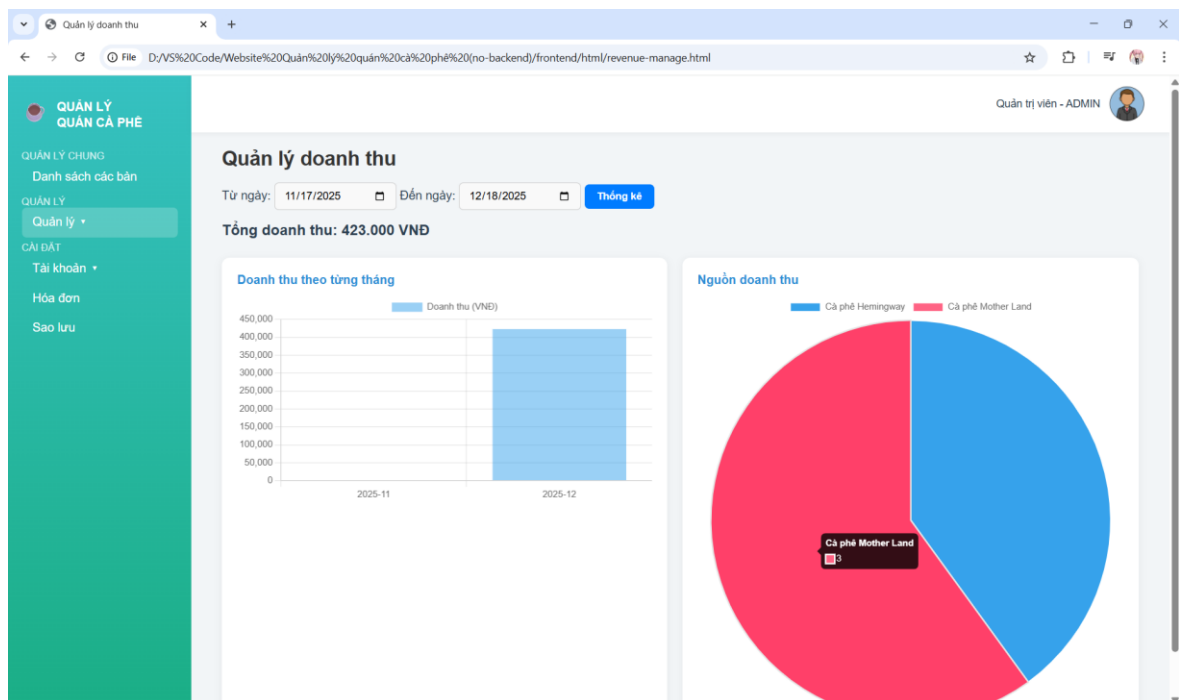
+ Giao diện trang quản lý món

Đồ án 2: Thiết kế website quản lý quán cà phê



Hình 3. 15: Giao diện trang Quản lý món

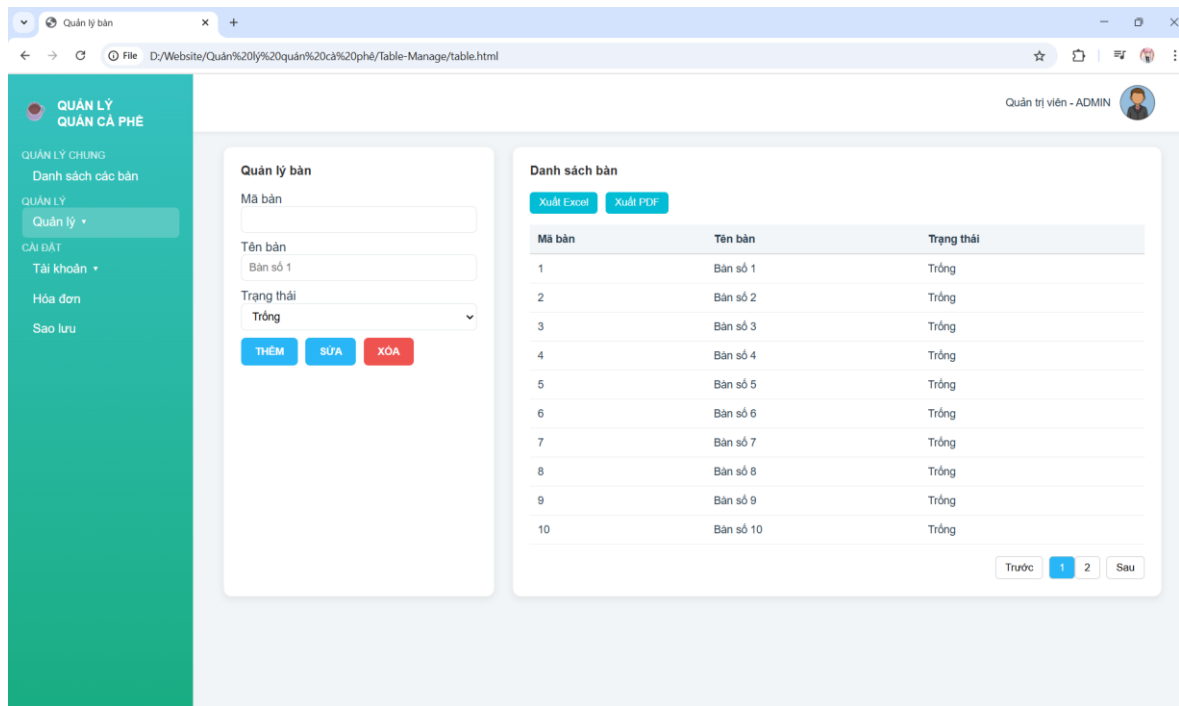
+ Giao diện trang thống kê doanh thu



Hình 3. 16: Giao diện trang Thống kê doanh thu

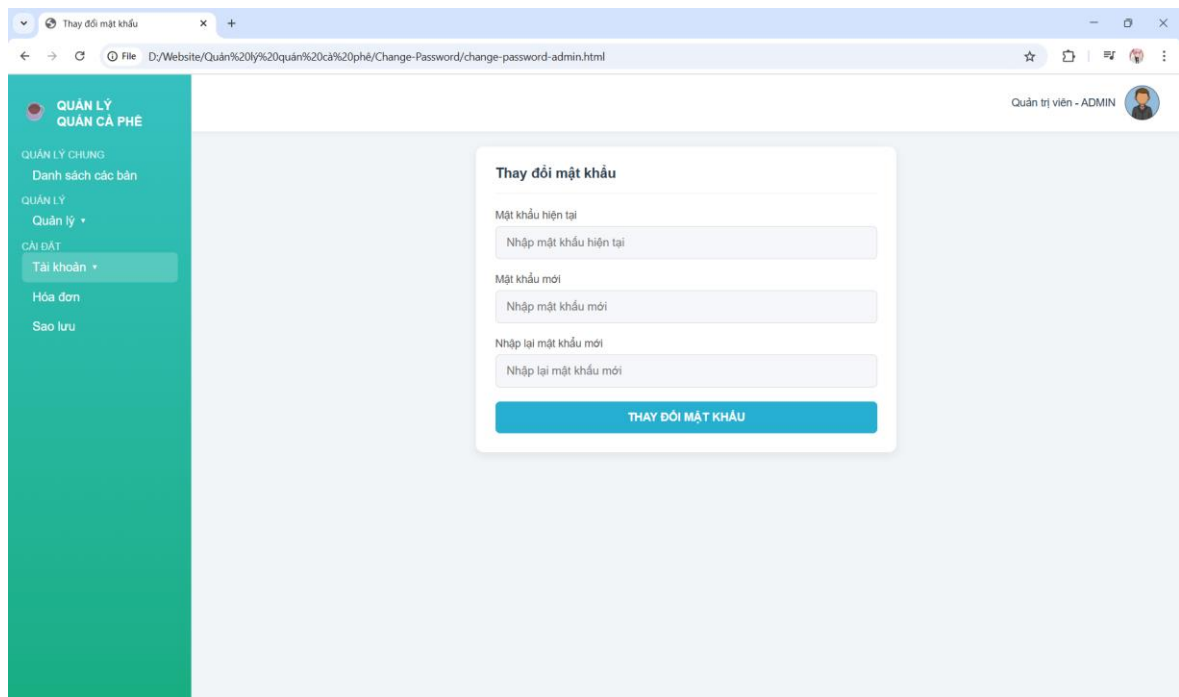
+ Giao diện trang quản lý bàn

Đồ án 2: Thiết kế website quản lý quán cà phê



Hình 3. 17: Giao diện trang Quản lý bàn

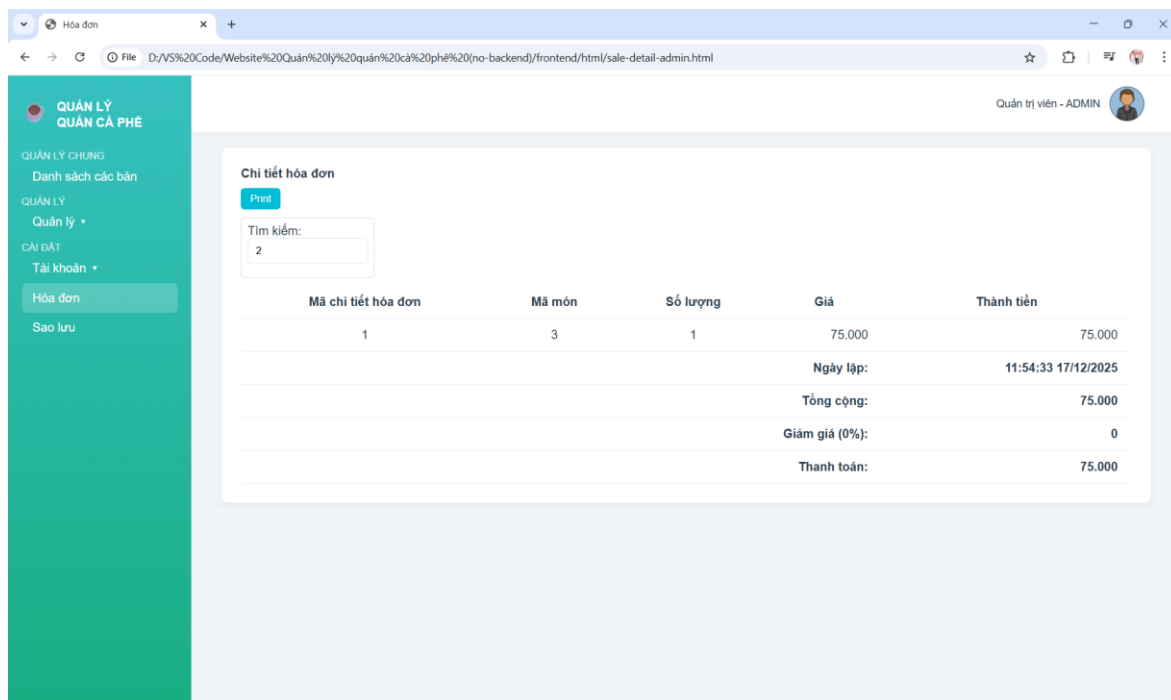
+ Giao diện trang đổi mật khẩu tài khoản



Hình 3. 18: Giao diện trang Thay đổi mật khẩu

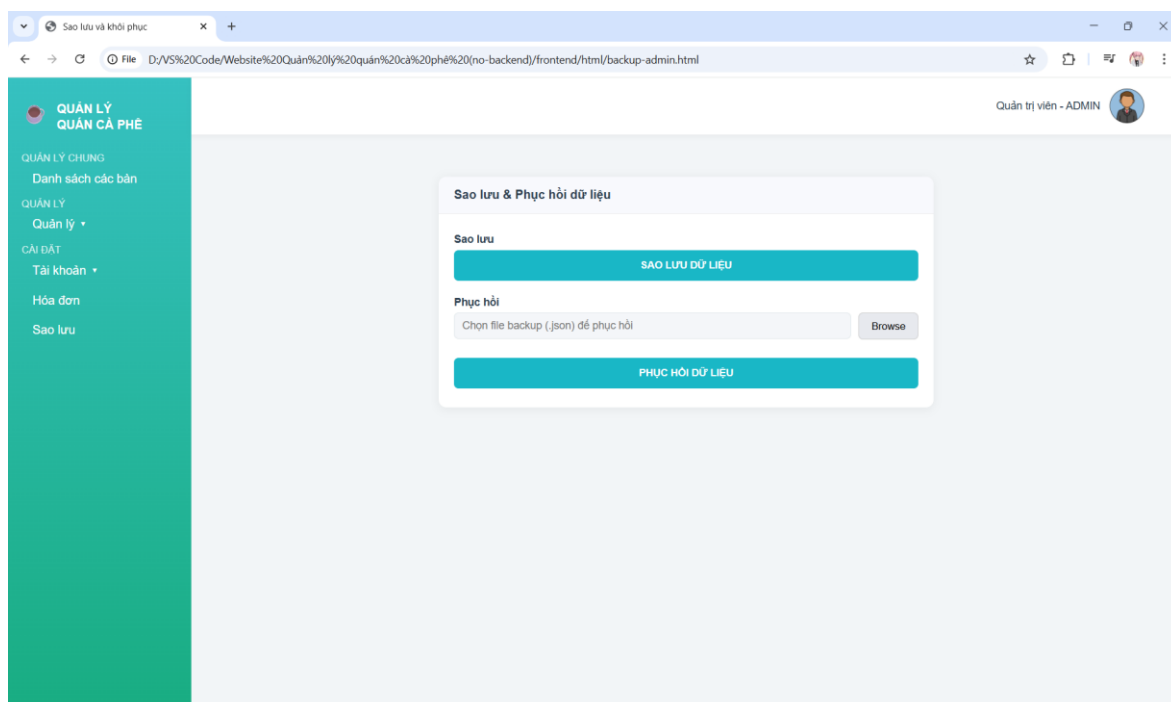
+ Giao diện trang truy xuất hóa đơn theo mã hóa đơn

Đồ án 2: Thiết kế website quản lý quán cà phê



Hình 3. 19: Giao diện trang Truy xuất hóa đơn bán

+ Giao diện trang sao lưu và phục hồi dữ liệu



Hình 3. 20: Giao diện trang Sao lưu và khôi phục dữ liệu

CHƯƠNG 4. TRIỂN KHAI WEBSITE

4.1. Triển khai các chức năng cho phân hệ quản trị nội dung

4.1.1. Trang danh sách bàn

1. Thiết kế bố cục trang Chủ bằng HTML

+ Trang Chủ của hệ thống quản lý quán cà phê (giao diện Admin – Danh sách bàn) được xây dựng bằng HTML5, sử dụng các thẻ ngữ nghĩa kết hợp với các thẻ khối (div, aside, main) nhằm phân chia bố cục một cách rõ ràng, khoa học và dễ bảo trì.

1.1. Cấu trúc tổng thể của trang

+ Ví dụ mã nguồn minh họa

```
<body>
  <aside class="sidebar">
    <!-- Thanh điều hướng -->
  </aside>

  <main class="main">
    <!-- Nội dung chính -->
  </main>
</body>
```

Hình 4. 1: Ví dụ mã nguồn minh họa (Trang danh sách bàn - html)

+ Toàn bộ giao diện được chia thành hai khu vực chính:

- **Sidebar (Thanh điều hướng):** đặt bên trái màn hình, dùng để hiển thị logo hệ thống và các menu chức năng như quản lý bàn, quản lý danh mục, quản lý món, doanh thu, tài khoản và sao lưu dữ liệu.
- **Main Content (Nội dung chính):** đặt bên phải, hiển thị các chức năng nghiệp vụ chính như chọn bàn, thêm món, xem hóa đơn và thanh toán.

+ Cấu trúc tổng quát của trang được tổ chức như sau:

- Thẻ <aside> dùng cho Sidebar
- Thẻ <main> dùng cho khu vực nội dung chính

➔ Cách tổ chức này giúp giao diện tuân theo chuẩn HTML5, dễ mở rộng và thuận tiện cho việc phát triển về sau.

1.2. Các thành phần trong nội dung chính

+ Khu vực nội dung chính bao gồm các thành phần:

- **Topbar:** hiển thị thông tin người đăng nhập (vai trò Admin) và ảnh đại diện.
- **Khu vực thao tác:** gồm hai khối chức năng là thêm món và thanh toán.
- **Khu vực hiển thị dữ liệu:** bao gồm danh sách các bàn và bảng hóa đơn của bàn đang được chọn.

➔ Các thành phần này được bố trí logic theo luồng thao tác của người dùng, giúp quá trình sử dụng hệ thống diễn ra thuận tiện và trực quan.

2. Kỹ thuật định dạng giao diện bằng CSS

+ Hệ thống sử dụng CSS thuần để định dạng giao diện, đảm bảo tính thẩm mỹ, dễ sử dụng và tương thích với nhiều kích thước màn hình.

+ Ví dụ mã nguồn minh họa (Sử dụng Flexbox cho bố cục chính và Grid cho danh sách bàn)

```
body {  
  display: flex;  
  min-height: 100vh;  
}  
  
.main {  
  flex: 1;  
  display: flex;  
  flex-direction: column;  
}
```

```
.tables {  
  display: grid;  
  grid-template-columns: repeat(auto-fill, minmax(130px, 1fr));  
  gap: 10px;  
}
```

Hình 4. 2: Ví dụ mã nguồn minh họa (Trang danh sách bàn – css)

2.1. Tổ chức bố cục bằng Flexbox và Grid

- **Flexbox** được sử dụng cho bố cục tổng thể của trang và các khối chức năng, giúp các thành phần tự động co giãn theo kích thước màn hình.
- **CSS Grid** được áp dụng cho khu vực hiển thị danh sách bàn, cho phép các bàn được sắp xếp theo dạng lưới và tự động thay đổi số cột.

➔ Cách kết hợp Flexbox và Grid giúp giao diện vừa linh hoạt vừa dễ quản lý.

2.2. Màu sắc và trạng thái hiển thị

+ Mỗi bàn được biểu diễn bằng một nút bấm với màu sắc thể hiện trạng thái:

- Màu xanh: bàn đang trống.
- Màu đỏ: bàn đang có khách hoặc đã có hóa đơn tạm.

➔ Việc sử dụng màu sắc trực quan giúp người quản lý nhanh chóng nắm bắt tình trạng hoạt động của các bàn trong quán.

2.3. Thiết kế giao diện thích ứng (Responsive)

+ Ví dụ code minh họa

```
CSS

@media (max-width: 900px) {
  .sidebar { display: none; }
}
```

Hình 4. 3: Ví dụ mã nguồn minh họa (Responsive)

+ Trang web được thiết kế responsive nhằm đảm bảo hiển thị tốt trên nhiều thiết bị khác nhau. Khi kích thước màn hình nhỏ:

- Sidebar sẽ tự động ẩn.
- Số lượng cột trong danh sách bàn được điều chỉnh phù hợp.

➔ Điều này giúp hệ thống có thể sử dụng linh hoạt trên máy tính bảng hoặc màn hình nhỏ.

3. Lập trình chức năng bằng JavaScript

+ JavaScript được sử dụng để xử lý **toàn bộ các chức năng động của trang**, bao gồm quản lý dữ liệu hóa đơn, xử lý nghiệp vụ bán hàng và cập nhật giao diện người dùng **theo thời gian thực**.

+ Trong phạm vi đồ án, **dữ liệu được lưu trữ cục bộ bằng localStorage**.

+ Ví dụ code minh họa

```
// =====  
// LOCALSTORAGE KEYS  
// =====  
const TABLE_KEY = "tables";  
const CATEGORY_KEY = "categories";  
const FOOD_KEY = "foods";  
const SALE_KEY = "sales";  
const SALE_COUNTER_KEY = "sale_counter"; // 👉 dùng cho ID tăng dần  
  
// =====  
// HELPER  
// =====  
function formatNumber(num) {  
  return Number(num).toLocaleString("vi-VN");  
}  
  
function getData(key) {  
  return JSON.parse(localStorage.getItem(key)) || [];  
}  
  
function saveData(key, data) {  
  localStorage.setItem(key, JSON.stringify(data));  
}  
  
// =====  
// SALE ID (HD1, HD2, HD3...)  
// =====  
function generateSaleID() {  
  let counter = Number(localStorage.getItem(SALE_COUNTER_KEY)) || 0;  
  counter++;  
  localStorage.setItem(SALE_COUNTER_KEY, counter);  
  return "HD" + counter;  
}
```

```
// =====
// BILL TEMP (theo bàn)
// =====
function saveBill(tableID) {
  localStorage.setItem(`bill_${tableID}`, JSON.stringify(currentBill));
}

function loadBill(tableID) {
  return JSON.parse(localStorage.getItem(`bill_${tableID}`)) || [];
}

// =====
// LOAD TABLES
// =====
function loadAllTables() {
  const tables = getData(TABLE_KEY);
  const wrapper = document.querySelector(".tables");
  wrapper.innerHTML = "";

  tables.forEach((t) => {
    const btn = document.createElement("button");
    const tempBill = loadBill(t.TableID);
    const occupied = tempBill.length > 0;

    btn.className = `table ${occupied ? "red" : "green"}`;
    btn.innerHTML = `${t.TableName}<br><small>${
      occupied ? "Đã có người" : "Trống"
    }</small>`;
    btn.dataset.tableId = t.TableID;

    btn.onclick = () => {
      currentTableID = t.TableID;
      currentBill = loadBill(currentTableID);
      renderBill(t.TableName);
    };

    wrapper.appendChild(btn);
  });
}
```

Hình 4. 4: Ví dụ mã nguồn minh họa (Trang danh sách bàn – js)

3.1. Chức năng tải và hiển thị danh sách bàn

- + Khi trang được tải, hệ thống **đọc danh sách các bàn từ localStorage** và hiển thị lên giao diện người dùng.
- + Trạng thái của mỗi bàn (trống hoặc đang có khách) được xác định dựa trên:

- Hóa đơn tạm tương ứng của bàn đó được lưu trong localStorage.
- Nếu hóa đơn tạm tồn tại và có món → bàn đang có khách.
- Nếu không có hóa đơn hoặc hóa đơn rỗng → bàn trống.

3.2. Chức năng chọn bàn và quản lý hóa đơn tạm

+ Mỗi bàn có **một hóa đơn tạm riêng**, được lưu dưới dạng bill_<TableID> trong localStorage.

+ Khi người dùng chọn một bàn:

1. Hệ thống tải hóa đơn tạm tương ứng từ localStorage.
2. Hiện thị danh sách các món đã gọi của bàn.
3. Cập nhật tiêu đề hóa đơn theo tên bàn đang được chọn.

→ Cách tiếp cận này giúp dữ liệu không bị mất khi tải lại trang hoặc khi người dùng chuyển đổi giữa các bàn khác nhau.

3.3. Chức năng thêm, sửa và xóa món

+ Quy trình thêm món vào hóa đơn được thực hiện theo các bước:

1. Kiểm tra người dùng đã chọn bàn hay chưa.
2. Lấy thông tin món ăn, đơn giá và số lượng từ giao diện.
3. Nếu món đã tồn tại trong hóa đơn thì cộng dồn số lượng.
4. Nếu món chưa tồn tại thì thêm mới vào danh sách hóa đơn.
5. Lưu hóa đơn đã cập nhật vào localStorage.
6. Cập nhật lại giao diện hóa đơn và trạng thái của bàn.

+ Ngoài ra, hệ thống cho phép:

- **Chỉnh sửa số lượng món** trong hóa đơn.
- **Xóa món** khỏi hóa đơn.
- Khi hóa đơn không còn món nào, **trạng thái bàn tự động chuyển về trống**.

3.4. Chức năng giảm giá và thanh toán

+ Hệ thống tự động áp dụng chính sách giảm giá dựa trên tổng tiền hóa đơn:

- Tổng tiền lớn hơn **500.000 VNĐ**: giảm **3%**.
- Tổng tiền lớn hơn **1.000.000 VNĐ**: giảm **7%**.

+ Khi thực hiện thanh toán:

1. Hệ thống tính tổng tiền hóa đơn và áp dụng mức giảm giá tương ứng.
2. Hiện thị hộp thoại xác nhận thanh toán cho người dùng.

3. Lưu thông tin hóa đơn đã thanh toán vào localStorage (danh sách hóa đơn).
4. Xóa hóa đơn tạm của bàn đang thanh toán.
5. Cập nhật trạng thái bàn về trống và làm mới giao diện.

→ Chức năng này đảm bảo dữ liệu hóa đơn được xử lý chính xác và nhất quán trong môi trường không sử dụng backend.

4.1.2. Trang đăng nhập

1. Thiết kế bố cục trang đăng nhập bằng HTML

+ Trang đăng nhập được xây dựng bằng HTML5 với cấu trúc đơn giản, tập trung vào chức năng xác thực người dùng. Toàn bộ nội dung được đặt trong một thẻ div có lớp card, đóng vai trò là khu vực chính của trang.

+ Ví dụ mã nguồn minh họa:

```
<div class="card" role="main" aria-labelledby="loginTitle">
  <h1 id="loginTitle">QUẢN LÝ QUÁN CÀ PHÊ</h1>
  <form onsubmit="login(event)">
    <input type="text" placeholder="Tên đăng nhập" required />
    <input type="password" placeholder="Mật khẩu" required />
    <button type="submit">ĐĂNG NHẬP</button>
  </form>
</div>
```

Hình 4. 5: Ví dụ mã nguồn minh họa (Trang đăng nhập – html)

→ Đoạn mã trên thể hiện cách tổ chức giao diện đăng nhập với tiêu đề hệ thống và form nhập liệu, giúp người dùng dễ dàng thao tác.

2. Kỹ thuật định dạng giao diện bằng CSS

+ CSS được sử dụng để tạo giao diện hiện đại với nền gradient và thẻ đăng nhập dạng card nổi bật ở trung tâm màn hình.

+ Ví dụ mã nguồn minh họa:

```
body {  
  background: linear-gradient(180deg, #4cc7d1, #10a879);  
  display: flex;  
  justify-content: center;  
  padding-top: 56px;  
}  
  
.card {  
  background: #fff;  
  border-radius: 6px;  
  padding: 44px;  
  box-shadow: 0 18px 30px rgba(0,0,0,0.12);  
}
```

Hình 4. 6: Ví dụ mã nguồn minh họa (Trang đăng nhập – css)

➔ Cách định dạng này giúp giao diện đăng nhập rõ ràng, dễ nhìn và tạo cảm giác chuyên nghiệp cho hệ thống.

3. Lập trình chức năng đăng nhập bằng JavaScript

+ JavaScript được sử dụng để xử lý quá trình xác thực người dùng và điều hướng theo vai trò (Admin hoặc Nhân viên).

+ Ví dụ mã nguồn minh họa:

```
function login(event) {  
  event.preventDefault();  
  const username = document.getElementById("user").value;  
  const password = document.getElementById("pass").value;  
  
  const account = accounts.find(  
    acc => acc.username === username && acc.password === password  
  );  
  
  if (!account) {  
    alert("Sai tên đăng nhập hoặc mật khẩu!");  
    return;  
  }  
  
  window.location.href = account.redirect;  
}
```

Hình 4. 7: Ví dụ mã nguồn minh họa (Trang đăng nhập - js)

➔ Đoạn mã trên minh họa thuật toán đăng nhập: kiểm tra thông tin người dùng, xác định quyền truy cập và chuyển hướng sang trang chức năng phù hợp.

➔ Ngoài ra, hệ thống còn hỗ trợ chức năng **nhớ mật khẩu** thông qua localStorage, giúp nâng cao trải nghiệm người dùng.

4.1.3. Trang thống kê doanh thu

1. Thiết kế giao diện trang thống kê doanh thu bằng HTML

+ Trang thống kê doanh thu được xây dựng nhằm hỗ trợ quản trị viên theo dõi tình hình kinh doanh của quán cà phê theo từng khoảng thời gian. Giao diện được tổ chức rõ ràng, trực quan, bao gồm bộ lọc ngày, khu vực hiển thị tổng doanh thu và các biểu đồ thống kê.

+ Ví dụ mã nguồn minh họa:

```
<div class="date-filter">
  <label>Từ ngày:</label>
  <input type="date" id="from-date" />
  <label>Đến ngày:</label>
  <input type="date" id="to-date" />
  <button id="btn-statistics">Thống kê</button>
</div>

<div class="charts-container">
  <canvas id="lineChart"></canvas>
  <canvas id="pieChart"></canvas>
</div>
```

Hình 4. 8: Ví dụ mã nguồn minh họa (Trang thống kê doanh thu - html)

➔ Đoạn mã trên thể hiện cách bố trí các thành phần chính của trang thống kê, cho phép người dùng lựa chọn khoảng thời gian và xem kết quả thống kê thông qua biểu đồ.

2. Kỹ thuật định dạng giao diện bằng CSS

+ CSS được sử dụng để định dạng giao diện theo phong cách Dashboard, đảm bảo bố cục rõ ràng và dễ theo dõi số liệu.

+ Ví dụ mã nguồn minh họa:

```
.charts-container {  
  display: flex;  
  gap: 20px;  
  margin-top: 25px;  
}  
  
.chart-box {  
  background: #fff;  
  padding: 20px;  
  border-radius: 10px;  
  box-shadow: 0 2px 10px rgba(0, 0, 0, 0.06);  
}
```

Hình 4. 9: Ví dụ mã nguồn minh họa (Trang thống kê doanh thu – css)

➔ Việc sử dụng Flexbox giúp các biểu đồ được sắp xếp song song, đồng thời dễ dàng điều chỉnh khi thay đổi kích thước màn hình.

3. Lập trình chức năng thống kê doanh thu bằng JavaScript

+ JavaScript đóng vai trò xử lý **toàn bộ nghiệp vụ thống kê doanh thu**, từ việc lấy dữ liệu hóa đơn đã thanh toán trong localStorage cho đến xử lý và hiển thị kết quả thống kê trên giao diện người dùng.

+ Dữ liệu doanh thu được lấy từ danh sách hóa đơn đã thanh toán, được lưu trữ dưới khóa sales trong localStorage.

+ Ví dụ mã nguồn minh họa – xử lý thống kê:

```
// =====
// Parse saleDate: "HH:mm:ss dd/MM/yyyy"
// =====
function parseSaleDate(dateStr) {
    const [time, date] = dateStr.split(" ");
    const [day, month, year] = date.split("/");
    const [hour, minute, second] = time.split(":");
    return new Date(year, month - 1, day, hour, minute, second);
}

// =====
// Lấy các tháng trong khoảng
// =====
function getMonthsBetween(fromDate, toDate) {
    const months = [];
    let current = new Date(fromDate.getFullYear(), fromDate.getMonth(), 1);
    const end = new Date(toDate.getFullYear(), toDate.getMonth(), 1);

    while (current <= end) {
        const key = `${current.getFullYear()}-${String(
            current.getMonth() + 1
        ).padStart(2, "0")}`;
        months.push(key);
        current.setMonth(current.getMonth() + 1);
    }
    return months;
}
```

```
// =====
// Thống kê doanh thu
// =====
document.getElementById("btn-statistics").addEventListener("click", () => {
  const fromDateStr = document.getElementById("from-date").value;
  const toDateStr = document.getElementById("to-date").value;

  if (!fromDateStr || !toDateStr) {
    alert("Vui lòng chọn từ ngày và đến ngày!");
    return;
  }

  const fromDate = new Date(fromDateStr);
  const toDate = new Date(toDateStr);
  toDate.setHours(23, 59, 59, 999); // lấy trọn ngày

  const sales = JSON.parse(localStorage.getItem("sales")) || [];

  // =====
  // LỌC HÓA ĐƠN ĐÚNG NGÀY
  // =====
  const filteredSales = sales.filter((s) => {
    const d = parseSaleDate(s.saleDate);
    return d >= fromDate && d <= toDate;
  });

  if (filteredSales.length === 0) {
    alert("Không có hóa đơn trong khoảng thời gian này");
    return;
  }

  // =====
  // TỔNG DOANH THU
  // =====
  const totalRevenue = filteredSales.reduce((sum, s) => sum + s.finalTotal, 0);

  document.getElementById("total-revenue").textContent =
    formatNumber(totalRevenue) + " VNĐ";
});
```

Hình 4. 10: Ví dụ mã nguồn (Trang thống kê doanh thu – xử lý thống kê)

➔ Đoạn mã trên minh họa việc sử dụng Fetch API để lấy dữ liệu doanh thu trong khoảng thời gian người dùng lựa chọn.

+ Sau khi nhận dữ liệu, hệ thống tiến hành:

- Tính tổng doanh thu
- Gom nhóm doanh thu theo từng tháng

- Chuẩn bị dữ liệu cho biểu đồ

4. Trực quan hóa dữ liệu bằng biểu đồ (Chart.js)

+ Hệ thống sử dụng thư viện **Chart.js** để trực quan hóa dữ liệu doanh thu, giúp người quản lý dễ dàng phân tích và so sánh số liệu một cách trực quan.

- **Biểu đồ cột (Bar chart):** thể hiện tổng doanh thu theo từng tháng trong khoảng thời gian được chọn. Biểu đồ này giúp nhận biết xu hướng tăng hoặc giảm doanh thu theo thời gian.
- **Biểu đồ tròn (Pie chart):** thể hiện tỷ lệ số lượng món bán ra, từ đó xác định các món bán chạy và đóng góp chính vào doanh thu.

+ Ví dụ mã nguồn minh họa

- Biểu đồ doanh thu theo tháng:

```
// =====
// BIỂU ĐỒ DOANH THU THEO THÁNG
// =====
const months = getMonthsBetween(fromDate, toDate);
const revenueByMonth = {};
months.forEach((m) => (revenueByMonth[m] = 0));

filteredSales.forEach((s) => {
  const d = parseSaleDate(s.saleDate);
  const key = `${d.getFullYear()}-${String(d.getMonth() + 1).padStart(
    2,
    "0"
  )}`;
  revenueByMonth[key] += s.finalTotal;
});

if (window.lineChartInstance) window.lineChartInstance.destroy();
window.lineChartInstance = new Chart(document.getElementById("lineChart"), {
  type: "bar",
  data: {
    labels: months,
    datasets: [
      {
        label: "Doanh thu (VNĐ)",
        data: months.map((m) => revenueByMonth[m]),
      },
    ],
  },
  options: { responsive: true },
});
```

Hình 4. 11: Ví dụ mã nguồn minh họa (Biểu đồ cột - js)

- Đối với biểu đồ tròn, dữ liệu được tổng hợp dựa trên số lượng từng món bán ra trong các hóa đơn:

```
// =====
// BIỂU ĐỒ TỶ LỆ MÓN BÁN
// =====
const itemCountMap = {};

filteredSales.forEach((s) => {
  s.items.forEach((i) => {
    itemCountMap[i.ItemName] = (itemCountMap[i.ItemName] || 0) + i.Quantity;
  });
});

const pieLabels = Object.keys(itemCountMap);
const pieData = pieLabels.map((k) => itemCountMap[k]);

if (window.pieChartInstance) window.pieChartInstance.destroy();
window.pieChartInstance = new Chart(document.getElementById("pieChart"), {
  type: "pie",
  data: {
    labels: pieLabels,
    datasets: [{ data: pieData }],
  },
});
});
```

Hình 4. 12: Ví dụ mã nguồn minh họa (Biểu đồ tròn - js)

➔ Việc trực quan hóa bằng biểu đồ giúp quản trị viên nhanh chóng nắm bắt tình hình kinh doanh mà không cần xem chi tiết từng hóa đơn.

4.2. Kiểm thử và triển khai ứng dụng

4.2.1. Kiểm thử

1. Kiểm thử chức năng Đăng nhập

Bảng 4. 1: Bảng test case chức năng Đăng Nhập

Test Case	Input	Hành động	Output mong đợi
TC01	Username, Password hợp lệ	Nhấn Đăng nhập	Đăng nhập thành công
TC02	Username đúng, Password sai	Nhấn Đăng nhập	Thông báo sai mật khẩu

TC03	Username không tồn tại	Nhấn Đăng nhập	Thông báo tài khoản không tồn tại
TC04	Bỏ trống Username hoặc Password	Nhấn Đăng nhập	Thông báo yêu cầu nhập đủ thông tin
TC05	Tài khoản Admin	Đăng nhập	Hiển thị đầy đủ chức năng quản lý
TC06	Tài khoản Staff	Đăng nhập	Chỉ hiển thị chức năng bán hàng

2. Kiểm thử chức năng Quản lý bàn

Bảng 4. 2: Bảng test case chức năng Quản lý bàn

Test Case	Input	Hành động	Output mong đợi
TC07	Tên bàn hợp lệ	Nhấn Thêm	Thêm bàn thành công
TC08	Tên bàn trống	Nhấn Thêm	Thông báo lỗi
TC09	Tên bàn đã tồn tại	Nhấn Thêm	Thông báo trùng tên
TC10	Chọn bàn, sửa tên	Nhấn Sửa	Cập nhật thành công
TC11	Chọn bàn	Nhấn Xóa	Xóa bàn thành công
TC12	Bàn đang có hóa đơn	Nhấn Xóa	Không thể xóa bàn

3. Kiểm thử chức năng Quản lý danh mục và món

Bảng 4. 3: Bảng test case chức năng Quản lý danh mục và món

Test Case	Input	Hành động	Output mong đợi
TC13	Tên danh mục hợp lệ	Thêm danh mục	Thêm thành công

TC14	Tên danh mục trống	Thêm danh mục	Thông báo lỗi
TC15	Sửa tên danh mục	Nhấn Sửa	Cập nhật thành công
TC16	Danh mục có món	Nhấn Xóa	Không thể xóa
TC17	Tên món, giá, danh mục hợp lệ	Thêm món	Thêm món thành công
TC18	Giá món < 0	Thêm món	Thông báo lỗi
TC19	Sửa giá món	Nhấn Sửa	Giá được cập nhật
TC20	Chọn món	Nhấn Xóa	Xóa món thành công

4. Kiểm thử chức năng Tạo hóa đơn bán

Bảng 4. 4: Bảng test case chức năng Tạo hóa đơn bán

Test Case	Input	Hành động	Output mong đợi
TC21	Chọn bàn trống	Thêm món	Món được thêm vào hóa đơn
TC22	Thêm cùng món nhiều lần	Thêm món	Tăng số lượng
TC23	Sửa số lượng món	Cập nhật	Thành tiền cập nhật
TC24	Xóa món	Nhấn Xóa	Món bị loại bỏ
TC25	Hóa đơn có món	Nhấn Thanh toán	Lưu hóa đơn, bàn trống
TC26	Hóa đơn chưa có món	Nhấn Thanh toán	Thông báo lỗi

5. Kiểm thử chức năng Truy xuất hóa đơn

Bảng 4. 5: Bảng test case chức năng Truy xuất hóa đơn

Test Case	Input	Hành động	Output mong đợi
TC27	Mã hóa đơn hợp lệ	Tìm kiếm	Hiển thị chi tiết hóa đơn
TC28	Mã hóa đơn không tồn tại	Tìm kiếm	Thông báo không tìm thấy
TC29	Mã hóa đơn trống	Tìm kiếm	Thông báo lỗi

6. Kiểm thử chức năng Quản lý doanh thu

Bảng 4. 6: Bảng test case chức năng Quản lý doanh thu

Test Case	Input	Hành động	Output mong đợi
TC30	Từ ngày – đến ngày hợp lệ	Nhấn Thống kê	Hiển thị doanh thu
TC31	Khoảng thời gian không có dữ liệu	Nhấn Thống kê	Doanh thu = 0
TC32	Từ ngày > đến ngày	Nhấn Thống kê	Thông báo lỗi
TC33	Xem chi tiết doanh thu	Click hóa đơn	Hiển thị danh sách hóa đơn

7. Kiểm thử chức năng thay đổi mật khẩu

Bảng 4. 7: Bảng test case chức năng Thay đổi mật khẩu

Test Case	Input	Hành động	Output mong đợi
TC34	Mật khẩu cũ đúng, mật khẩu mới hợp lệ	Nhấn Đổi mật khẩu	Đổi mật khẩu thành công
TC35	Mật khẩu cũ sai	Nhấn Đổi mật khẩu	Thông báo mật khẩu cũ không đúng

TC36	Mật khẩu mới trống	Nhấn Đổi mật khẩu	Thông báo lỗi
TC37	Mật khẩu mới quá ngắn	Nhấn Đổi mật khẩu	Thông báo mật khẩu không hợp lệ
TC38	Mật khẩu mới và xác nhận không khớp	Nhấn Đổi mật khẩu	Thông báo không khớp
TC39	Đổi mật khẩu thành công	Đăng nhập lại	Đăng nhập bằng mật khẩu mới thành công

8. Kiểm thử chức năng sao lưu dữ liệu

Bảng 4. 8: Bảng test case chức năng Sao lưu dữ liệu

Test Case	Input	Hành động	Output mong đợi
TC40	Admin đăng nhập hợp lệ	Nhấn Sao lưu	Tạo file sao lưu thành công
TC41	Không có quyền sao lưu	Nhấn Sao lưu	Thông báo không có quyền
TC42	Đường dẫn lưu hợp lệ	Nhấn Sao lưu	File .bak được tạo
TC43	Đường dẫn không hợp lệ	Nhấn Sao lưu	Thông báo lỗi
TC44	Hệ thống đang hoạt động	Nhấn Sao lưu	Không mất dữ liệu

9. Kiểm thử chức năng khôi phục dữ liệu

Bảng 4. 9: Bảng test case chức năng khôi phục dữ liệu

Test Case	Input	Hành động	Output mong đợi
-----------	-------	-----------	-----------------

TC45	File sao lưu hợp lệ	Nhấn Khôi phục	Khôi phục thành công
TC46	File không tồn tại	Nhấn Khôi phục	Thông báo lỗi
TC47	File sai định dạng	Nhấn Khôi phục	Thông báo lỗi
TC48	Không có quyền Admin	Nhấn Khôi phục	Từ chối truy cập
TC49	Khôi phục xong	Tải lại hệ thống	Dữ liệu hiển thị đúng

4.2.2. Đóng gói ứng dụng

+ Mục đích của việc đóng gói sản phẩm

- Quy trình đóng gói sản phẩm nhằm tổng hợp đầy đủ các thành phần của đồ án **Website quản lý quán cà phê**, đảm bảo hệ thống có thể được triển khai, chạy thử và đánh giá một cách thuận tiện trên môi trường **trình duyệt web (localhost)**.
- Việc đóng gói giúp giảng viên và người dùng dễ dàng tiếp cận, kiểm tra chức năng và vận hành hệ thống mà không cần cài đặt thêm các dịch vụ máy chủ hay hệ quản trị cơ sở dữ liệu phức tạp.

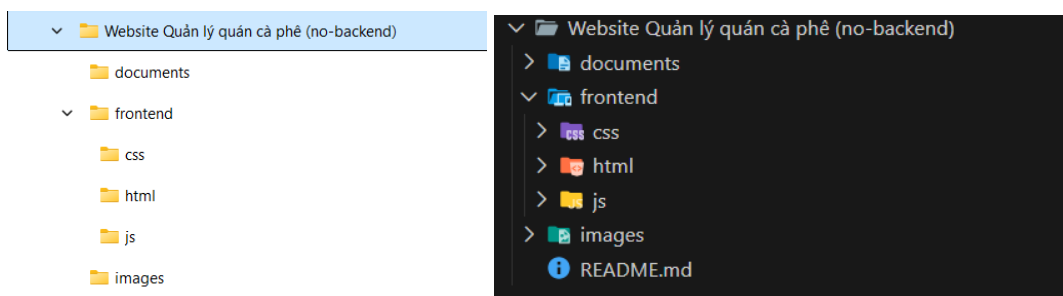
+ Thành phần của sản phẩm sau khi đóng gói

Sau khi hoàn thành quá trình phát triển và kiểm thử, sản phẩm đồ án được đóng gói bao gồm các thành phần chính sau:

- **Mã nguồn phía giao diện (Frontend)**
 - Công nghệ sử dụng: HTML, CSS, JavaScript
 - Chức năng:
 - Đăng nhập hệ thống
 - Quản lý bàn, danh mục và món
 - Tạo hóa đơn bán hàng
 - Thay đổi mật khẩu
 - Truy xuất hóa đơn
 - Thống kê và quản lý doanh thu
 - **Đặc điểm tổ chức mã nguồn:**

- Các tệp HTML, CSS và JavaScript được phân tách rõ ràng theo từng chức năng
 - Mã JavaScript xử lý toàn bộ nghiệp vụ và tương tác giao diện
 - Dữ liệu được lưu trữ và quản lý bằng localStorage, giúp hệ thống hoạt động độc lập và ổn định trên trình duyệt
- **Dữ liệu hệ thống (localStorage)**
- Toàn bộ dữ liệu của hệ thống được lưu trữ tại phía trình duyệt thông qua localStorage, bao gồm:
 - Danh sách bàn
 - Danh mục món
 - Danh sách món
 - Hóa đơn bán hàng
 - Thông tin người dùng đăng nhập
 - Dữ liệu phục vụ thống kê doanh thu
 - Dữ liệu được lưu dưới dạng JSON, mô phỏng cấu trúc dữ liệu của một hệ thống quản lý thực tế.

+ Cấu trúc thư mục đóng gói sản phẩm



Hình 4.13: Cấu trúc thư mục sau khi đóng gói

+ Quy trình đóng gói sản phẩm

- Bước 1: Kiểm tra và hoàn thiện hệ thống

Toàn bộ chức năng của hệ thống được kiểm thử nhằm đảm bảo hoạt động đúng yêu cầu, bao gồm đăng nhập, bán hàng, quản lý danh mục, quản lý món, quản lý bàn, thay đổi mật khẩu, thống kê doanh thu, sao lưu và khôi phục dữ liệu.

- Bước 2: Chuẩn hóa mã nguồn

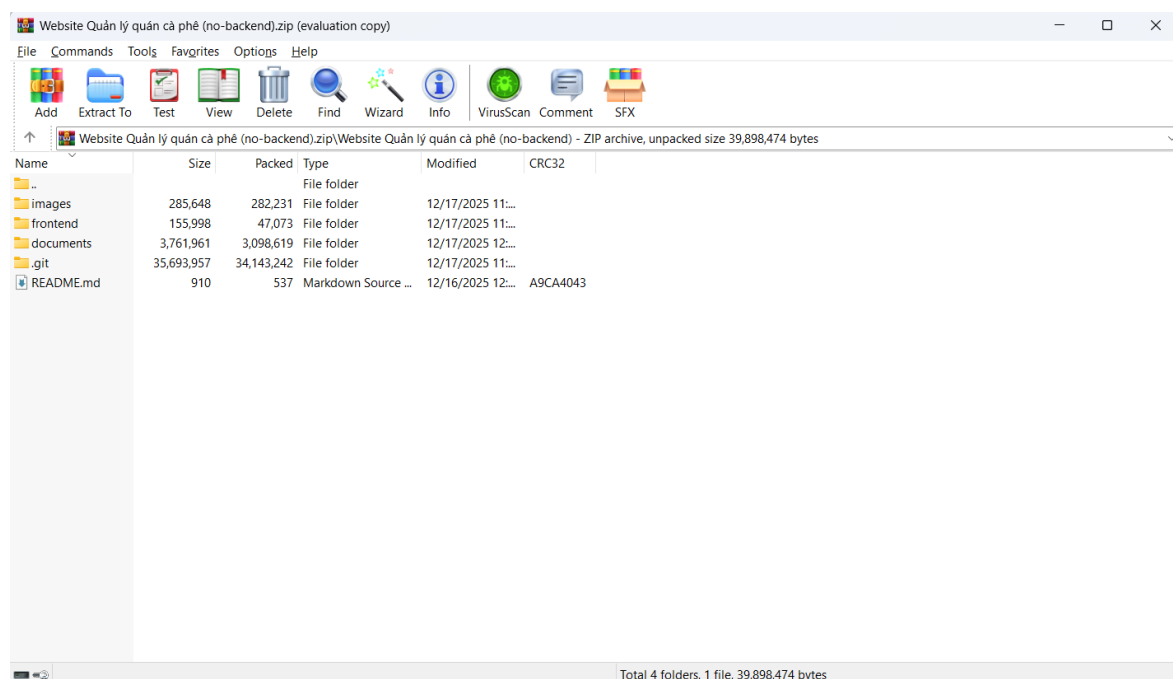
Mã nguồn được rà soát, loại bỏ các tệp không cần thiết, sắp xếp lại cấu trúc thư mục và bổ sung chú thích nhằm tăng khả năng bảo trì.

- Bước 3: Xuất cơ sở dữ liệu

Cơ sở dữ liệu được xuất ra file sao lưu để đảm bảo khả năng phục hồi và phục vụ quá trình triển khai hệ thống.

- Bước 4: Đóng gói sản phẩm

Toàn bộ các thành phần được gom lại trong một thư mục gốc và nén thành file .zip để thuận tiện cho việc nộp và triển khai.



Hình 4. 14: Cấu trúc file .zip

4.2.3. Triển khai ứng dụng

+ Mục đích

Phần này trình bày các điều kiện về **phần cứng** và **phần mềm** cần thiết để triển khai và vận hành hệ thống Website quản lý cà phê trên môi trường localhost, đảm bảo ứng dụng hoạt động ổn định và đúng chức năng.

+ Điều kiện phần cứng

Hệ thống có thể triển khai trên máy tính cá nhân với cấu hình trung bình. Yêu cầu phần cứng tối thiểu như sau:

Bảng 4. 10: Bảng điều kiện phần cứng

Thành phần	Yêu cầu tối thiểu
Bộ xử lý (CPU)	Intel Core i3 hoặc tương đương
Bộ nhớ RAM	4 GB
Ổ cứng	20 GB trống
Thiết bị hiển thị	Màn hình độ phân giải tối thiểu 1366x789
Kết nối	Kết nối mạng nội bộ hoặc internet

+ **Điều kiện phần mềm**

- **Hệ điều hành**

- Windows 10 / Windows 11
- Hệ điều hành 64-bit

+ **Phần mềm và công cụ phát triển**

Bảng 4. 11: Bảng phần mềm và công cụ phát triển

Phần mềm	Phiên bản/ Ghi chú
Trình duyệt Web	Google Chrome/ Microsoft Edge
VS Code	Soạn thảo mã nguồn
Git	Quản lý mã nguồn

+ **Công nghệ sử dụng trong hệ thống**

- Frontend: HTML, CSS, JavaScript
- Backend: NodeJS
- Cơ sở dữ liệu: SQL Server
- Mô hình triển khai: Localhost

+ **Môi trường triển khai**

- Hệ thống được triển khai và chạy trên **máy tính cá nhân**
- Backend chạy trên **NodeJS Server (localhost)**
- Cơ sở dữ liệu được quản lý bằng **SQL Server**
- Người dùng truy cập hệ thống thông qua trình duyệt web

+ **Điều kiện vận hành**

- Máy chủ và cơ sở dữ liệu phải được khởi động trước khi truy cập hệ thống

- Người dùng cần có tài khoản hợp lệ (Admin hoặc Staff)
- Dữ liệu được sao lưu định kỳ để đảm bảo an toàn

KẾT LUẬN

+ Trong quá trình thực hiện đề tài “**Website quản lý quán cà phê**”, em đã tiến hành phân tích yêu cầu, thiết kế, xây dựng và triển khai một hệ thống quản lý phục vụ hoạt động kinh doanh quán cà phê trên môi trường **trình duyệt web (localhost)**. Đề tài hướng đến việc tin học hóa các nghiệp vụ cơ bản như quản lý bàn, danh mục, món, lập hóa đơn bán hàng và thống kê doanh thu, qua đó góp phần nâng cao hiệu quả quản lý và giảm thiểu sai sót trong quá trình vận hành.

+ Kết quả đạt được cho thấy hệ thống đã đáp ứng đầy đủ các yêu cầu chức năng đã đề ra, bao gồm đăng nhập và quản lý tài khoản, quản lý dữ liệu bàn – danh mục – món, tạo và lưu trữ hóa đơn bán hàng, tra cứu chi tiết hóa đơn, thống kê doanh thu, thay đổi mật khẩu, cũng như sao lưu và khôi phục dữ liệu. Các chức năng được xây dựng logic, kiểm thử cẩn thận thông qua các kịch bản kiểm thử (test case), đảm bảo hệ thống hoạt động ổn định và đúng yêu cầu.

+ Về mặt công nghệ, hệ thống được phát triển hoàn toàn bằng **HTML, CSS và JavaScript**, trong đó JavaScript đảm nhiệm vai trò xử lý nghiệp vụ và quản lý dữ liệu thông qua **localStorage** của trình duyệt. Việc lựa chọn mô hình này giúp hệ thống hoạt động độc lập, không phụ thuộc vào máy chủ hay cơ sở dữ liệu bên ngoài, đồng thời thuận tiện cho việc triển khai, kiểm thử và đánh giá trong phạm vi đồ án học phần.

+ Bên cạnh những kết quả đạt được, đề tài vẫn còn một số hạn chế nhất định, chẳng hạn như hệ thống chưa được triển khai trên môi trường máy chủ thực tế, dữ liệu chỉ được lưu trữ cục bộ trên trình duyệt, và chưa tích hợp các chức năng nâng cao như thanh toán trực tuyến, quản lý nhiều chi nhánh hoặc báo cáo phân tích chuyên sâu. Tuy nhiên, đây cũng chính là những hướng phát triển tiềm năng trong tương lai khi hệ thống được mở rộng sang mô hình có backend và cơ sở dữ liệu tập trung.

→ **Tổng kết lại**, đề tài *Website quản lý quán cà phê* đã hoàn thành các mục tiêu đặt ra, thể hiện khả năng vận dụng kiến thức đã học vào thực tiễn, đồng thời tạo nền tảng ban đầu cho việc tiếp tục nghiên cứu, nâng cấp và phát triển hệ thống trong các giai đoạn tiếp theo.

TÀI LIỆU THAM KHẢO

[1] W3Schools, *HTML Tutorial*

[2] W3Schools, *CSS Tutorial*

[3] W3Schools, *JavaScript Tutorial*