TRƯỜNG ĐẠI HỌC TRÀ VINH

KHOA KỸ THUẬT VÀ CÔNG NGHỆ

**BỘ MÔN CÔNG NGHỆ THÔNG TIN**

**----------****-----------**

****

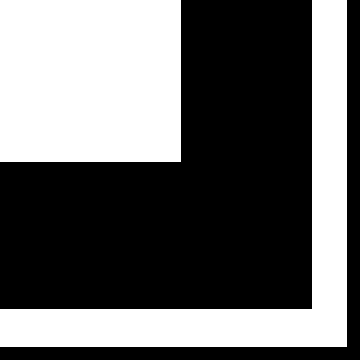
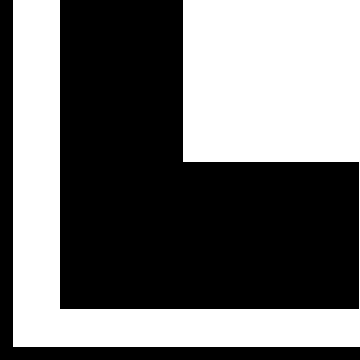
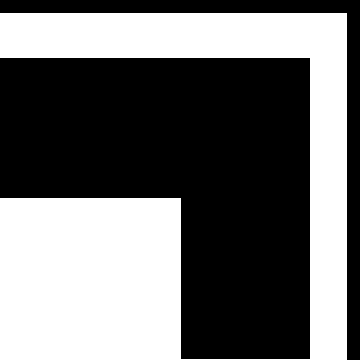
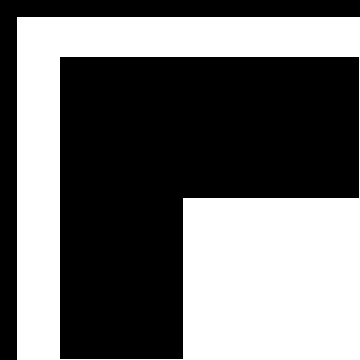
BÁO CÁO MÔN HỌC

**HỌC PHẦN: CÔNG NGHỆ PHẦN MỀM**

**WEBSITE QUẢN LÝ CỬA HÀNG GIẶT SẤY**

|  |  |  |  |
| --- | --- | --- | --- |
| *Giáo viên hướng dẫn:* |  | *Sinh viên thực hiện - Mã số sinh viên:* | |
| **ThS. Nguyễn Bảo Ân** | | Lưu Bích Ngọc  Trương Nguyễn Tố Nguyên  Nguyễn Thị Thanh Phụng | 110122123  110122127  110122143 |
|  | |
| Lớp DA22TTB | |

**Trà Vinh, tháng 7 năm 2025**



**NHẬN XÉT CỦA GIÁO VIÊN HƯỚNG DẪN**

*Trà Vinh, ngày …. tháng …… năm ……*

**Giáo viên hướng dẫn**

*(Ký tên và ghi rõ họ tên)*

**NHẬN XÉT CỦA THÀNH VIÊN HỘI ĐỒNG**

*Trà Vinh, ngày …. tháng …… năm ……*

**Thành viên hội đồng**

*(Ký tên và ghi rõ họ tên)*

**LỜI CẢM ƠN**

Lời nói đầu, chúng em xin chân thành cảm ơn thầy Nguyễn Bảo Ân đã hỗ trợ, hướng dẫn tận tình chúng em trong thời gian em làm môn Công nghệ phần mềm, những kiến thức mà thầy đã tận tình dạy bảo là hành trang quý báu trên con đường học vấn và phát triển sự nghiệp tương lai rộng mở của chúng em. Thầy đã luôn kiên nhẫn, nhiệt tình trong việc truyền đạt kiến thức và kinh nghiệm quý báu, giúp chúng em vượt qua những khó khăn và thử thách trong quá trình học tập và nghiên cứu.

Những lời khuyên, góp ý của thầy không chỉ là kim chỉ nam cho sự phát triển của đồ án môn học này mà còn là nguồn động viên, khích lệ tinh thần lớn lao cho chúng em.

Do chưa có nhiều kinh nghiệm cũng như hạn chế về kiến thức nên không tránh khỏi những thiếu sót trong bài báo cáo. Chúng em rất mong nhận được sự góp ý của quý thầy cô để bài báo cáo được hoàn thiện tốt hơn. Chúng em xin hứa sẽ tiếp tục nỗ lực không ngừng để không phụ lòng thầy đã dành cho chúng em. Chúng em xin chân thành cảm ơn!

*Sinh viên ký và ghi rõ họ và tên*

|  |  |  |
| --- | --- | --- |
| **Sinh viên 1** | **Sinh viên 2** | **Sinh viên 3** |
| Lưu Bích Ngọc | Trương Nguyễn Tố Nguyên | Nguyễn Thị Thanh Phụng |

**MỤC LỤC**

[CHƯƠNG 1: GIỚI THIỆU 1](#_Toc205561078)

[1.1. Lý do chọn đề tài 1](#_Toc205561079)

[1.2. Mục tiêu đề tài 1](#_Toc205561080)

[1.3. Phạm vi đề tài 1](#_Toc205561081)

[1.4. Đối tượng sử dụng 2](#_Toc205561082)

[1.5. Phương pháp thực hiện 2](#_Toc205561083)

[CHƯƠNG 2: CƠ SỞ LÝ THUYẾT VÀ CÔNG NGHỆ SỬ DỤNG 3](#_Toc205561084)

[2.1. Cơ sở lý thuyết 3](#_Toc205561085)

[2.2. Công cụ và công nghệ sử dụng 3](#_Toc205561088)

[CHƯƠNG 3: PHÂN TÍCH VÀ THIẾT KẾ HỆ THỐNG 5](#_Toc205561097)

[3.1. Khảo sát hiện trạng 5](#_Toc205561098)

[3.2. Phân tích yêu cầu hệ thống 5](#_Toc205561099)

[3.3. Thiết kế hệ thống 5](#_Toc205561102)

[3.4. Thiết kế cơ sở dữ liệu 14](#_Toc205561106)

[3.5. Thiết kế giao diện 15](#_Toc205561107)

[3.6. Thiết kế xử lý 23](#_Toc205561110)

[CHƯƠNG 4: CÀI ĐẶT VÀ TRIỂN KHAI 25](#_Toc205561116)

[4.1. Mô hình phát triển phần mềm (Đáp ứng tiêu chí L2) 25](#_Toc205561117)

[4.2. Tính cần thiết của phần mềm dạng phân tán và triển khai trên nền tảng đám mây (Đáp ứng tiêu chí L2 và L6) 25](#_Toc205561120)

[4.3. Môi trường phát triển 26](#_Toc205561123)

[4.4. Cài đặt hệ thống 27](#_Toc205561124)

[4.5. Triển khai 35](#_Toc205561127)

[CHƯƠNG 5: KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN 39](#_Toc205561128)

[5.1. Kết luận 39](#_Toc205561129)

[5.2. Ưu và nhược điểm 39](#_Toc205561130)

[5.3. Hướng phát triển 39](#_Toc205561133)

[TÀI LIỆU THAM KHẢO 41](#_Toc205561134)

**DANH MỤC HÌNH ẢNH**

[Hình 3.1. Giao diện trang chủ 16](#_Toc205559560)

[Hình 3.2. Trang giới thiệu 17](#_Toc205559561)

[Hình 3.3. Trang dịch vụ 18](#_Toc205559562)

[Hình 3.4. Trang đặt dịch vụ 19](#_Toc205559563)

[Hình 3.5. Trang liên hệ 20](#_Toc205559564)

[Hình 3.6. Trang đăng nhập 21](#_Toc205559565)

[Hình 3.7. Trang tổng quan 21](#_Toc205559566)

[Hình 3.8. Trang quản lí đơn hàng 22](#_Toc205559567)

[Hình 3.9. Trang quản lý hóa đơn 22](#_Toc205559568)

[Hình 3.10. Trang quản lý nhân viên 23](#_Toc205559569)

**DANH MỤC BẢNG**

[Bảng 3.1. Mô tả thực thể KhachHang 7](#_Toc205559458)

[Bảng 3.2. Mô tả thực thể DichVu 7](#_Toc205559459)

[Bảng 3.3. Mô tả thực thể NhanVien 8](#_Toc205559460)

[Bảng 3.4. Mô tả thực thể CaTruc 8](#_Toc205559461)

[Bảng 3.5. Mô tả thực thể Truc 9](#_Toc205559462)

[Bảng 3.6. Mô tả thực thể NhaCungCap 9](#_Toc205559463)

[Bảng 3.7. Mô tả thực thể PhanQuyen 10](#_Toc205559464)

[Bảng 3.8. Mô tả thực thể TaiKhoan 10](#_Toc205559465)

[Bảng 3.9. Mô tả thực thể DonDat 11](#_Toc205559466)

[Bảng 3.10. Mô tả thực thể ChiTietDonDat 12](#_Toc205559467)

[Bảng 3.11. Mô tả thực thể HoaDon 13](#_Toc205559468)

[Bảng 3.12. Mô tả thực thể NguyenVatLieu 13](#_Toc205559469)

[Bảng 3.13. Mô tả thực thể PhieuNhap 14](#_Toc205559470)

[Bảng 3.14. Mô tả thực thể Nhap 14](#_Toc205559471)

# TÓM TẮT ĐỀ TÀI

**Mô tả đề tài**

Trong thời đại công nghệ số phát triển mạnh mẽ, việc ứng dụng các hệ thống quản lý thông minh vào đời sống hằng ngày đang trở thành xu hướng tất yếu. Ngành dịch vụ giặt sấy cũng không nằm ngoài xu thế đó, đặc biệt khi nhu cầu của khách hàng ngày càng tăng cao về sự tiện lợi, nhanh chóng và chính xác trong khâu đặt dịch vụ, theo dõi đơn hàng, thanh toán và quản lý hoạt động kinh doanh.

Xuất phát từ nhu cầu thực tế đó, nhóm chúng em đã xây dựng đề tài: **"Website quản lý cửa hàng giặt sấy"** dưới dạng một hệ thống web hoàn chỉnh nhằm hỗ trợ quản lý quy trình hoạt động của cửa hàng giặt sấy, bao gồm từ khâu tiếp nhận đơn đặt, quản lý nhân viên, quản lý dịch vụ, tạo hóa đơn đến phân quyền sử dụng hệ thống.

**Mục tiêu của đề tài**

* Thiết kế và triển khai hệ thống web quản lý dịch vụ giặt sấy.
* Cho phép khách hàng đặt dịch vụ trực tuyến.
* Hỗ trợ nhân viên quản lý đơn đặt và hóa đơn.
* Tối ưu quy trình chuyển đổi đơn đặt thành hóa đơn.
* Tích hợp phân quyền người dùng (quản trị và nhân viên).

**Công cụ và công nghệ sử dụng**

* Ngôn ngữ Backend: Node.js (Express.js).
* Frontend: HTML, CSS, JavaScript.
* Database: MySQL.
* Thiết kế giao diện và sơ đồ Use Case: Figma.
* Thiết kế sơ đồ cơ sở dữ liệu (ERD): draw.io.
* Quản lý dự án: Jira.
* Lưu trữ mã nguồn: GitHub.

**Phạm vi**

Hệ thống bao gồm các chức năng chính sau:

* Đặt dịch vụ giặt sấy (khách hàng).
* Quản lý đơn đặt và chuyển sang hóa đơn (nhân viên/ quản trị viên).
* Quản lý dịch vụ, đơn vị tính.
* Quản lý nhân viên.
* Phân quyền hệ thống và quản lý tài khoản.

# 

# CHƯƠNG 1: GIỚI THIỆU

## 1.1. Lý do chọn đề tài

Với nhịp sống nhanh và bộn bề ngày nay, nhu cầu sử dụng các dịch vụ giặt sấy ngày càng trở nên phổ biến, đặc biệt tại các khu dân cư đông đúc, khu công nghiệp và khu vực sinh viên. Tuy nhiên, nhiều cửa hàng giặt sấy vẫn còn vận hành theo phương thức thủ công, gây khó khăn trong việc quản lý đơn đặt, theo dõi hóa đơn. Điều này không chỉ làm giảm hiệu quả kinh doanh mà còn ảnh hưởng đến trải nghiệm của khách hàng.

Nhằm giải quyết các hạn chế trên, nhóm chúng em quyết định xây dựng một hệ thống web hỗ trợ quản lý dịch vụ giặt sấy. Hệ thống không chỉ giúp khách hàng đặt dịch vụ trực tuyến nhanh chóng mà còn giúp nhân viên quản lý đơn đặt và hóa đơn một cách khoa học, minh bạch và tiện lợi.

## 1.2. Mục tiêu đề tài

Đề tài hướng đến các mục tiêu cụ thể sau:

* Xây dựng một hệ thống web hoàn chỉnh cho cửa hàng giặt sấy, giúp khách hàng có thể đặt dịch vụ online.
* Hỗ trợ nhân viên quản lý đơn đặt, tạo hóa đơn, nhập nguyên vật liệu và theo dõi ca trực hiệu quả.
* Phân quyền rõ ràng giữa các đối tượng sử dụng hệ thống (quản trị, nhân viên).
* Thiết kế giao diện đơn giản, dễ sử dụng trên cả desktop và thiết bị di động.
* Ứng dụng các công nghệ hiện đại như Node.js, MySQL, Docker, giúp hệ thống hoạt động ổn định và dễ triển khai.

## 1.3. Phạm vi đề tài

Hệ thống quản lý giặt sấy gồm các chức năng chính:

* Đăng nhập và phân quyền người dùng.
* Quản lý nhân viên.
* Khách hàng đặt dịch vụ giặt sấy trực tuyến.
* Nhân viên xác nhận đơn đặt và tạo hóa đơn tương ứng.
* Quản lý danh sách dịch vụ và đơn vị tính.

## 1.4. Đối tượng sử dụng

* **Khách hàng**: Sử dụng hệ thống để đặt dịch vụ giặt sấy.
* **Nhân viên**: Quản lý đơn đặt, tạo hóa đơn.
* **Quản trị viên**: Quản lý nhân viên, tài khoản, phân quyền, dịch vụ và toàn bộ hệ thống.

## 1.5. Phương pháp thực hiện

* Phân tích yêu cầu chức năng hệ thống.
* Thiết kế mô hình dữ liệu (ERD), Use Case và giao diện bằng Figma, draw.io.
* Triển khai hệ thống với Node.js và MySQL.
* Kiểm thử, hiệu chỉnh và triển khai thử nghiệm trên môi trường cục bộ với Docker.
* Ghi nhận kết quả và đánh giá hệ thống.

# CHƯƠNG 2: CƠ SỞ LÝ THUYẾT VÀ CÔNG NGHỆ SỬ DỤNG

## 2.1. Cơ sở lý thuyết

### 2.1.1. Mô hình quan hệ

Mô hình quan hệ mô tả dữ liệu dưới dạng các bảng (table) với các hàng (record) và cột (field). Mỗi bảng đại diện cho một thực thể và các mối quan hệ giữa các bảng được thể hiện qua khóa chính (Primary Key) và khóa ngoại (Foreign Key).

### 2.1.2. Use Case Diagram

Use Case là sơ đồ mô tả các chức năng mà hệ thống cung cấp cho người dùng (actor). Nó giúp làm rõ các yêu cầu chức năng và định hướng thiết kế.

## 2.2. Công cụ và công nghệ sử dụng

### 2.2.1. Node.js

Node.js là môi trường chạy JavaScript phía máy chủ. Với ưu điểm bất đồng bộ (asynchronous), nhẹ và nhanh, Node.js phù hợp để xây dựng các ứng dụng web hiện đại và có khả năng mở rộng.

### 2.2.2. Express.js

Express là framework phổ biến của Node.js, giúp xây dựng các API một cách nhanh chóng với cú pháp đơn giản, rõ ràng.

### 2.2.3. MySQL

MySQL là hệ quản trị cơ sở dữ liệu quan hệ phổ biến, hỗ trợ thao tác dữ liệu nhanh, bảo mật cao và dễ tích hợp với nhiều công nghệ web.

### 2.2.4. Docker

Docker là công cụ đóng gói ứng dụng cùng môi trường chạy thành container, giúp triển khai hệ thống dễ dàng, đồng nhất giữa các môi trường (dev, test, production).

### 2.2.5. GitHub

GitHub là nền tảng quản lý mã nguồn và làm việc nhóm. Trong quá trình làm việc, nhóm sử dụng GitHub để quản lý phiên bản code, phân công nhiệm vụ và theo dõi tiến độ.

### 2.2.6. Jira

Jira là công cụ quản lý dự án và công việc theo mô hình Agile. Nhóm sử dụng Jira để tạo task, theo dõi deadline và đánh giá hiệu suất làm việc.

### 2.2.7. Figma

Figma là công cụ thiết kế giao diện UI/UX trực tuyến. Giao diện các trang đặt dịch vụ, quản trị... được nhóm thiết kế trên Figma để đảm bảo tính trực quan và dễ dùng.

### 2.2.8. Draw.io (hoặc Dawio)

Draw.io được dùng để thiết kế các sơ đồ như mô hình quan hệ thực thể (ERD), sơ đồ Use Case, sơ đồ phân rã chức năng... hỗ trợ trực quan hóa thiết kế hệ thống.

# CHƯƠNG 3: PHÂN TÍCH VÀ THIẾT KẾ HỆ THỐNG

## 3.1. Khảo sát hiện trạng

Hiện nay, nhiều tiệm giặt sấy tại địa phương vẫn quản lý theo phương pháp thủ công như ghi đơn hàng bằng giấy, tính tiền và lưu hóa đơn bằng tay, gây mất thời gian và dễ xảy ra sai sót. Khách hàng muốn đặt dịch vụ phải đến tận nơi, không thể đặt online hoặc theo dõi tình trạng đơn hàng. Việc thiếu một hệ thống quản lý dịch vụ giặt sấy hiệu quả gây ảnh hưởng đến năng suất làm việc và trải nghiệm khách hàng.

Do đó, nhóm xây dựng hệ thống web giặt sấy nhằm tự động hóa quy trình tiếp nhận đơn đặt dịch vụ, quản lý hóa đơn và phân quyền tài khoản sử dụng.

## 3.2. Phân tích yêu cầu hệ thống

### 3.2.1. Yêu cầu chức năng

* Khách hàng:
  + Đặt dịch vụ giặt sấy online.
  + Xem trạng thái đơn hàng.
* Nhân viên:
  + Đăng nhập tài khoản.
  + Xử lý đơn đặt (cập nhật trạng thái).
  + Lập hóa đơn cho đơn đặt.
* Quản trị viên:
  + Quản lý nhân viên.
  + Quản lý dịch vụ, đơn vị tính.
  + Quản lý nguyên vật liệu.
  + Quản lý phân quyền và tài khoản.

### 3.2.2. Yêu cầu phi chức năng

* Giao diện đơn giản, thân thiện.
* Hệ thống hoạt động ổn định trên mọi trình duyệt.
* Dữ liệu được lưu trữ bảo mật.
* Có khả năng mở rộng khi quy mô cửa hàng tăng.

## 3.3. Thiết kế hệ thống

### 3.3.1. Sơ đồ Use Case

Sơ đồ Use Case mô tả các chức năng chính của hệ thống với ba tác nhân:

* + Khách hàng
  + Nhân viên
  + Quản trị viên

### 3.3.2. Mô hình thực thể - kết hợp (ERD)

Mô hình ERD gồm các thực thể:

1. KhachHang
2. DichVu
3. NhanVien
4. CaTruc
5. Truc (phân công lịch trực)
6. NhaCungCap
7. PhanQuyen
8. TaiKhoan
9. DonDat
10. ChiTietDonDat
11. HoaDon
12. NguyenVatLieu
13. PhieuNhap
14. Nhap (chi tiết phiếu nhập)

### 3.3.3. Mô tả thực thể

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Thuộc tính** | **Diễn giải** | **Loại giá trị** | **Kiểu dữ liệu** | **Miền giá trị** | **Chiều dài** | **Ghi chú** |
| MaKH | Mã khách hàng | Bắt buộc | Varchar | Khóa chính | 50 |  |
| TenKH | Tên khách hàng | Bắt buộc | Varchar |  | 100 |  |
| DiaChi | Địa chỉ | Bắt buộc | Varchar |  | 200 |  |
| SDT\_KH | Số điện thoại của khách hàng | Bắt buộc | Varchar |  | 15 |  |
| EmailKH | Email của khách hàng | Bắt buộc | Varchar |  | 100 |  |

Bảng 3.1. Mô tả thực thể KhachHang

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Thuộc tính** | **Diễn giải** | **Loại giá trị** | **Kiểu dữ liệu** | **Miền giá trị** | **Chiều dài** | **Ghi chú** |
| MaDichVu | Mã dịch vụ | Bắt buộc | Varchar | Khóa chính | 10 |  |
| TenDichVu | Tên dịch vụ | Bắt buộc | Varchar |  | 100 |  |
| DonGia | Đơn giá | Bắt buộc | Decimal |  | 18, 2 |  |

Bảng 3.2. Mô tả thực thể DichVu

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Thuộc tính** | **Diễn giải** | **Loại giá trị** | **Kiểu dữ liệu** | **Miền giá trị** | **Chiều dài** | **Ghi chú** |
| MaNV | Mã nhân viên | Bắt buộc | Varchar | Khóa chính | 10 |  |
| TenNV | Tên nhân viên | Bắt buộc | Varchar |  | 100 |  |
| GioiTinh | Giới tính | Bắt buộc | Varchar | Nam/Nữ | 5 |  |
| NgaySinh | Ngày sinh | Bắt buộc | Date |  |  |  |
| SDT\_NV | Số điện thoại của nhân viên | Bắt buộc | Varchar |  | 15 |  |

Bảng 3.3. Mô tả thực thể NhanVien

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Thuộc tính** | **Diễn giải** | **Loại giá trị** | **Kiểu dữ liệu** | **Miền giá trị** | **Chiều dài** | **Ghi chú** |
| STT | Số thứ tự | Bắt buộc | Int | Khóa chính |  |  |
| TGBatDau | Thời gian bắt đầu | Bắt buộc | Time |  |  |  |
| TGKetThuc | Thời gian kết thúc | Bắt buộc | Time |  |  |  |
| TenCa | Tên ca |  | Varchar |  | 50 |  |

Bảng 3.4. Mô tả thực thể CaTruc

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Thuộc tính** | **Diễn giải** | **Loại giá trị** | **Kiểu dữ liệu** | **Miền giá trị** | **Chiều dài** | **Ghi chú** |
| MaNV | Mã nhân viên | Bắt buộc | Varchar | Khóa chính,  FK -> NhanVien | 10 |  |
| STT | Số thứ tự | Bắt buộc | Int | Khóa chính,  FK -> CaTruc |  |  |
| NgayTruc | Ngày trực | Bắt buộc | Date | Khóa chính |  |  |

Bảng 3.5. Mô tả thực thể Truc

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Thuộc tính** | **Diễn giải** | **Loại giá trị** | **Kiểu dữ liệu** | **Miền giá trị** | **Chiều dài** | **Ghi chú** |
| MaNCC | Mã nhà cung cấp | Bắt buộc | Varchar | Khóa chính | 10 |  |
| TenNCC | Tên nhà cung cấp | Bắt buộc | Varchar |  | 100 |  |
| DiaChiNCC | Địa chỉ nhà cung cấp | Bắt buộc | Varchar |  | 200 |  |
| SDT\_NCC | Số điện thoại của nhà cung cấp | Bắt buộc | Varchar |  | 15 |  |

Bảng 3.6. Mô tả thực thể NhaCungCap

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Thuộc tính** | **Diễn giải** | **Loại giá trị** | **Kiểu dữ liệu** | **Miền giá trị** | **Chiều dài** | **Ghi chú** |
| MaPQ | Mã phân quyền | Bắt buộc | Varchar | Khóa chính | 10 |  |
| TenPQ | Tên phân quyền | Bắt buộc | Varchar |  | 50 |  |

Bảng 3.7. Mô tả thực thể PhanQuyen

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Thuộc tính** | **Diễn giải** | **Loại giá trị** | **Kiểu dữ liệu** | **Miền giá trị** | **Chiều dài** | **Ghi chú** |
| MaTK | Mã tài khoản | Bắt buộc | Varchar | Khóa chính | 50 |  |
| TenTK | Tên tài khoản | Bắt buộc | Varchar |  | 50 |  |
| MatKhau | Mật khẩu | Bắt buộc | Varchar |  | 50 |  |
| MaPQ | Mã phân quyền | Bắt buộc | Varchar | FK -> PhanQuyen | 10 |  |
| MaNV | Mã nhân viên | Bắt buộc | Varchar | FK -> NhanVien | 10 |  |

Bảng 3.8. Mô tả thực thể TaiKhoan

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Thuộc tính** | **Diễn giải** | **Loại giá trị** | **Kiểu dữ liệu** | **Miền giá trị** | **Chiều dài** | **Ghi chú** |
| MaDonDat | Mã đơn đặt | Bắt buộc | Varchar | Khóa chính | 10 |  |
| TenKhachHang | Tên khách hàng | Bắt buộc | Varchar |  | 100 |  |
| SDT\_KH | Số điện thoại của khách hàng | Bắt buộc | Varchar |  | 15 |  |
| NgayDat | Ngày đặt | Bắt buộc | Date |  |  |  |
| MaDichVu | Mã dịch vụ | Bắt buộc | Varchar | FK -> DichVu | 10 |  |
| GhiChu | Ghi chú |  | Text |  |  |  |
| TrangThai | Trạng thái đơn đặt | Bắt buộc | Varchar |  | 50 |  |
| MaKH | Mã khách hàng | Bắt buộc | Varchar | FK -> KhachHang | 10 |  |

Bảng 3.9. Mô tả thực thể DonDat

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Thuộc tính** | **Diễn giải** | **Loại giá trị** | **Kiểu dữ liệu** | **Miền giá trị** | **Chiều dài** | **Ghi chú** |
| MaDonDat | Mã đơn đặt | Bắt buộc | Varchar | Khóa chính,  FK -> DonDat | 10 |  |
| MaDichVu | Mã dịch vụ | Bắt buộc | Varchar | Khóa chính,  FK -> DichVu | 10 |  |
| DonGia | Đơn giá | Bắt buộc | Decimal |  | 10, 2 |  |
| SoLuong | Số lượng | Bắt buộc | Int |  |  |  |
| ThanhTien | Thành tiền | Bắt buộc | Decimal |  | 18, 2 |  |
| DonViTinh | Đơn vị tính | Bắt buộc | Varchar |  | 50 |  |

Bảng 3.10. Mô tả thực thể ChiTietDonDat

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Thuộc tính** | **Diễn giải** | **Loại giá trị** | **Kiểu dữ liệu** | **Miền giá trị** | **Chiều dài** | **Ghi chú** |
| MaHD | Mã hóa đơn | Bắt buộc | Int | Khóa chính |  |  |
| NgayLap | Ngày lập | Bắt buộc | Date |  |  |  |
| MaDonDat | Mã đơn đặt | Bắt buộc | Varchar | FK -> DonDat | 10 |  |
| SoLuong | Số lượng | Bắt buộc | Int |  |  |  |
| TrangThai | Trạng thái | Bắt buộc | Varchar |  | 50 |  |

Bảng 3.11. Mô tả thực thể HoaDon

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Thuộc tính** | **Diễn giải** | **Loại giá trị** | **Kiểu dữ liệu** | **Miền giá trị** | **Chiều dài** | **Ghi chú** |
| MaNVL | Mã nguyên vật liệu | Bắt buộc | Int | Khóa chính |  |  |
| TenNVL | Tên nguyên vật liệu | Bắt buộc | Varchar |  | 100 |  |
| GiaTien | Giá tiền | Bắt buộc | Decimal |  | 18, 2 |  |

Bảng 3.12. Mô tả thực thể NguyenVatLieu

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Thuộc tính** | **Diễn giải** | **Loại giá trị** | **Kiểu dữ liệu** | **Miền giá trị** | **Chiều dài** | **Ghi chú** |
| MaPN | Mã phiếu nhập | Bắt buộc | Int | Khóa chính |  |  |
| NgayNhap | Ngày nhập | Bắt buộc | Date |  |  |  |
| DotNhap | Đợt nhập | Bắt buộc | Varchar |  |  |  |
| MaNV | Mã nhân viên | Bắt buộc | Varchar | FK -> NhanVien |  |  |
|  |  |  |  |  |  |  |
| MaNCC | Mã nhà cung cấp | Bắt buộc | Varchar | FK -> NhaCungCap |  |  |

Bảng 3.13. Mô tả thực thể PhieuNhap

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Thuộc tính** | **Diễn giải** | **Loại giá trị** | **Kiểu dữ liệu** | **Miền giá trị** | **Chiều dài** | **Ghi chú** |
| MaPN | Mã phiếu nhập | Bắt buộc | Int | Khóa chính,  FK -> PhieuNhap |  |  |
| MaNVL | Mã nguyên vật liệu | Bắt buộc | Int | Khóa chính,  FK -> NguyenVatLieu |  |  |
| SoLuongNhap | Số lượng nhập | Bắt buộc | Int |  |  |  |

Bảng 3.14. Mô tả thực thể Nhap

## 3.4. Thiết kế cơ sở dữ liệu

Dựa theo mô hình thực thể - kết hợp trước đó đã thiết kế, cơ sở dữ liệu sẽ gồm các bảng sau:

* KhachHang: lưu thông tin khách hàng sử dụng dịch vụ.
* DichVu: danh sách dịch vụ giặt sấy được cung cấp.
* NhanVien: lưu thông tin nhân viên trong cửa hàng.
* CaTruc, Truc: quản lý lịch trực của nhân viên.
* NhaCungCap: lưu thông tin nhà cung cấp nguyên vật liệu cho cửa hàng.
* PhanQuyen, TaiKhoan: phân quyền và đăng nhập hệ thống.
* DonDat: lưu các đơn đặt dịch vụ của khách hàng.
* ChiTietDonDat: chi tiết từng dịch vụ trong đơn đặt.
* HoaDon: thông tin hóa đơn sau khi thanh toán.
* NguyenVatLieu, PhieuNhap, Nhap: quản lý nguyên vật liệu và nhập kho.

## 3.5. Thiết kế giao diện

### 3.5.1. Nguyên tắc thiết kế giao diện

* + Đảm bảo đơn giản, dễ sử dụng, thân thiện với người dùng.
  + Sử dụng màu sắc nhất quán, phù hợp với chủ đề dịch vụ giặt sấy (chủ đạo: xanh dương).
  + Tổ chức bố cục hợp lý, dễ quan sát và thao tác.
  + Hỗ trợ hiển thị tốt trên nhiều kích thước màn hình (responsive).

### 3.5.2. Các giao diện chính

* + **Trang chủ:** hiển thị giới thiệu cửa hàng, quy trình giao nhận hàng, danh sách dịch vụ chính.



Hình 3.1. Giao diện trang chủ

* + **Trang giới thiệu:** hiển thị giới thiệu cửa hàng, quy trình giao nhận hàng và các tiện ích.

Hình 3.2. Trang giới thiệu

* + **Trang dịch vụ:** hiển thị thông tin các dịch vụ của cửa hàng.



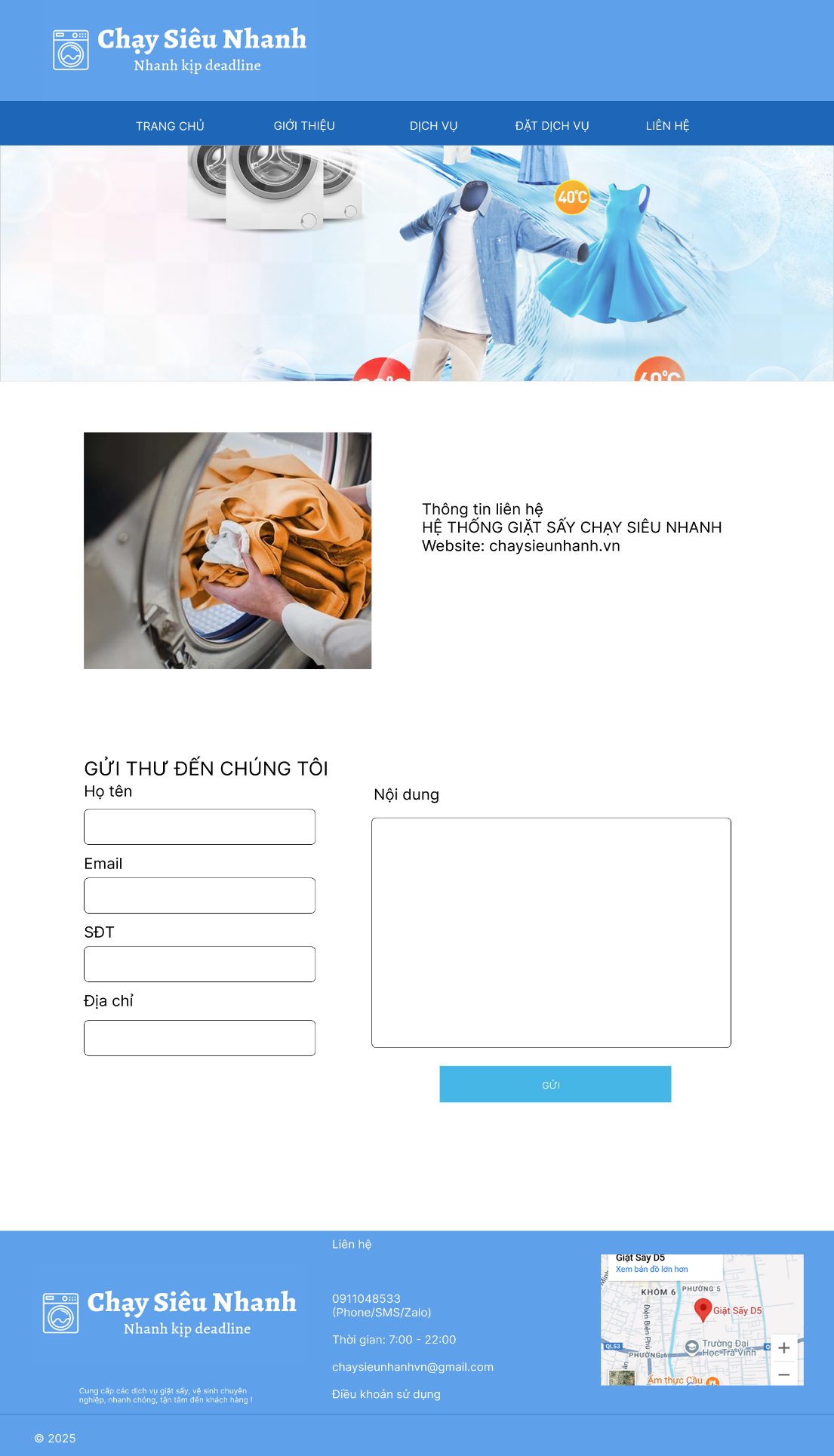
Hình 3.3. Trang dịch vụ

* + **Trang đặt dịch vụ:** cho phép khách hàng chọn dịch vụ, nhập thông tin liên hệ và xác nhận đơn hàng.



Hình 3.4. Trang đặt dịch vụ

* + **Trang liên hệ:** cho phép khách hàng nhập thông tin và nghi vấn, góp ý gửi cho cửa hàng.



Hình 3.5. Trang liên hệ

* + **Trang đăng nhập**

A screenshot of a login screen

AI-generated content may be incorrect.

Hình 3.6. Trang đăng nhập

* + **Trang tổng quan**

A screenshot of a computer

AI-generated content may be incorrect.

Hình 3.7. Trang tổng quan

* + **Trang quản lý đơn hàng**

A screenshot of a computer

AI-generated content may be incorrect.

Hình 3.8. Trang quản lí đơn hàng

* + **Trang quản lý hóa đơn**

A screenshot of a computer

AI-generated content may be incorrect.

Hình 3.9. Trang quản lý hóa đơn

* + **Trang quản lý nhân viên**



Hình 3.10. Trang quản lý nhân viên

## 3.6. Thiết kế xử lý

### 3.6.1. Luồng xử lý chức năng đặt dịch vụ

* Người dùng nhập thông tin vào biểu mẫu trên trang **datdv.html**
* Hệ thống kiểm tra tính hợp lệ (không thiếu trường bắt buộc)
* Gửi dữ liệu về server thông qua HTTP POST (don\_hang.js)
* Server lưu vào bảng DonDat, ChiTietDonDat và phản hồi kết quả thành công hoặc lỗi
* Giao diện hiển thị thông báo đã đặt thành công

### 3.6.2. Xử lý đăng nhập cho nhân viên và admin

* Nhân viên nhập tên tài khoản và mật khẩu
* Server kiểm tra thông tin đăng nhập trong bảng TaiKhoan
* Nếu đúng, lưu session và chuyển hướng đến **trang\_chu\_admin.html** hoặc **dang\_nhap.html**
* Nếu sai thì báo lỗi

### 3.6.3. Xử lý quản lý đơn hàng

* Server truy xuất danh sách đơn hàng từ bảng DonDat, liên kết với KhachHang

### 3.6.4. Xử lý tạo và quản lý hóa đơn

* Nhân viên chọn đơn hàng đã hoàn thành để lập hóa đơn
* Hệ thống truy xuất dịch vụ, tính tổng tiền
* Thêm bản ghi vào bảng HoaDon và liên kết với bảng ChiTietDonDat
* Truy xuất toàn bộ hóa đơn đã lập để hiển thị danh sách trong **hoa\_don.html**

### 3.6.5. Xử lý phân quyền

* Hệ thống phân biệt quyền qua bảng PhanQuyen (ví dụ: quyền admin, quyền nhân viên)
* Trên mỗi route backend đều có kiểm tra quyền trước khi xử lý yêu cầu (middleware kiểm tra phân quyền trong Node.js)

# CHƯƠNG 4: CÀI ĐẶT VÀ TRIỂN KHAI

## 4.1. Mô hình phát triển phần mềm (Đáp ứng tiêu chí L2)

Trong quá trình phát triển hệ thống web quản lý cửa hàng giặt sấy, nhóm lựa chọn mô hình phát triển phần mềm Scrum – một mô hình thuộc phương pháp luận Agile, nhằm đảm bảo tính linh hoạt, khả năng thích nghi cao và khả năng kiểm soát tiến độ qua các vòng lặp**.**

### 4.1.1. Giới thiệu mô hình Scrum

Scrum là một mô hình phát triển phần mềm linh hoạt, lấy người dùng làm trung tâm, tập trung vào việc phát triển theo từng vòng lặp ngắn gọi là Sprint (thường từ 1–4 tuần). Mỗi Sprint là một chu kỳ phát triển hoàn chỉnh, bao gồm lập kế hoạch, thiết kế, hiện thực, kiểm thử và đánh giá.

Trong nhóm, vai trò Scrum được phân chia như sau:

* Product Owner: xác định yêu cầu chức năng từ người dùng, đưa ra danh sách tính năng (Product Backlog).
* Scrum Master: hỗ trợ team vận hành đúng theo quy trình Scrum.
* Development Team: thực hiện thiết kế giao diện, backend, frontend, database, kiểm thử,...

### 4.1.2. Quá trình áp dụng

Nhóm triển khai dự án theo các Sprint như sau:

Sprint Nội dung thực hiện

Sprint 1 Khảo sát yêu cầu, phân tích chức năng chính, thiết kế cơ sở dữ liệu

Sprint 2 Thiết kế giao diện (Figma), xây dựng frontend cơ bản

Sprint 3 Phát triển backend (API đặt dịch vụ, hóa đơn, đăng nhập)

Sprint 4 Tích hợp giao diện - xử lý backend, kiểm thử hệ thống

Sprint 5 Hoàn thiện tài liệu, triển khai bằng Docker, đánh giá kết quả

Mỗi Sprint kết thúc bằng phiên review và retrospective để cải tiến quy trình và chất lượng sản phẩm.

## 4.2. Tính cần thiết của phần mềm dạng phân tán và triển khai trên nền tảng đám mây (Đáp ứng tiêu chí L2 và L6)

### 4.2.1. Tính cần thiết của phần mềm dạng phân tán

Hệ thống được xây dựng theo hướng phân tán, gồm frontend (trình duyệt), backend (Node.js + Express), và cơ sở dữ liệu (MySQL). Cách tiếp cận này mang lại nhiều lợi ích:

* Tách biệt giao diện và xử lý, dễ dàng bảo trì và nâng cấp.
* Mỗi phần có thể phát triển độc lập, ví dụ giao diện người dùng có thể cải tiến mà không ảnh hưởng backend.
* Hỗ trợ mở rộng theo chiều ngang, thích hợp nếu muốn triển khai cho nhiều chi nhánh trong tương lai.

### 4.2.2. Ứng dụng công nghệ đám mây

Để tăng tính linh hoạt và khả năng triển khai đa môi trường, nhóm đã sử dụng Docker để đóng gói hệ thống thành các container:

* Container 1: Backend Node.js
* Container 2: MySQL Database

Điều này giúp:

* Triển khai nhanh trên bất kỳ nền tảng cloud nào (VD: AWS, Google Cloud, Azure,...).
* Dễ bảo trì và mở rộng, chỉ cần cấu hình docker-compose.
* Tiết kiệm thời gian cài đặt, đảm bảo môi trường đồng nhất giữa các thành
* viên phát triển và môi trường sản xuất.

## 4.3. Môi trường phát triển

Hệ thống được xây dựng và phát triển trong môi trường gồm:

* **Hệ điều hành**: Windows 10
* **Trình biên tập mã nguồn**: Visual Studio Code
* **Ngôn ngữ backend**: JavaScript (Node.js + Express.js)
* **Ngôn ngữ frontend**: HTML, CSS, JavaScript
* **Cơ sở dữ liệu**: MySQL
* **Thiết kế cơ sở dữ liệu**: dbdiagram.io (daw.io)
* **Thiết kế giao diện và Use Case**: Figma
* **Quản lý mã nguồn**: GitHub
* **Quản lý công việc**: Jira
* **Docker**: Sử dụng để đóng gói ứng dụng thành các container (Node.js + MySQL)

## 4.4. Cài đặt hệ thống

### 4.4.1. Cài đặt Node.js

* Tải và cài đặt Node.js từ <https://nodejs.org>
* Kiểm tra bằng lệnh:

node -v

npm -v

### 4.4.2. Khởi tạo dự án và cài đặt thư việc

**1. Khởi tạo dự án Node.js:**

mkdir cnpm-webquanlycuahanggiatsay

cd cnpm-webquanlycuahanggiatsay

npm init -y

**2. Cấu hình package.json**

Sau khi khởi tạo dự án bằng lệnh npm init -y, hệ thống sẽ tạo ra

tệp package.json có cấu trúc như sau:

"name": "cnpm-webquanlycuahanggiatsay",

"version": "1.0.0",

* "name": Tên định danh của project. Nếu bạn publish lên npm, nó phải là duy nhất.
* "version": Phiên bản hiện tại, theo dạng Major.Minor.Patch, ví dụ:
  + 1.0.0: lần đầu công bố.
  + 1.1.0: thêm chức năng mới không phá vỡ cũ.
  + 2.0.0: thay đổi lớn, không tương thích cũ.

🡺Hệ thống dùng name và version để định danh dự án.

* Chỉ định file **bắt đầu chạy chương trình**, thường dùng trong require() nếu project này là một module.
* Ở đây là server.js, chứa toàn bộ code backend Node.js của bạn (API, kết nối MySQL,...).
* Khi chạy npm start → sẽ gọi node server.js.

"scripts": {

" test": "echo \"Error: no test specified\" && exit 1",

" start": "node server.js"

}

},

Khi chạy câu lệnh npm start sẽ khởi động backend server Node.js (Express)

"repository": {

" type": "git",

" url": "git+https://github.com/truongtonguyen/cnpm-webquanlycuahanggiatsay.git"

}

},

* Dùng để chỉ nguồn code (nếu người khác muốn đóng góp hay xem).
* git+https là URL chuẩn cho liên kết tới GitHub.

"keywords": [],

"author": "",

"license": "ISC",

"keywords": Dùng để định danh dự án khi publish.

"author": Nên điền tên bạn (ví dụ: "author": "Nguyễn Văn A").

"license": ISC là một loại giấy phép mở (tương tự MIT).

"bugs": {

"url": "https://github.com/truongtonguyen/cnpm-webquanlycuahanggiatsay/issues"

},

"homepage": "https://github.com/truongtonguyen/cnpm-webquanlycuahanggiatsay#readme",

* "bugs.url": Link để người dùng báo lỗi trên GitHub.
* "homepage": Link trang chủ tài liệu, ở đây trỏ tới phần README của project.

"dependencies": {

"cors": "^2.8.5",

"express": "^5.1.0",

"mysql2": "^3.14.2"

}

Các thư viện cần thiết khi chạy

Server.js:

const express = require("express");

const bodyParser = require("body-parser");

const cors = require("cors");

const mysql = require("mysql2");

const app = express();

const PORT = 3000;

// Kết nối database MySQL

const db = mysql.createConnection({

host: "localhost",

user: "root", // thay bằng user thật nếu có

password: "user", // thay bằng mật khẩu nếu có

database: "quanlycuahanggs" // tên cơ sở dữ liệu của bạn

});

db.connect((err) => {

if (err) {

console.error("❌ Lỗi kết nối CSDL:", err);

process.exit(1);

}

console.log("✅ Đã kết nối MySQL thành công");

});

// Middleware

app.use(cors());

app.use(bodyParser.json());

app.get('/hoa-don', (req, res) => {

const sql = `

SELECT

hd.MaHD,

dd.TenKH,

dv.TenDichVu,

hd.SoLuong,

dv.DonGia,

dv.DonGia \* hd.SoLuong AS ThanhTien,

hd.TrangThai

FROM HoaDon hd

JOIN DonDat dd ON hd.MaDonDat = dd.MaDonDat

JOIN DichVu dv ON dd.MaDichVu = dv.MaDichVu

db.query(sql, (err, rows) => {

if (err) {

console.error("Lỗi SQL: ", err);

return res.status(500).json({ error: 'Lỗi truy vấn CSDL' });

}

res.json(rows);

});

});

// -------------------- ĐĂNG NHẬP --------------------

const adminUser = {

username: "admin",

password: "123456"

};

app.post("/api/login", (req, res) => {

const { username, password } = req.body;

if (username === adminUser.username && password === adminUser.password) {

res.json({ success: true });

} else {

res.json({ success: false, message: "Sai tài khoản hoặc mật khẩu" });

}

});

// -------------------- ĐẶT ĐƠN HÀNG --------------------

app.post('/dat-lich', (req, res) => {

const { hoten, sdt, email, diachi, lichhen, dichvu, ghichu } = req.body;

const maDonDat = `DD${Math.floor(Math.random() \* 10000).toString().padStart(4, '0')}`;

const maKH = `KH${Math.floor(Math.random() \* 10000).toString().padStart(4, '0')}`;

// Tạo khách hàng mới

const insertKH = `

INSERT INTO KhachHang (MaKH, TenKH, DiaChi, SDT\_KH, EmailKH)

VALUES (?, ?, ?, ?, ?) `;

db.query(insertKH, [maKH, hoten, diachi, sdt, email], (errKH) => {

if (errKH) {

console.error("❌ Lỗi tạo khách hàng:", errKH);

return res.status(500).send("Lỗi khi thêm khách hàng");

}

// Tìm mã dịch vụ theo tên dịch vụ

const queryDV = `SELECT MaDichVu FROM DichVu WHERE TenDichVu = ?`;

db.query(queryDV, [dichvu], (errDV, rows) => {

if (errDV || rows.length === 0) {

console.error("❌ Lỗi dịch vụ:", errDV);

return res.status(400).send("Dịch vụ không hợp lệ");

}

const maDV = rows[0].MaDichVu;

const insertDon = `

INSERT INTO DonDat (MaDonDat, TenKH, SDT\_KH, NgayDat, MaDichVu, GhiChu, TrangThai, MaKH)

VALUES (?, ?, ?, ?, ?, ?, 'Chưa xử lý', ?)’:

db.query(insertDon, [maDonDat, hoten, sdt, lichhen, maDV, ghichu, maKH], (errDon) => {

if (errDon) {

console.error("❌ Lỗi đặt hàng:", errDon);

return res.status(500).send("Lỗi khi đặt hàng");

}

Res.send("✔️ Đặt lịch thành công!");

});

});

});

});});

// -------------------- XEM ĐƠN HÀNG --------------------

app.get('/don-hang', (req, res) => {

const sql = `

SELECT

dd.MaDonDat,

dd.TenKH AS HoTen,

dv.TenDichVu,

dd.NgayDat AS NgayHen,

dd.SDT\_KH AS SoDienThoai,

dd.GhiChu,

dd.TrangThai

FROM DonDat dd

JOIN DichVu dv ON dd.MaDichVu = dv.MaDichVu

WHERE dd.TrangThai = 'Chưa xử lý' `;

db.query(sql, (err, result) => {

if (err) {

console.error("❌ Lỗi truy vấn đơn hàng:", err);

return res.status(500).send("Lỗi truy vấn đơn hàng");

}

res.json(result);

});

});

// -------------------- TRANG TỔNG QUAN -------------------

app.get('/tong-quan', (req, res) => {

const sql = `

SELECT

(SELECT COUNT(\*) FROM DonDat) AS tongDon,

(SELECT COUNT(\*) FROM HoaDon) AS daTaoHD,

(SELECT COUNT(\*) FROM HoaDon WHERE TrangThai = 'Chưa thanh toán') AS chuaTT;

db.query(sql, (err, rows) => {

if (err) {

console.error("❌ Lỗi tổng quan:", err);

return res.status(500).send("Lỗi tổng quan");

}

res.json(rows[0]);

});

});

app.get('/don-hang', (req, res) => {

const sql = `

SELECT dd.MaDon, dd.HoTen, dv.TenDV, dd.NgayHen, dd.SoDienThoai, dd.GhiChu, dd.TrangThai

FROM DonDat dd

JOIN DichVu dv ON dd.MaDV = dv.MaDV

WHERE dd.TrangThai = 'Chưa xử lý'

`;

db.query(sql, (err, rows) => {

if (err) {

console.error("❌ Lỗi lấy đơn hàng:", err);

return res.status(500).send("Lỗi đơn hàng");

}

res.json(rows);

});

});

// -------------------- TẠO HÓA ĐƠN --------------------

app.post('/tao-hoa-don', (req, res) => {

const { maDon, soLuong } = req.body;

console.log("📦 Nhận từ client:", req.body);

// Kiểm tra trùng hóa đơn

db.query('SELECT \* FROM HoaDon WHERE MaDonDat = ?', [maDon], (errCheck, rows) => {

if (errCheck) {

console.error("❌ Lỗi kiểm tra hóa đơn:", errCheck);

return res.status(500).send("Lỗi kiểm tra");

}

if (rows.length > 0) {

return res.status(400).send("Đơn hàng này đã có hóa đơn!");

}

// Tiếp tục insert

const insertHD = `

INSERT INTO HoaDon (MaDonDat, SoLuong, NgayLap, TrangThai)

VALUES (?, ?, CURDATE(), 'Chưa thanh toán') `;

db.query(insertHD, [maDon, soLuong], (err, result) => {

if (err) {

console.error("❌ Lỗi tạo hóa đơn:", err);

return res.status(500).send("Lỗi tạo hóa đơn");

}

db.query(

`UPDATE DonDat SET TrangThai = 'Đã tạo hóa đơn' WHERE MaDonDat = ?`,

[maDon],

(err2) => {

if (err2) {

console.error("❌ Lỗi cập nhật đơn hàng:", err2);

return res.status(500).send("Lỗi cập nhật đơn hàng");

}

res.send("Đã tạo hóa đơn thành công!");

}

);

});

});

});

//Lấy danh sách nhân viên

app.get("/api/nhanvien", (req, res) => {

const sql = "SELECT MaNV, TenNV, GioiTinh, NgaySinh, SDT\_NV FROM NhanVien";

db.query(sql, (err, results) => {

if (err) {

console.error("❌ Lỗi truy vấn danh sách nhân viên:", err);

return res.status(500).json({ error: "Lỗi truy vấn dữ liệu" });

}

res.json(results);

});

});

// Thêm nhân viên mới

app.post("/api/nhanvien", (req, res) => {

const { tenNV, gioiTinh, ngaySinh, sdtNV } = req.body;

const maNV = `NV${Math.floor(Math.random() \* 10000).toString().padStart(4, '0')}`;

const sql = `

INSERT INTO NhanVien (MaNV, TenNV, GioiTinh, NgaySinh, SDT\_NV)

VALUES (?, ?, ?, ?, ?)`;

db.query(sql, [maNV, tenNV, gioiTinh, ngaySinh, sdtNV], (err, result) => {

if (err) {

console.error("❌ Lỗi thêm nhân viên:", err);

return res.status(500).json({ error: "Lỗi khi thêm nhân viên" });

}

res.json({ success: true, maNV });

});

});

// Xoá nhân viên theo mã

app.delete("/api/nhanvien/:maNV", (req, res) => {

const maNV = req.params.maNV;

const sql = "DELETE FROM NhanVien WHERE MaNV = ?";

db.query(sql, [maNV], (err, result) => {

if (err) {

console.error("❌ Lỗi khi xoá nhân viên:", err);

return res.status(500).json({ error: "Lỗi khi xoá nhân viên" });

}

if (result.affectedRows === 0) {

return res.status(404).json({ error: "Không tìm thấy nhân viên" });

}

res.json({ success: true });

});

});

// -------------------- LẤY DANH SÁCH HÓA ĐƠN --------------------

app.get('/hoa-don', (req, res) => {

app.get('/hoa-don', (req, res) => {

const sql = `

SELECT hd.MaHD, dd.HoTen, dv.TenDV, hd.SoLuong, hd.DonViTinh, hd.DonGia,

(hd.SoLuong \* hd.DonGia) AS ThanhTien, hd.TrangThai

FROM HoaDon hd

JOIN DonDat dd ON hd.MaDon = dd.MaDonDat

JOIN DichVu dv ON dd.MaDV = dv.MaDV

`;

db.query(sql, (err, rows) => {

if (err) {

console.error("❌ Lỗi lấy hóa đơn:", err);

return res.status(500).send("Lỗi hóa đơn");

}

res.json(rows);

});

});

// -------------------- CHẠY SERVER --------------------

app.listen(PORT, () => {

console.log(`🚀 Server đang chạy tại http://localhost:${PORT}`);

});

* **Framework chính**: Express.js
* **Cơ sở dữ liệu**: MySQL
* **Cổng server**: 3000
* **Thư viện sử dụng**:
  + express: tạo server và routing
  + body-parser: xử lý dữ liệu JSON
  + cors: cho phép frontend truy cập
  + mysql2: kết nối và truy vấn cơ sở dữ liệu MySQL

**3. Kết nối đến MySQL**

Ứng dụng kết nối tới MySQL bằng thông tin cấu hình sau:

* Host: localhost
* User: root
* Password: user
* Database: quanlycuahanggs

Nếu kết nối thành công, in ra thông báo "Đã kết nối MySQL thành công". Nếu lỗi, hiển thị lỗi và kết thúc chương trình.

**4. Các chức năng API chính**

**a. Đăng nhập hệ thống (Admin)**

* **Đường dẫn**: POST /api/login
* **Dữ liệu gửi lên**: username, password

Cấu trúc chung

Kiểm tra: so sánh với thông tin admin tĩnh (admin / 123456)

Trả về: { success: true } nếu đúng, ngược lại { success: false }.

b. Đặt lịch dịch vụ (Khách hàng)

Đường dẫn: POST /dat-lich

Chức năng:

Sinh mã khách hàng và mã đơn đặt.

Thêm khách hàng mới vào bảng KhachHang.

Lấy mã dịch vụ tương ứng theo tên.

Thêm đơn đặt vào bảng DonDat.

Trạng thái mặc định của đơn đặt: 'Chưa xử lý'

Trả về: thông báo thành công hoặc lỗi.

c. Xem danh sách đơn hàng chưa xử lý

Đường dẫn: GET /don-hang

Chức năng:

Truy vấn danh sách đơn đặt chưa được xử lý.

Kết hợp với bảng DichVu để hiển thị tên dịch vụ.

d. Thống kê tổng quan

Đường dẫn: GET /tong-quan

Chức năng:

Đếm tổng số đơn đặt (DonDat)

Đếm tổng số hóa đơn (HoaDon)

Đếm số hóa đơn chưa thanh toán (HoaDon WHERE TrangThai = 'Chưa thanhtoán')

Trả về: JSON chứa ba số liệu tổng hợp.

e. Tạo hóa đơn từ đơn đặt

Đường dẫn: POST /tao-hoa-don

Chức năng:

Nhận mã đơn và số lượng.

Kiểm tra đơn hàng đã có hóa đơn chưa.

Nếu chưa có, thêm vào bảng HoaDon.

Cập nhật trạng thái đơn đặt thành 'Đã tạo hóa đơn'.

Trả về: thành công hoặc lỗi.

f. Quản lý nhân viên

- Lấy danh sách nhân viên

Đường dẫn: GET /api/nhanvien

Trả về: danh sách tất cả nhân viên trong bảng NhanVien.

- Thêm nhân viên

Đường dẫn: POST /api/nhanvien

Chức năng:

Nhận thông tin nhân viên từ client.

Tạo mã nhân viên tự động (NVxxxx).

Thêm vào bảng NhanVien.

Trả về: { success: true, maNV } nếu thành công.

- Xoá nhân viên

Đường dẫn: DELETE /api/nhanvien/:maNV

Chức năng:

Nhận mã nhân viên qua URL.

Xoá nhân viên tương ứng trong bảng.

Kiểm tra nếu không tìm thấy thì trả lỗi.

g. Lấy danh sách hóa đơn

Đường dẫn: GET /hoa-don

Chức năng:

Truy vấn danh sách hóa đơn từ bảng HoaDon, kết hợp DonDat và DichVu.

Tính thành tiền = số lượng \* đơn giá.

Trả về: danh sách hóa đơn với đầy đủ thông tin.

5. Khởi động server

Lệnh: app.listen(PORT, () => {...})

Chạy trên cổng: 3000

Hiển thị: "Server đang chạy tại http://localhost:3000"

## 4.5. Triển khai

**4.5.1. Đóng gói ứng dụng bằng Docker**

Để đảm bảo hệ thống có thể chạy ổn định trên mọi môi trường, dự án

được đóng gói bằng Docker, bao gồm hai thành phần chính:

* + Backend (Node.js)
  + Cơ sở dữ liệu (MySQL)

**4.3.2. Tạo Dockerfile cho Backend**

# Sử dụng image Node.js chính thức

FROM node:18

# Tạo thư mục làm việc trong container

WORKDIR /app

# Sao chép file package.json và package-lock.json

COPY package\*.json ./

# Cài đặt dependencies

RUN npm install

# Sao chép toàn bộ mã nguồn vào container

COPY . .

# Mở cổng 3000

EXPOSE 3000

# Lệnh chạy ứng dụng

CMD ["npm", "start"]

**4.3.3. Tạo Docker Compose**

version: "3.8"

services:

backend:

build: .

container\_name: backend

ports:

- "3000:3000"

environment:

- MYSQL\_HOST=db

- MYSQL\_USER=root

- MYSQL\_PASSWORD=user

- MYSQL\_DATABASE=QuanLyCuaHangGS

depends\_on:

- db

networks:

- app-network

db:

image: mysql:5.7

container\_name: mysql

restart: always

environment:

MYSQL\_ROOT\_PASSWORD: user

MYSQL\_DATABASE: QuanLyCuaHangGS

ports:

- "3306:3306"

volumes:

- db-data:/var/lib/mysql

networks:

- app-network

volumes:

db-data:

networks:

app-network:

**4.3.4. Khởi chạy hệ thống**

Chạy các lệnh sau để triển khai hệ thống:

docker-compose up –build

Kết quả:

* Backend chạy ở địa chỉ: http://localhost:3000
* MySQL có thể truy cập từ container backend qua host db với:
* user: root
* password: user
* database: QuanLyCuaHangGS

**4.3.5. Kiểm tra kết nối và hoạt động**

Sau khi khởi chạy thành công:

* Truy cập trình duyệt để kiểm tra các giao diện frontend
* Kiểm tra API bằng các công cụ như Postman, hoặc frontend tương tác
* Kiểm tra cơ sở dữ liệu bằng công cụ như MySQL Workbench hoặc CLI

# CHƯƠNG 5: KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

## 5.1. Kết luận

Đề tài đã hoàn thành việc xây dựng hệ thống web quản lý cửa hàng giặt sấy, hỗ trợ đầy đủ các chức năng: Đặt dịch vụ, lập hóa đơn, quản lý nhân viên, danh mục dịch vụ

Hệ thống giúp: Tăng tính tự động hóa Giảm thất thoát dữ liệu Nâng cao hiệu quả vận hành cho cửa hàng Giao diện thân thiện, dễ thao tác cho cả khách hàng và nhân viên.

## 5.2. Ưu và nhược điểm

### 5.2.1. Ưu điểm

* Hệ thống có giao diện đơn giản, dễ sử dụng, thân thiện với người dùng.
* Chức năng rõ ràng, hỗ trợ đầy đủ cho quá trình vận hành cửa hàng giặt sấy.
* Cơ sở dữ liệu được thiết kế có cấu trúc hợp lý, đảm bảo dễ bảo trì và mở rộng.
* Triển khai bằng Docker giúp dễ dàng đóng gói và cài đặt hệ thống.
* Có thể dễ dàng tích hợp với các công cụ khác như GitHub, Jira để quản lý mã nguồn và tiến độ.

### 5.2.2. Nhược điểm

* Giao diện chưa thực sự hiện đại, chưa hỗ trợ responsive trên thiết bị di động.
* Hệ thống chưa hỗ trợ phân quyền nâng cao cho các vai trò khác nhau.
* Chưa có chức năng thống kê, báo cáo chuyên sâu phục vụ quản lý.
* Chưa triển khai được trên docker.

## 5.3. Hướng phát triển

Trong tương lai, hệ thống có thể được phát triển thêm các chức năng như:

* Xây dựng chức năng phân quyền chi tiết cho các vai trò: quản lý, nhân viên, khách hàng.
* Cập nhật giao diện theo hướng responsive, tối ưu cho thiết bị di động và máy tính bảng.
* Thêm chức năng thống kê doanh thu theo ngày, tháng, quý.
* Hỗ trợ in hóa đơn trực tiếp từ hệ thống.
* Phát triển ứng dụng khách hàng giúp khách có thể theo dõi đơn giặt, tra cứu lịch sử hóa đơn và đặt lịch từ xa.
* Tích hợp thanh toán trực tuyến (Momo, VNPay, v.v.) Xây dựng ứng dụng di động (Mobile App) để khách hàng đặt dịch vụ nhanh hơn Bổ sung chức năng giao – nhận tại nhà (theo lịch trình).
* Tích hợp thông báo tự động qua Email/SMS/Zalo Cải tiến báo cáo thống kê: doanh thu, dịch vụ sử dụng nhiều nhất, v.v.
* Tối ưu hóa bảo mật tài khoản và quản lý phân quyền sâu hơn.

# TÀI LIỆU THAM KHẢO

1. Pressman, R. S. (2010). Software Engineering: A Practitioner's Approach (7th Edition).

2. Sommerville, I. (2016). Software Engineering (10th Edition).