**TRƯỜNG ĐẠI HỌC SÀI GÒN**



**KHOA CÔNG NGHỆ THÔNG TIN**

Ảnh có chứa vòng tròn, biểu tượng, Phông chữ, văn bản

Mô tả được tạo tự động

**Công nghệ lập trình hiện đại**

**XÂY DỰNG HỆ THỐNG QUẢN LÝ QUÁN ĂN**

**FASTFOOD – EXPRESS**

*Sinh viên thực hiện:*

Lý Quốc Thành – 3121411194

Khưu Minh Khoa – 3121411102

Nguyễn Hoàng Chương – 3121411030

*Giảng viên hướng dẫn*:

Thành phố Hồ Chí Minh, tháng 3 năm 2025

# **DANH MỤC CÁC BẢNG BIỂU**

# **DANH MỤC CÁC HÌNH ẢNH**

**MỤC LỤC**

[**DANH MỤC CÁC BẢNG BIỂU** 2](#_Toc195024584)

[**DANH MỤC CÁC HÌNH ẢNH** 3](#_Toc195024585)

[**MỤC LỤC** 4](#_Toc195024586)

[**LỜI MỞ ĐẦU** 5](#_Toc195024587)

[**CHƯƠNG 1: ĐỀ CƯƠNG, TỔNG QUAN ĐỀ TÀI** 6](#_Toc195024588)

[**1.1.** **Tổng quan đề tài** 6](#_Toc195024589)

[**1.2.** **Phân công công việc** 6](#_Toc195024590)

[**1.3.**  **Thực hiện dự án** 7](#_Toc195024591)

[**CHƯƠNG 2: CÁC VẤN ĐỀ LIÊN QUAN** 8](#_Toc195024592)

[**2.1.** **Nghiên cứu công nghệ:** 8](#_Toc195024593)

[**2.1.1.** **Node.js** 8](#_Toc195024594)

[**2.1.2.** **React.js** 8](#_Toc195024595)

[**2.1.3.** **Cơ sở dữ liệu: MongoDB** 8](#_Toc195024596)

[**2.1.4.** **Công cụ hỗ trợ khác** 8](#_Toc195024597)

[**2.2.** **Lý do lựa chọn công nghệ** 9](#_Toc195024598)

[**2.3.** **Kế hoạch nghiên cứu:** 9](#_Toc195024599)

[**2.4.** **Tài liệu tham khảo** 11](#_Toc195024600)

[**2.5.** **Phân tích và đánh giá** 11](#_Toc195024601)

# **LỜI MỞ ĐẦU**

Trong thời đại công nghệ số hiện nay, việc ứng dụng các hệ thống phần mềm vào hoạt động kinh doanh ngày càng trở nên phổ biến và cần thiết, đặc biệt trong lĩnh vực dịch vụ ăn uống. Các quán ăn, nhà hàng ngày càng chú trọng đến việc tối ưu quy trình đặt món, thanh toán và quản lý hoạt động nhằm nâng cao chất lượng phục vụ và hiệu quả kinh doanh.

Đề tài *"Xây dựng hệ thống quản lý quán ăn FastFood - Express "* sử dụng công nghệ Node.js và React được thực hiện với mục tiêu xây dựng một hệ thống phần mềm hiện đại, thân thiện với người dùng, hỗ trợ việc quản lý quán ăn từ khâu đặt món, xử lý đơn hàng, đến quản lý nhân viên và thống kê doanh thu.  
Thông qua quá trình thực hiện đề tài, nhóm không chỉ áp dụng kiến thức chuyên môn đã học mà còn nâng cao kỹ năng làm việc nhóm, tư duy thiết kế hệ thống, cũng như khả năng giải quyết các vấn đề thực tế.

# **CHƯƠNG 1: ĐỀ CƯƠNG, TỔNG QUAN ĐỀ TÀI**

## **Tổng quan đề tài**

Hệ thống quản lý quán ăn Fast Food là một giải pháp phần mềm nhằm số hóa quy trình hoạt động của quán ăn, giúp tiết kiệm thời gian, giảm thiểu sai sót và nâng cao trải nghiệm người dùng.  
Đề tài tập trung vào việc xây dựng một hệ thống web gồm hai phần chính:

* **Frontend**: Giao diện người dùng được xây dựng bằng React, cung cấp trải nghiệm mượt mà, thân thiện và dễ sử dụng cho cả khách hàng và nhân viên.
* **Backend**: Hệ thống xử lý nghiệp vụ được phát triển bằng Node.js kết hợp với cơ sở dữ liệu MongoDB, đảm bảo tốc độ xử lý nhanh và khả năng mở rộng tốt.

Chức năng chính của hệ thống bao gồm:

* Quản lý thực đơn (món ăn, danh mục)
* Đặt món và xử lý đơn hàng
* Quản lý người dùng (khách hàng, nhân viên, quản lý)
* Thống kê doanh thu, báo cáo
* Tích hợp thanh toán (có thể mở rộng trong tương lai)

## **Phân công công việc**

Nhóm gồm 3 thành viên, mỗi người đảm nhận các vai trò chính như sau:

| **Họ và tên** | **Nhiệm vụ chính** |
| --- | --- |
| Khưu Minh Khoa | Phát triển giao diện người dùng bằng React; kết nối API từ frontend |
| Lý Quốc Thành | Xây dựng backend với Node.js; triển khai các API xử lý nghiệp vụ |
| Nguyễn Hoàng Chương | Thiết kế cơ sở dữ liệu; hỗ trợ backend; kiểm thử và viết tài liệu |

Ngoài ra, các thành viên phối hợp chặt chẽ trong quá trình thiết kế kiến trúc hệ thống, kiểm thử sản phẩm và hoàn thiện báo cáo cuối kỳ.

## **1.3. Thực hiện dự án**

Quá trình thực hiện được chia thành các giai đoạn rõ ràng với thời gian phù hợp như sau:

| **Giai đoạn** | **Thời gian** | **Nội dung công việc chính** |
| --- | --- | --- |
| Phân tích yêu cầu | Tuần 1 - Tuần 2 | Khảo sát mô hình quán ăn, xác định chức năng cần thiết |
| Thiết kế hệ thống | Tuần 3 - Tuần 4 | Thiết kế kiến trúc hệ thống, sơ đồ luồng dữ liệu, giao diện mẫu |
| Triển khai frontend | Tuần 5 - Tuần 6 | Xây dựng giao diện người dùng bằng React |
| Phát triển backend | Tuần 7 - Tuần 8 | Xây dựng server Node.js, tạo và xử lý API kết nối với frontend |
| Kết nối và hoàn thiện hệ thống | Tuần 9 - Tuần 10 | Tích hợp frontend và backend, xử lý các chức năng chính |
| Kiểm thử và sửa lỗi | Tuần 11 - Tuần 12 | Thực hiện kiểm thử chức năng, hiệu năng; chỉnh sửa lỗi |
| Tổng hợp, viết báo cáo | Tuần 13 - Tuần 14 | Viết tài liệu báo cáo, chuẩn bị trình bày sản phẩm cuối kỳ |

# **CHƯƠNG 2: CÁC VẤN ĐỀ LIÊN QUAN**

* 1. **Nghiên cứu công nghệ:**

Trong đề tài xây dựng hệ thống quản lý quán fast food, nhóm lựa chọn những công nghệ hiện đại, phổ biến trong phát triển web hiện nay. Việc nghiên cứu, lựa chọn công nghệ phù hợp đóng vai trò then chốt giúp hệ thống đạt được các mục tiêu như: tốc độ xử lý nhanh, giao diện thân thiện, dễ bảo trì và mở rộng về sau.

* + 1. **Node.js**

Node.js là một nền tảng phía máy chủ được xây dựng trên JavaScript runtime của Chrome. Với cơ chế xử lý bất đồng bộ (non-blocking I/O), Node.js cho phép xây dựng các ứng dụng có hiệu năng cao và phản hồi nhanh, rất phù hợp cho các hệ thống có nhiều request đồng thời như hệ thống đặt món ăn trực tuyến.

* + 1. **React.js**

React.js là thư viện JavaScript được phát triển bởi Facebook, cho phép xây dựng giao diện người dùng dưới dạng các component (thành phần). Với kiến trúc SPA (Single Page Application), trang web sẽ không cần tải lại toàn bộ nội dung mỗi khi người dùng chuyển trang, từ đó nâng cao trải nghiệm sử dụng.

Ngoài ra, React hỗ trợ state management, tái sử dụng component, có thể kết hợp với Redux hoặc Context API để quản lý trạng thái ứng dụng phức tạp, phù hợp với các hệ thống quản lý có nhiều vai trò người dùng như: nhân viên, quản lý, khách hàng,…

* + 1. **Cơ sở dữ liệu: MongoDB**

Tuỳ vào yêu cầu cụ thể, nhóm có thể sử dụng MongoDB (NoSQL). MongoDB phù hợp khi dữ liệu có cấu trúc linh hoạt, thay đổi thường xuyên, ví dụ như thông tin đơn hàng, người dung.

Cả hai cơ sở dữ liệu này đều dễ dàng tích hợp với Node.js thông qua các thư viện như Mongoose (cho MongoDB) giúp thao tác dữ liệu hiệu quả.

* + 1. **Công cụ hỗ trợ khác**
* **Postman**: Được sử dụng để kiểm thử API một cách trực quan. Các request (GET, POST, PUT, DELETE) có thể mô phỏng như client thật, kiểm tra response trả về để đảm bảo backend hoạt động chính xác.
* **Figma**: Là công cụ thiết kế giao diện online, cho phép nhóm tạo wireframe, prototype, mockup giao diện người dùng từ sớm, giúp đồng bộ giữa thiết kế và lập trình.
* **Git & GitHub**: Git được dùng để kiểm soát phiên bản mã nguồn, còn GitHub hỗ trợ làm việc nhóm, quản lý các nhánh phát triển (branch), theo dõi bug, ghi chú công việc và kiểm tra mã nguồn qua pull request.
  1. **Lý do lựa chọn công nghệ**

Các công nghệ trên đều là mã nguồn mở, có cộng đồng hỗ trợ lớn, nhiều tài liệu hướng dẫn, giúp nhóm dễ học hỏi, giải quyết sự cố nhanh chóng. Ngoài ra, việc lựa chọn React cho frontend và Node.js cho backend (JavaScript full stack) giúp đồng bộ ngôn ngữ lập trình, giảm chi phí đào tạo, đồng thời giúp hệ thống dễ bảo trì và phát triển trong tương lai.

* 1. **Kế hoạch nghiên cứu:**

Để đảm bảo quá trình triển khai đồ án được thực hiện một cách hiệu quả, nhóm đã xây dựng một kế hoạch nghiên cứu và phát triển hệ thống rõ ràng, chia theo từng giai đoạn cụ thể. Kế hoạch này vừa giúp nhóm dễ dàng kiểm soát tiến độ công việc, vừa đảm bảo hệ thống được xây dựng đầy đủ về chức năng, ổn định về hiệu suất và dễ dàng mở rộng về sau.

**Giai đoạn 1: Khởi tạo & Phân tích yêu cầu**

Trong giai đoạn này, nhóm tiến hành các hoạt động khởi động dự án bao gồm:

* Tổ chức buổi họp **kick-off meeting** để thảo luận và phân chia công việc.
* Tiến hành **thu thập yêu cầu** từ người dùng mục tiêu (quản lý quán, nhân viên, khách hàng).
* **Phân tích yêu cầu** để xác định rõ các chức năng cần thiết, phạm vi hệ thống, và công nghệ sẽ sử dụng.

**Giai đoạn 2: Thiết kế hệ thống**

Sau khi đã hiểu rõ yêu cầu, nhóm bước vào giai đoạn thiết kế:

* **Thiết kế kiến trúc hệ thống**, hướng đến mô hình microservices hoặc chia module hợp lý giữa backend và frontend.
* **Thiết kế API**: Xác định các endpoint chính phục vụ chức năng như đặt món, quản lý đơn hàng, thanh toán,...
* **Thiết kế UI/UX**: Sử dụng Figma để tạo wireframe, mockup giao diện người dùng trực quan.
* **Thiết kế cơ sở dữ liệu**: Xác định cấu trúc bảng, mối quan hệ giữa các bảng (nếu dùng MySQL) hoặc thiết kế collection (nếu dùng MongoDB).

**Giai đoạn 3: Phát triển Backend**

Backend được phát triển qua hai giai đoạn:

* **Phần 1**: Xây dựng các API cơ bản cho User Service, Menu Service. Viết Unit Test cho các API này để đảm bảo độ ổn định.
* **Phần 2**: Tiếp tục xây dựng các chức năng nâng cao như Order Service, Payment, Notification. Tiến hành kiểm thử API và sửa lỗi (debug & fix bugs).

**Giai đoạn 4: Phát triển Frontend**

Frontend cũng được chia làm 2 phần:

* **Phần 1**: Khởi tạo dự án React, tạo component cơ bản và thiết lập giao diện người dùng sơ khởi.
* **Phần 2**: Kết nối với API backend, kiểm thử giao diện và xử lý logic người dùng. Tối ưu giao diện theo phản hồi.

**Giai đoạn 5: Thiết lập môi trường & DevOps**

Để dễ dàng triển khai và kiểm thử, nhóm sử dụng:

* **Docker**: Tạo Dockerfile cho cả frontend và backend.
* **Docker Compose**: Kết hợp các service thành hệ thống chạy đồng bộ.
* **CI/CD**: Thiết lập pipeline triển khai tự động qua GitHub Actions.

**Giai đoạn 6: Tích hợp & Kiểm thử**

Sau khi hoàn thành các phần chính:

* Tích hợp toàn bộ hệ thống: backend, frontend, cơ sở dữ liệu.
* **Kiểm thử end-to-end**, kiểm thử chức năng, kiểm thử hiệu suất, đảm bảo hệ thống hoạt động đúng như yêu cầu.
* **Sửa lỗi** phát sinh và tối ưu hiệu suất toàn hệ thống.

**Giai đoạn 7: Triển khai & Demo**

* **Triển khai** hệ thống thực tế trên môi trường production (local server hoặc cloud).
* Tổ chức buổi demo minh họa quá trình sử dụng hệ thống từ đầu đến cuối.

**Giai đoạn 8: Tài liệu & Bảo trì**

* Viết tài liệu hướng dẫn cài đặt, sử dụng hệ thống.
* Ghi chú các lỗi, hướng cải tiến, cập nhật phiên bản trong tương lai.
  1. **Tài liệu tham khảo**

Trong quá trình nghiên cứu và thực hiện đồ án, nhóm đã tham khảo nhiều tài liệu và nguồn kiến thức khác nhau nhằm đảm bảo tính chính xác và hiệu quả trong việc triển khai hệ thống. Các tài liệu tham khảo bao gồm:

* **Tài liệu chính thức của Node.js**: Cung cấp thông tin chi tiết về môi trường chạy JavaScript phía máy chủ, cách quản lý module, xử lý bất đồng bộ bằng callback, promise và async/await.  
  https://nodejs.org/en/docs
* **React.js Documentation**: Tài liệu chính thức giúp nhóm hiểu rõ về cách xây dựng component, quản lý state, props và sử dụng hooks.  
  https://reactjs.org/docs/getting-started.html
* **MongoDB Manual:** Hướng dẫn cách thiết kế cơ sở dữ liệu, viết truy vấn, và tối ưu hoá lưu trữ dữ liệu.
  + MongoDB: <https://www.mongodb.com/docs/manual/>
* **Postman Documentation**: Tài liệu giúp kiểm thử API nhanh chóng và hiệu quả.  
  https://learning.postman.com/
* **Figma**: Công cụ thiết kế UI/UX, hỗ trợ tạo wireframe và prototype cho giao diện người dùng.  
  <https://www.figma.com/>
* **Tài liệu DevOps (Docker, CI/CD)**:
  + Docker: https://docs.docker.com/
  + GitHub Actions: <https://docs.github.com/en/actions>
  1. **Phân tích và đánh giá**
     1. Phân tích chức năng

Hệ thống quản lý quán fast food được xây dựng nhằm hỗ trợ hoạt động kinh doanh hiệu quả hơn, tự động hóa quy trình đặt món, thanh toán, và quản lý kho. Các chức năng chính được xác định như sau:

* **Quản lý thực đơn**: Cho phép người quản lý thêm, sửa, xóa các món ăn, phân loại theo danh mục (đồ uống, món chính, món phụ…).
* **Đặt món**: Khách hàng có thể xem thực đơn, chọn món và đặt hàng trực tuyến.
* **Quản lý đơn hàng**: Nhân viên có thể xem danh sách đơn hàng, cập nhật trạng thái (đã xác nhận, đang chế biến, hoàn thành, đã giao…).
* **Thanh toán**: Hỗ trợ thanh toán online và offline, lưu lại lịch sử giao dịch.
* **Thống kê doanh thu**: Hiển thị báo cáo doanh thu theo ngày/tháng, top món bán chạy,...
* **Quản lý người dùng**: Phân quyền giữa quản lý, nhân viên, và khách hàng.
  + 1. Phân tích yêu cầu phi chức năng
* Hiệu năng: Hệ thống phải xử lý được nhiều đơn hàng cùng lúc mà không bị gián đoạn.
* Khả năng mở rộng: Có thể dễ dàng thêm chức năng mới như giảm giá, voucher, loyalty points…
* Bảo mật: Quản lý quyền truy cập, bảo mật API và dữ liệu người dùng (sử dụng JWT, HTTPS…).
* Thân thiện với người dùng: Giao diện đơn giản, dễ thao tác cho cả nhân viên và khách hàng.
* Khả năng triển khai linh hoạt: Có thể chạy trên máy tính cục bộ hoặc triển khai lên cloud/server.
  + 1. Phân tích kiến trúc hệ thống

Ví dụ luồng đặt món:

1. Người dùng truy cập giao diện, xem thực đơn.
2. Chọn món và thêm vào giỏ hàng.
3. Xác nhận đơn hàng → frontend gửi API đặt hàng lên backend.
4. Backend xử lý và lưu đơn vào database.
5. Nhân viên nhận thông báo đơn mới trên giao diện quản lý.
6. Sau khi hoàn tất, đơn được cập nhật trạng thái và thanh toán.
   * 1. Đánh giá ban đầu

Dựa trên kế hoạch đã triển khai, nhóm nhận thấy hệ thống đáp ứng tốt các yêu cầu đề ra:

* Giao diện mượt, thân thiện, hoạt động ổn định.
* API phản hồi nhanh, xử lý được luồng nghiệp vụ phức tạp.
* Cơ sở dữ liệu được tổ chức chặt chẽ, dễ truy vấn và mở rộng.

Tuy nhiên, hệ thống vẫn còn một số hạn chế như:

* Chưa tối ưu hiệu suất khi chạy trên thiết bị cấu hình thấp.
* Một số chức năng nâng cao như phân tích hành vi khách hàng, gợi ý món chưa được tích hợp.