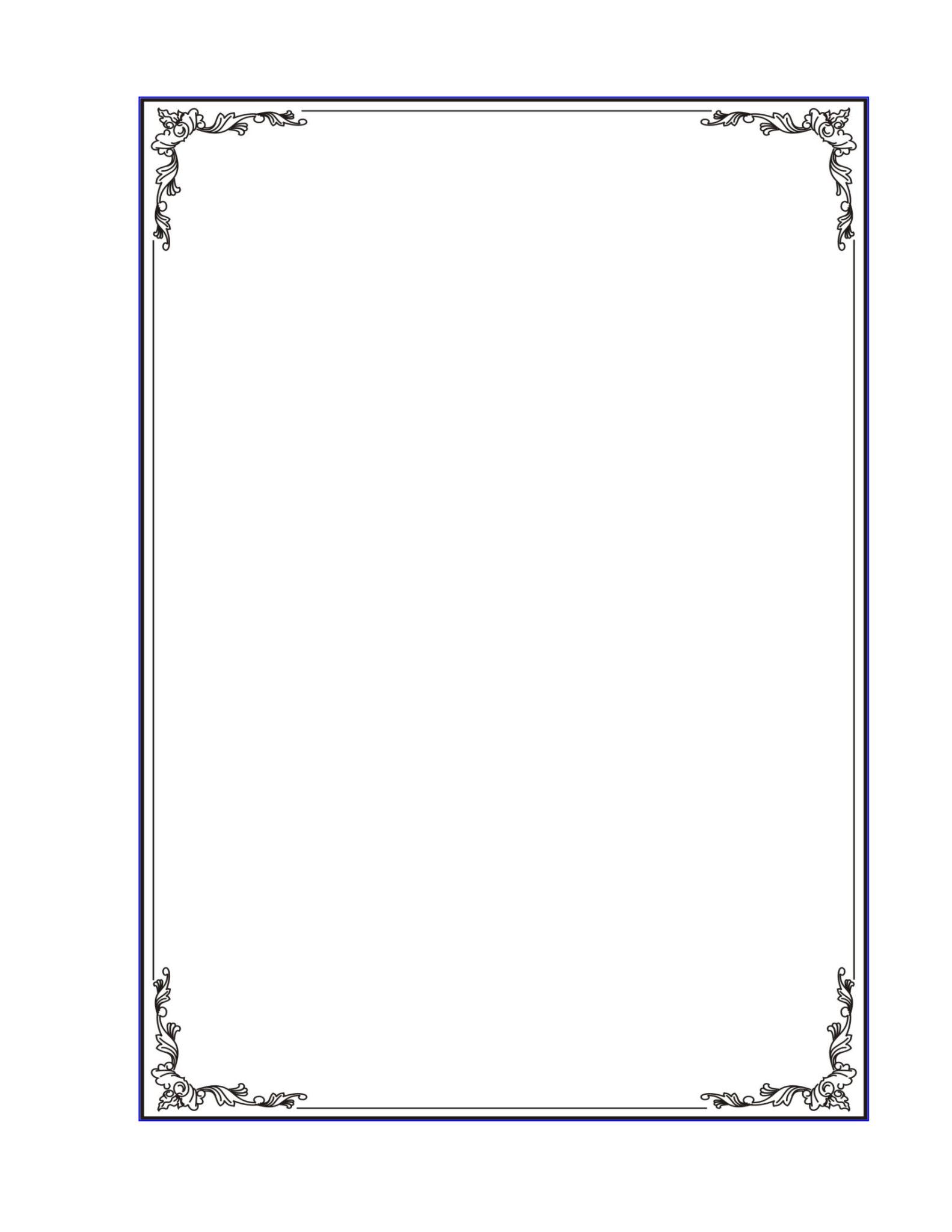
****

**HỌC VIỆN NGÂN HÀNG**

**KHOA HỆ THỐNG THÔNG TIN QUẢN LÝ**

****

**BÁO CÁO BÀI TẬP LỚN**

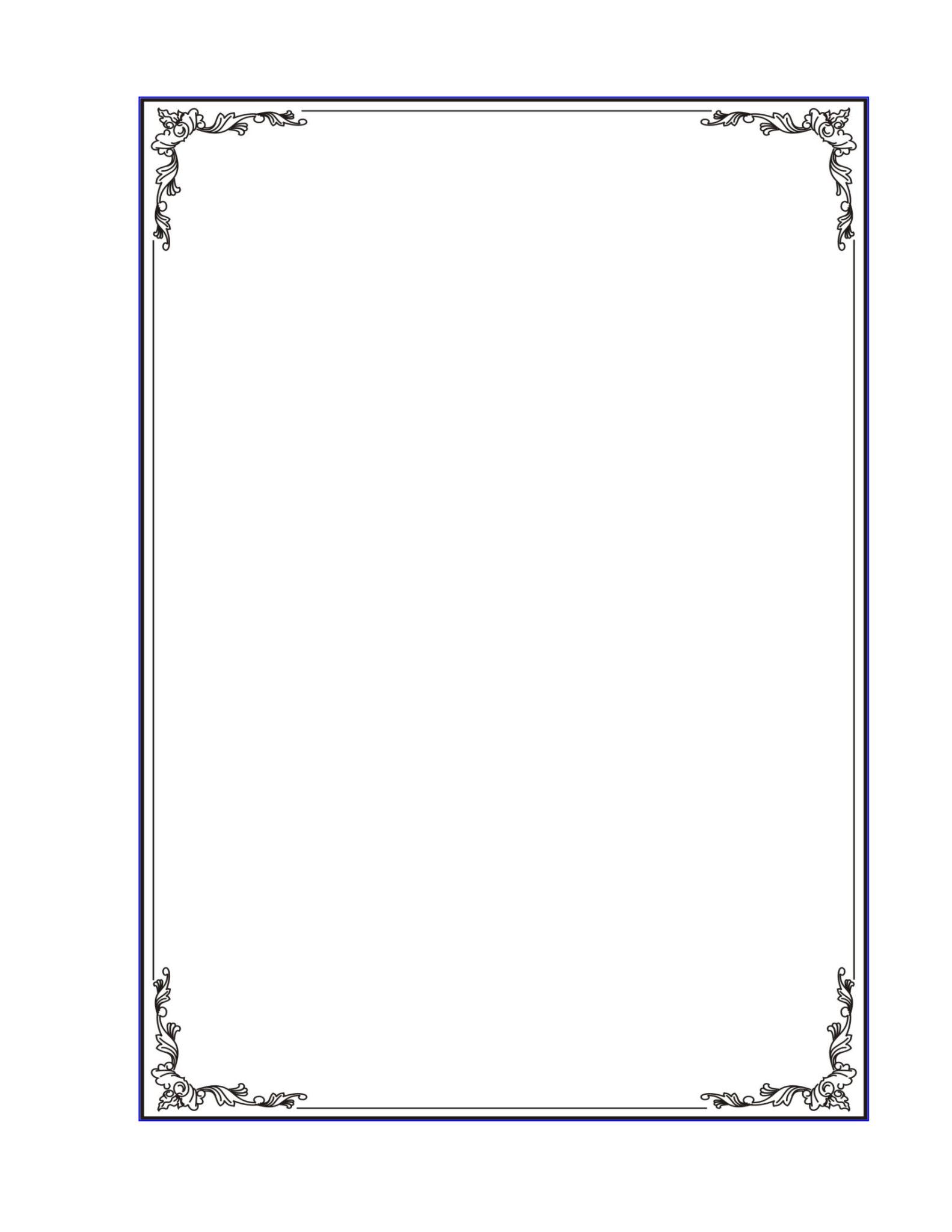
**MÔN KIỂM THỬ VÀ ĐẢM BẢO CHẤT LƯỢNG PHẦN MỀM**

**ĐỀ TÀI: Kiểm thử và đảm bảo chất lượng cho**

**website thương mại điện tử SigmaTech**

|  |  |
| --- | --- |
| **Giảng viên hướng dẫn** | : Thầy Nguyễn Ngọc Quang |
| **Lớp** | : 241IT02A01 |
| **Nhóm** | : 7 |

**Hà Nội – 2024**

****

**HỌC VIỆN NGÂN HÀNG**

**KHOA HỆ THỐNG THÔNG TIN QUẢN LÝ**

****

**BÁO CÁO BÀI TẬP LỚN**

**MÔN KHO DỮ LIỆU VÀ KINH DOANH THÔNG MINH**

**ĐỀ TÀI: ỨNG DỤNG DATA WAREHOUSE VÀ BI TRONG VIỆC TỐI**

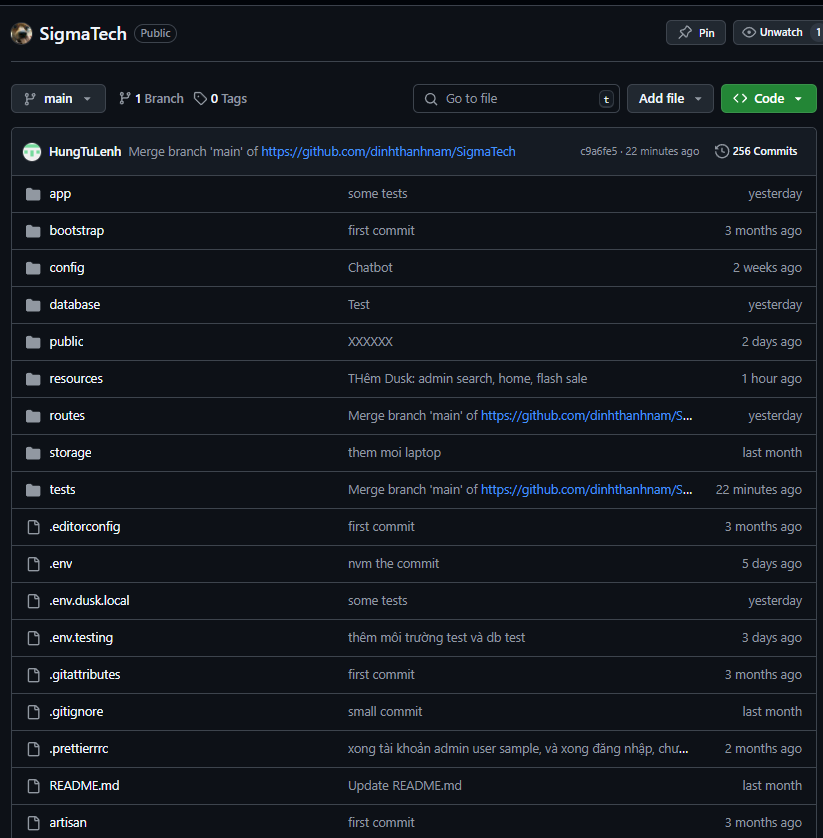
**ƯU HOÁ CHIẾN LƯỢC KINH DOANH CỦA NETFLIX**

|  |  |  |  |
| --- | --- | --- | --- |
| **Thành viên** | **Mã sinh viên** | **Tỉ lệ đóng góp** | **Xác nhận** |
| Đinh Thành Nam  (Nhóm trưởng) | 24A4042599 | 20% |  |
| Nguyễn Duy Hưng | 24A4042596 | 21.5% |  |
| Nguyễn Thế Nghĩa | 24A4042602 | 21.5% |  |
| Trần A Khang | 24A4040141 | 10.5% |  |
| Trương Ngọc Minh Khôi | 24A4040167 | 18.5% |  |

**Hà Nội – 2024**

**HOẠT ĐỘNG NHÓM**

**1. Theo dõi tiến độ trên Github**

**Link git:** https://github.com/dinhthanhnam/Sigmatech.git

**2. Hoạt động nhóm**

Nhóm hoạt động thường xuyên qua kênh discord để trao đổi và phát triển bài tập

# LỜI CẢM ƠN

Lời đầu tiên, nhóm 7 xin gửi lời cảm ơn đến Học viện Ngân hàng và khoa Công nghệ thông tin và kinh tế số đã tạo điều kiện cho chúng em và các bạn sinh viên có cơ hội được học tập trong một môi trường năng động, sáng tạo, tạo điều kiện nỗ lực và phát huy hết tiềm năng của mình. Nhóm xin gửi lời cảm ơn đặc biệt đến thầy Đinh Trọng Hiếu đã dành sự tâm huyết và nhiệt tình giảng dạy, giúp chúng em học được một môn học có ứng dụng cao trong ngành học và nghiệp vụ sau này, đáp ứng nhu cầu thực tiễn của các nhà tuyển dụng về kiến thức, kỹ năng cần có.

Trong học phần này, chúng em được tiếp xúc với các công cụ tạo dashboard, báo cáo, học về cơ bản của thiết kế kho dữ liệu, và được thực hiện một dự án thực tế thiết kế dashboard cho doanh nghiệp cụ thể. Đây là một trải nghiệm thú vị và là một kĩ năng quan trọng cho chúng em sau này.

Chúng em xin chân thành cảm ơn!

Nhóm sinh viên thực hiện

Nhóm 7

# LỜI CAM ĐOAN

Nhóm 12 xin cam đoan báo cáo này là sản phẩm nghiên cứu và thực hành của nhóm. Bài báo cáo đảm bảo tính liêm chính trong học tập, không đạo văn, gian lận, bịa đặt. Các thông tin tham khảo được trích dẫn nguồn đầy đủ và minh bạch.

Nhóm 12 xin chịu toàn bộ trách nhiệm nếu bài báo cáo vi phạm các điều trên.

Hà Nội, ngày 14 tháng 12 năm 2024

Đại diện nhóm 12

# MỤC LỤC

# DANH MỤC BẢNG, HÌNH VẼ

# LỜI MỞ ĐẦU

Trong thời đại cách mạng công nghệ 4.0, việc quản lý và khai thác dữ liệu trở thành một yếu tố quan trọng đối với sự phát triển bền vững của các doanh nghiệp. Netflix, một trong những công ty hàng đầu trong ngành công nghiệp giải trí trực tuyến, đã chứng tỏ sức mạnh của dữ liệu trong việc tối ưu hóa hoạt động kinh doanh và nâng cao trải nghiệm người dùng.

Dựa trên nhu cầu không ngừng tăng về việc phân tích dữ liệu, việc xây dựng một kho dữ liệu toàn diện, hiệu quả để quản lý, lưu trữ và truy xuất dữ liệu là vô cùng cần thiết. Kho dữ liệu này không chỉ giúp Netflix tối ưu hóa quá trình khuyến nghị phim cho người dùng, mà còn hỗ trợ việc phân tích hành vi người tiêu dùng, dự đoán xu hướng thị trường và ra quyết định chiến lược.

Trong bài tập lớn này, chúng em sẽ tập trung vào việc thiết kế và xây dựng một hệ thống kho dữ liệu cho Netflix. Hệ thống này sẽ được tối ưu hóa cho các mục tiêu như cải thiện hiệu quả truy xuất thông tin, hỗ trợ phân tích kinh doanh và cung cấp các công cụ báo cáo thông minh. Qua đó, bài tập không chỉ giúp người học củng cố kiến thức về kho dữ liệu, mà còn mang lại cái nhìn thực tế về ứng dụng công nghệ này trong ngành giải trí số.

# CHƯƠNG 1. TỔNG QUAN

## 1.1. Tổng quan về đảm bảo chất lượng phần mềm

### 1.1.1. Định nghĩa chất lượng phần mềm

- Định nghĩa theo IEEE (1991):

+ Định nghĩa 1: Chất lượng phần mềm là một mức độ mà một hệ thống, thành phần hệ thống hay tiến trình đáp ứng được yêu cầu đã được đặc tả.

+ Định nghĩa 2: Chất lượng phần mềm là mức độ mà một hệ thống, thành phần hệ thống hay tiến trình đáp ứng được yêu cầu và sự mong đợi của khách hàng hay người sử dụng.

- Định nghĩa theo Pressman: Chất lượng phần mềm là sự phù hợp của các yêu cầu cụ thể về hiệu năng và chức năng, các tiêu chuẩn phát triển phần mềm được ghi lại rõ ràng bằng tài liệu với các đặc tính ngầm định của tất cả các phần mềm được phát triển chuyên nghiệp.

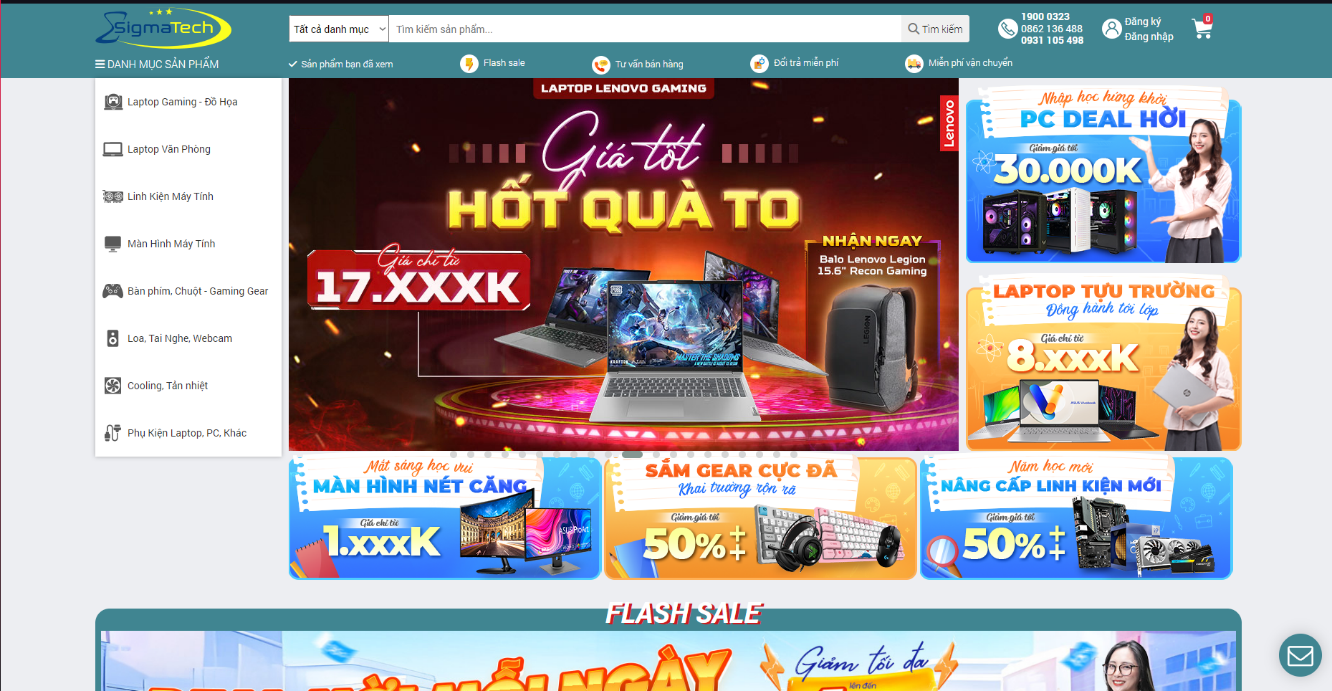
### 1.1.2. Định nghĩa về đảm bảo chất lượng phần mềm

Định nghĩa theo Daniel Galin: Đảm bảo chất lượng phần mềm là một tập hợp các hành động cần thiết được lên kế hoạch một cách hệ thống để cung cấp đầy đủ niềm tin rằng quá trình phát triển phần mềm phù hợp để thành lập các yêu cầu chức năng kỹ thuật cũng như các yêu cầu quản lý theo lịch trình và hoạt động trong giới hạn ngân sách.

## 1.2. Tổng quan về đề tài

### 1.2.1. Phát biểu bài toán

Trong thời đại công nghệ số, các website bán hàng điện tử đóng vai trò quan trọng trong việc cung cấp sản phẩm và dịch vụ đến người tiêu dùng một cách nhanh chóng và tiện lợi. SigmaTech, một website chuyên bán linh kiện điện tử phục vụ nhu cầu chơi game trên máy tính và laptop do chính nhóm tự thiết kế, cần đảm bảo rằng hệ thống của mình hoạt động ổn định, chính xác và đáp ứng tốt các yêu cầu của người dùng.



Ảnh 1: Website bán hàng điện tử SigmaTech

Vấn đề chính đặt ra là làm thế nào để phát hiện và khắc phục sớm các lỗi tiềm ẩn trong hệ thống, từ đó nâng cao trải nghiệm người dùng. Hệ thống SigmaTech bao gồm nhiều tính năng phức tạp như tìm kiếm và lọc sản phẩm, quản lý giỏ hàng, xử lý đơn hàng, thanh toán trực tuyến, quản lý tài khoản người dùng… Một lỗi nhỏ trong bất kỳ tính năng nào cũng có thể gây mất uy tín, ảnh hưởng đến doanh thu và lòng tin của khách hàng.

Do đó, mục tiêu của đề tài này là thực hiện kiểm thử và đảm bảo chất lượng cho website SigmaTech, tập trung vào việc phát hiện và sửa lỗi trong các chức năng cốt lõi. Thông qua việc áp dụng các chiến lược kiểm thử phù hợp, đề tài hướng đến việc đảm bảo hệ thống hoạt động chính xác, an toàn và thân thiện với người dùng, đồng thời đáp ứng được các tiêu chuẩn về chất lượng phần mềm. Kết quả cuối cùng là một báo cáo chi tiết về các lỗi được phát hiện, giải pháp khắc phục và đánh giá hiệu quả của quá trình kiểm thử.

### 1.2.2. Lý do chọn đề tài

Chúng tôi chọn đề tài này vì nhận thấy kiểm thử và đảm bảo chất lượng phần mềm là yếu tố then chốt trong việc phát triển và duy trì uy tín của các website bán hàng điện tử. Một trang web với tính năng không ổn định hoặc thiếu bảo mật có thể gây mất niềm tin nơi người dùng, ảnh hưởng trực tiếp đến doanh thu và hình ảnh thương hiệu.

Ngoài ra, lĩnh vực kiểm thử phần mềm đòi hỏi sự chính xác, sáng tạo và kỹ năng giải quyết vấn đề, rất phù hợp để áp dụng kiến thức học tập vào thực tiễn. Bằng cách thực hiện kiểm thử trên SigmaTech, chúng tôi không chỉ giải quyết bài toán kỹ thuật thực tế mà còn góp phần nâng cao trải nghiệm người dùng trong lĩnh vực thương mại điện tử.

* Đề tài này là cơ hội để nhóm nghiên cứu rèn luyện và ứng dụng các kỹ thuật kiểm thử hiện đại, đồng thời khám phá tầm quan trọng của chất lượng phần mềm đối với sự thành công của doanh nghiệp.

# CHƯƠNG 2. CƠ SỞ LÝ THUYẾT

## 2.1. Xây dựng kế hoạch đảm bảo chất lượng phần mềm

### 2.1.1. Kiểm soát tiến độ hoàn thành dự án

### 2.1.2. Kiểm soát tiến độ nội bộ và tác nhân tham gia bên ngoài

Dự án có sự tham gia của nhiều thành phần như nhóm phát triển nội bộ, nhóm kiểm thử và các bên liên quan từ bên ngoài. Việc phối hợp và quản lý tiến độ giữa các bên cần được thực hiện đồng bộ thông qua:

* Lịch họp định kỳ để cập nhật tiến độ và xử lý vấn đề phát sinh.
* Các báo cáo công việc hàng tuần và đánh giá hiệu quả công việc.
* Xác định trách nhiệm cụ thể cho từng thành viên và đội nhóm tham gia.

### 2.1.3. Kiểm soát lịch trình và tài nguyên dự án

Đảm bảo việc phân bổ thời gian, nhân lực và các tài nguyên khác hợp lý trong suốt quá trình thực hiện dự án:

* Ước tính thời gian và tài nguyên cần thiết cho từng hạng mục.
* Theo dõi việc sử dụng tài nguyên thông qua các công cụ quản lý.
* Đề xuất biện pháp điều chỉnh khi có sự chậm trễ hoặc thiếu hụt tài nguyên.

### 2.1.4. Đánh giá chất lượng của sản phẩm phần mềm

Đánh giá chất lượng sản phẩm sẽ dựa trên 4 các tiêu chí sau: chất lượng chức năng, hiệu suất, bảo mật và khả năng sử dụng.

## 2.2. Kiểm thử chức năng

## 2.3. Kiểm thử phi chức năng

### 2.3.1. Load Testing

- Kiểm thử tải (Load Testing): Kiểm tra khả năng của hệ thống khi có nhiều người dùng truy cập đồng thời.

- Phương pháp: Giả lập số lượng lớn người dùng hoặc các yêu cầu truy cập đồng thời.

- Kết quả mong đợi: Hệ thống đáp ứng tốt với x người dùng đồng thời.

### 2.3.2. Stress Testing

- Kiểm thử áp lực (Stress Testing): Đánh giá khả năng của hệ thống khi chịu tải cao hơn mức tối đa có thể chấp nhận được.

- Phương pháp: Tăng dần số lượng người dùng hoặc yêu cầu cho đến khi hệ thống gặp lỗi hoặc không thể đáp ứng.

- Kết quả mong đợi:

* Xác định "điểm vỡ" của hệ thống.
* Đánh giá khả năng hệ thống phục hồi sau khi gặp tải cao.

### 2.3.3. Usability Testing

- Kiểm thử khả năng sử dụng (Usability Testing): Đánh giá mức độ dễ sử dụng của hệ thống từ quan điểm của người dùng.

- Phương pháp: Thu thập phản hồi của người dùng về tính dễ hiểu, tiện lợi và thẩm mỹ của giao diện.

- Kết quả mong đợi: Giao diện dễ sử dụng, chức năng hoạt động mượt mà, người dùng không gặp vấn đề trong quá trình trải nghiệm.

### 2.3.4. Security Testing

- Kiểm thử bảo mật (Security Testing): Đảm bảo rằng hệ thống an toàn và bảo vệ các dữ liệu của người dùng.

- Phương pháp:

* Kiểm tra xác thực người dùng, bảo mật mật khẩu và mã hóa dữ liệu.
* Kiểm tra các lỗ hổng phổ biến như SQL Injection, XSS.
* Kiểm tra quyền truy cập của các vai trò người dùng khác nhau (Admin, Client).

- Kết quả mong đợi: Phát hiện các lỗ hổng và khắc phục trước khi triển khai website.

### 2.3.5. Portability Testing

- Kiểm thử tính di động (Portability Testing): Đánh giá khả năng của hệ thống để di chuyển, triển khai và hoạt động trên nhiều môi trường khác nhau.

- Phương pháp:

* Kiểm thử phần mềm trên các trình duyệt (Chrome, Firefox, Edge, Safari) và hệ điều hành khác nhau (Windows, macOS, Android, iOS).
* Kiểm tra khả năng thích ứng của giao diện trên các độ phân giải màn hình khác nhau (Desktop, Tablet, Mobile).

- Kết quả mong đợi: Phần mềm hoạt động ổn định và không gặp lỗi giao diện hoặc tính năng trên các nền tảng khác nhau.

# CHƯƠNG 2: KẾ HOẠCH PHÁT TRIỂN VÀ ĐẢM BẢO PHẦN MỀM CHO WEBSITE SIGMATECH

## 2.1. Lên kế hoạch phát triển

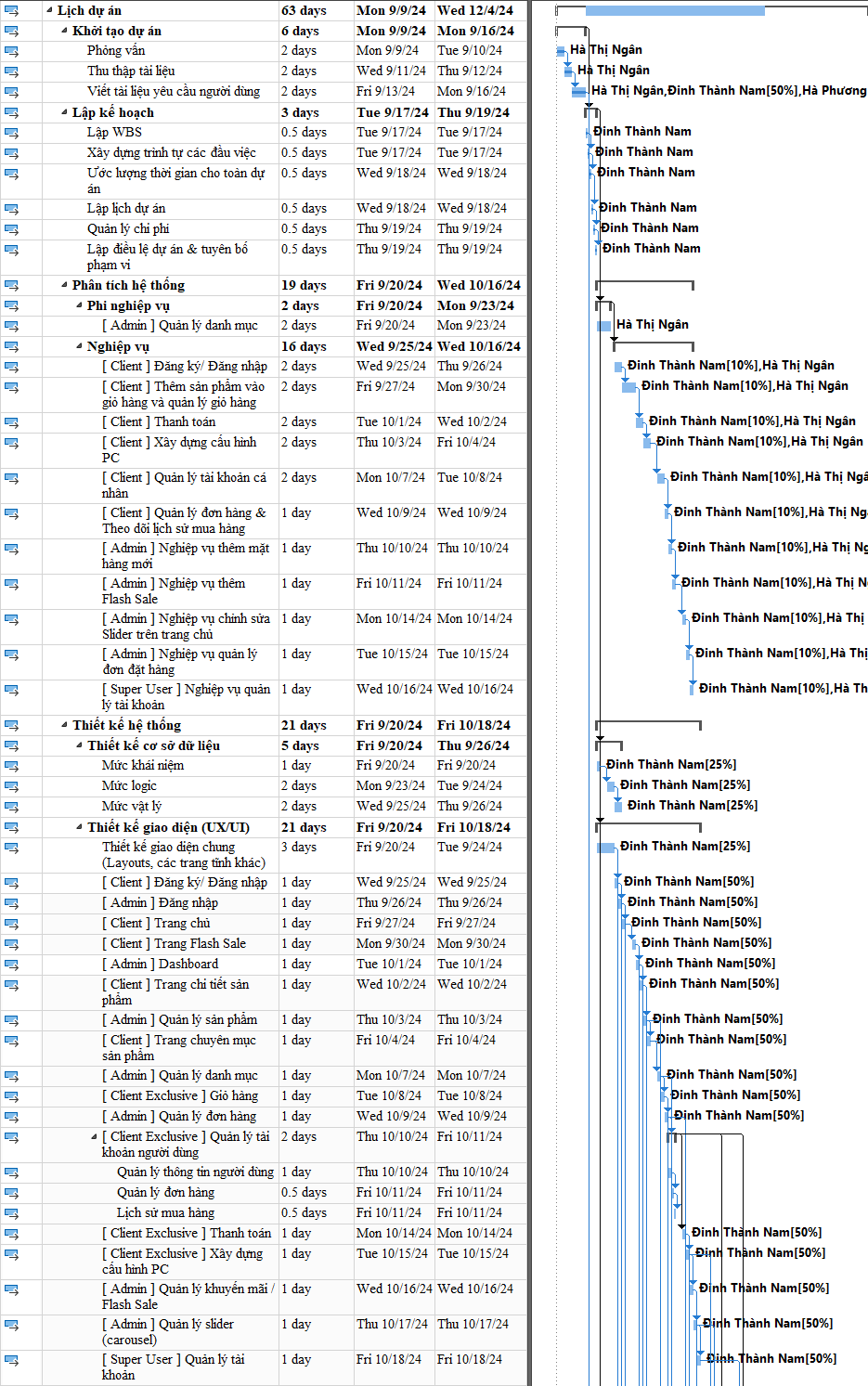
Để đảm bảo chất lượng phần mềm, tiến độ của từng giai đoạn trong dự án cần được kiểm soát chặt chẽ.

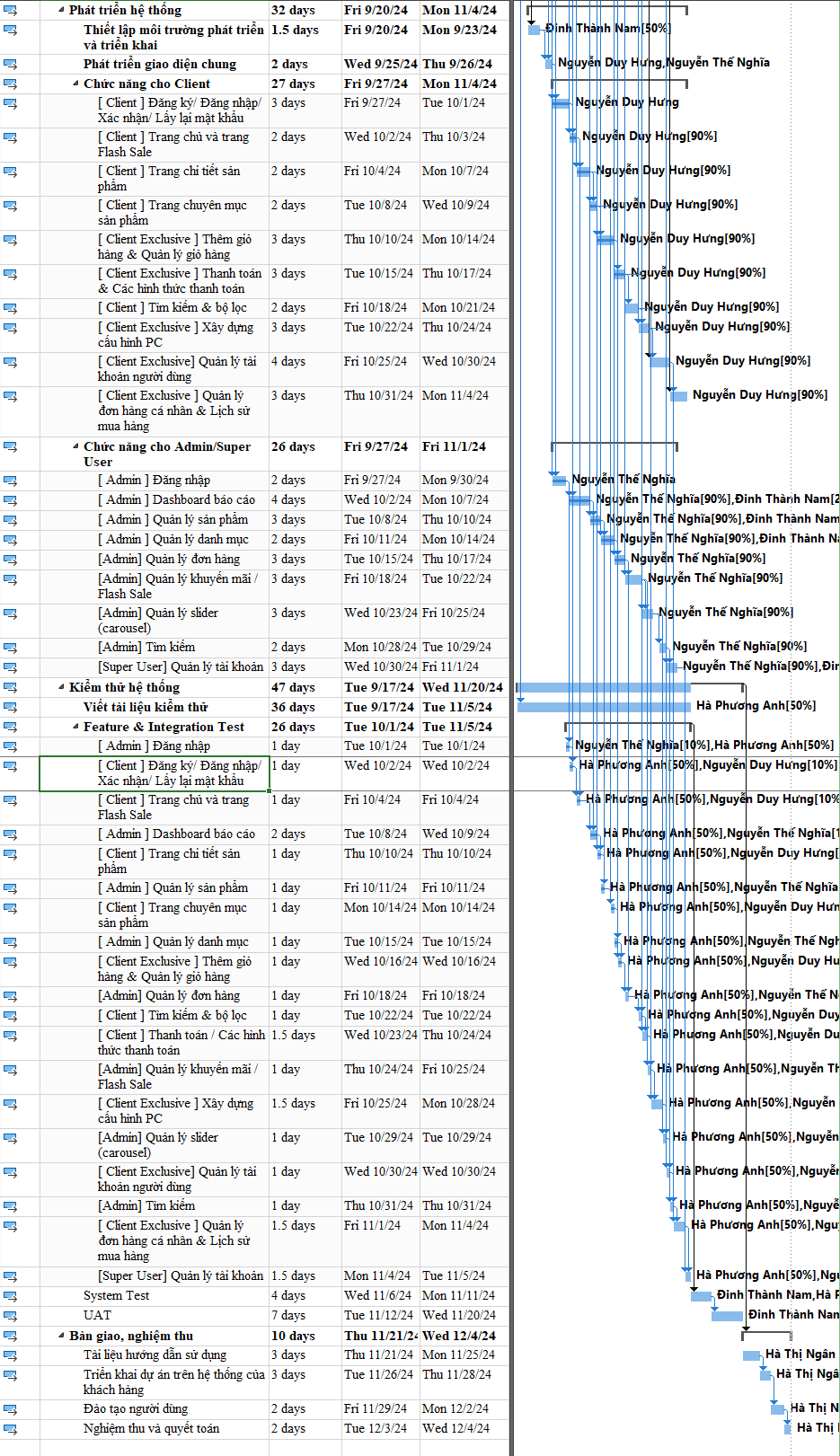
Trước tiên hết, cần phải đảm bảo các quy trình quản lý 1 dự án công nghệ thông tin, bao gồm: quản lý phạm vi, quản lý thời gian, quản lý chi phí, quản lý chất lượng, quản lý nhân lực, quản lý rủi ro.

Thời gian của dự án được chia thành các giai đoạn cụ thể:

* Giai đoạn khởi tạo: Thu thập và xác định đầy đủ các yêu cầu chức năng và phi chức năng.
* Giai đoạn lập kế hoạch: Xác định công việc, ước lượng thời gian dự án, lập lịch dự án.
* Giai đoạn thiết kế: Đảm bảo tài liệu thiết kế đáp ứng yêu cầu và khả thi về mặt kỹ thuật.
* Giai đoạn phát triển: Kiểm tra chất lượng mã nguồn trong từng vòng lặp phát triển.
* Giai đoạn kiểm thử: Thực hiện các loại kiểm thử như kiểm thử chức năng, hiệu năng, bảo mật, ...
* Giao đoạn triển khai và bàn giao: Là giai đoạn triển khai trên nền tảng cho khách hàng, bàn giao và nghiệm thu sản phẩm.

Nhóm cụ thể hoá các đầu việc và kế hoạch quản lý thời gian cho dự án “xây dựng website thương mại điện tử SigmaTech” bằng lịch biểu dự án trong Microsoft Project.





## 2.2. Triết lý và kỷ luật của đội ngũ phát triển

### 2.2.1. Triết lý phát triển

**Triết lý phát triển phần mềm** là một tập hợp các nguyên lý và phương pháp luận giúp đội ngũ phát triển phần mềm xây dựng sản phẩm hiệu quả, chất lượng cao và có khả năng thay đổi linh hoạt theo yêu cầu của khách hàng và thị trường.

**Agile Scrum** là một khung làm việc trong phương pháp Agile, tập trung vào việc phát triển sản phẩm qua các vòng lặp ngắn (sprint), mỗi vòng lặp có thời gian từ 1 đến 4 tuần. Trong mỗi sprint, nhóm phát triển sẽ hoàn thành một lượng công việc cụ thể và cung cấp phần mềm có thể sử dụng được. Phương pháp này thúc đẩy sự hợp tác, giao tiếp và khả năng thích ứng linh hoạt với yêu cầu thay đổi từ khách hàng.

**Sprint** là chu kỳ phát triển ngắn, mỗi lần kéo dài từ 1 đến 4 tuần. Mỗi sprint bao gồm các bước:

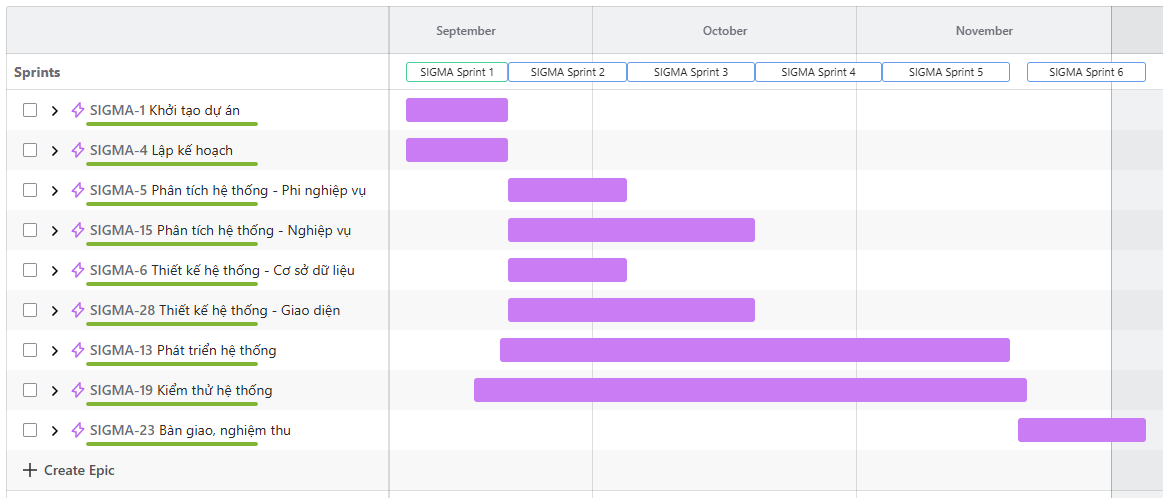
* **Sprint Planning:** Xác định các tính năng sẽ được phát triển trong sprint.
* **Daily Scrum (Họp hàng ngày):** Các thành viên trong nhóm báo cáo công việc đã làm, kế hoạch công việc và các vấn đề gặp phải.
* **Sprint Review:** Kiểm tra kết quả đạt được trong sprint.
* **Sprint Retrospective:** Đánh giá lại quy trình làm việc và tìm cách cải tiến cho sprint tiếp theo.

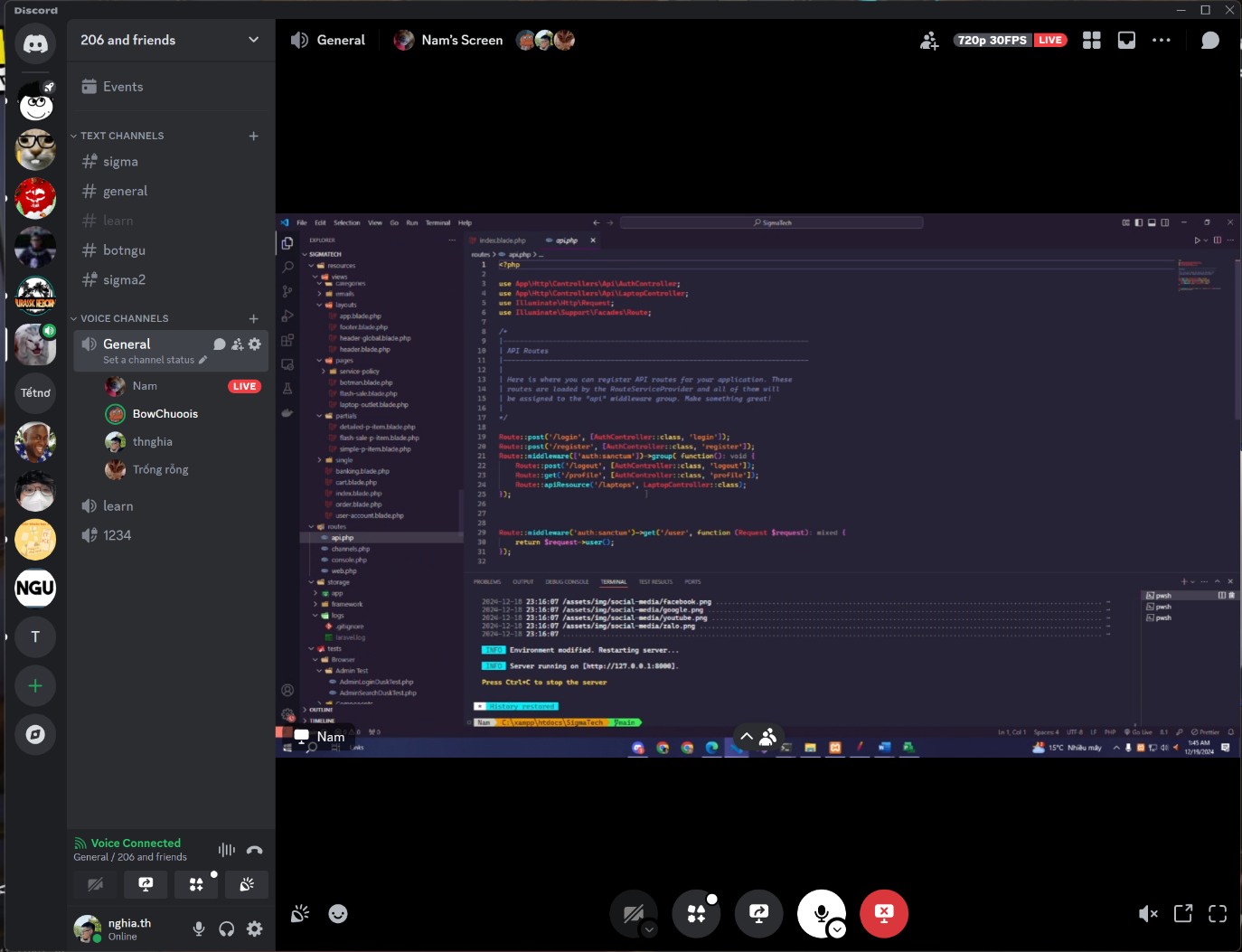
### 2.2.2. Áp dụng triết lý và tính kỷ luật của các thành viên để đảm bảo chất lượng phần mềm

Để đảm bảo chất lượng phần mềm trong quá trình phát triển **Sigmatech**, nhóm chúng em luôn đặt ra một triết lý phát triển rõ ràng và thúc đẩy tính kỷ luật cao trong công việc của từng thành viên. Mục tiêu của nhóm chúng em là không chỉ tạo ra sản phẩm chất lượng mà còn xây dựng một quy trình làm việc hiệu quả và linh hoạt, nơi mọi thành viên đều chủ động, sáng tạo và chịu trách nhiệm đối với công việc của mình.

**Triết lý phát triển :** nhóm chúng em dựa trên nguyên lý **Agile** và phương pháp **Scrum**, nhưng quan trọng hơn cả là việc nhóm duy trì kỷ luật và tinh thần hợp tác trong toàn đội. Mỗi thành viên đều đóng góp vào quá trình tạo ra **Sigmatech**, đảm bảo sản phẩm đáp ứng tốt nhất nhu cầu của khách hàng và người dùng cuối.

**Quản lý dự án với Jira:** Để giúp nhóm chúng em quản lý tiến độ và theo dõi các công việc liên quan đến **Sigmatech**, nhóm sử dụng **Jira** như công cụ chính. Mỗi công việc, từ việc triển khai tính năng đến việc sửa lỗi, đều được phân bổ rõ ràng trên Jira. Việc này giúp nhóm luôn cập nhật và kiểm soát được tình trạng của từng nhiệm vụ, từ đó tăng cường sự minh bạch và hiệu quả trong công việc. Nhóm luôn ghi lại các vấn đề phát sinh, từ đó xử lý kịp thời để đảm bảo tiến độ và chất lượng.

****

**Họp theo tuần để giải quyết vấn đề:** Nhóm tổ chức **cuộc họp hàng tuần bằng Discord** để thảo luận về các vấn đề phát sinh và tìm kiếm giải pháp cho những khó khăn mà từng thành viên gặp phải trong quá trình phát triển **Sigmatech**. Đây là cơ hội để nhóm trao đổi ý tưởng, cập nhật tiến độ công việc và giải quyết những vướng mắc. Những cuộc họp này giúp nhóm duy trì sự phối hợp chặt chẽ, đảm bảo rằng mọi vấn đề đều được giải quyết một cách nhanh chóng.

**Quản lý và log lỗi (Bug):** Trong suốt quá trình phát triển **Sigmatech**, khi phát hiện các lỗi (bug), nhóm sẽ **log ngay lập tức trên Discord** để theo dõi và xử lý. Việc ghi nhận lỗi một cách kịp thời giúp nhóm dễ dàng phân công công việc sửa lỗi cho từng thành viên, đồng thời đảm bảo rằng sản phẩm của nhóm luôn ổn định và không gặp phải các vấn đề nghiêm trọng khi triển khai.

**Đánh giá hoàn thành công việc:** Sau mỗi sprint và khi một công việc hoàn thành, nhóm sẽ **đánh dấu hoàn thành** trên Jira. Điều này giúp nhóm nhìn nhận lại những gì đã làm được và tiến độ mà nhóm đã đạt được. Các công việc được hoàn thành sẽ được kiểm tra lại kỹ lưỡng để đảm bảo tính chính xác và đáp ứng đầy đủ các yêu cầu, đồng thời giúp sản phẩm ngày càng hoàn thiện.

## 2.3. Đo lường chất lượng sản phẩm

Nhóm chúng em đặt ra những tiêu chí rõ ràng và đo lường chất lượng một cách cẩn thận trong quá trình phát triển website **Sigmatech** để đảm bảo rằng sản phẩm không chỉ hoàn thiện về mặt chức năng mà còn tối ưu về hiệu suất và trải nghiệm người dùng.

### 2.3.1. Xác định các tiêu chí đánh giá chất lượng khách quan và chủ quan

**Độ chính xác và tính đầy đủ của chức năng:** Kiểm tra xem các tính năng của website có hoạt động đúng theo yêu cầu ban đầu hay không, tham chiếu chặt chẽ đến yêu cầu của khách hàng cho sản phẩm của họ, thông qua quy trình phân tích nghiệp vụ.

**Hiệu suất hệ thống:** Đo lường tốc độ tải trang, thời gian phản hồi của website và khả năng xử lý lượng người dùng lớn trong cùng một thời điểm.

**Tính ổn định:** Đảm bảo rằng website hoạt động ổn định trong mọi trường hợp, bao gồm cả các tình huống tải cao hoặc các tương tác phức tạp từ người dùng.

**Trải nghiệm người dùng (UX):** Đặt bản thân mình vào người dùng, vừa phát triển vừa triển khai các chức năng đã hoàn thành và tự trải nghiệm đánh giá mức độ hài lòng.

**Tính bảo mật:**Kiểm tra khả năng bảo vệ thông tin người dùng và phòng chống các cuộc tấn công từ bên ngoài.

### 2.3.2. Các phương pháp đo lường chất lượng

**Đo lường chất lượng trong quá trình phát triển**

* Kiểm thử đơn vị (Unit Test)
* Kiểm thử tính năng (Feature Test)
* Kiểm thử tích hợp (Integration Test)
* Kiểm thử tự động/ Kiểm thử giao diện (Automated Testing/ UI Testing)
* Kiểm thử hiệu năng (Performance Testing)
* Kiểm thử bảo mật (Security Testing**)**

**Phản hồi từ người dùng:**Thu thập ý kiến và phản hồi từ người dùng cuối để đánh giá mức độ hài lòng và cải tiến sản phẩm.

## 2.4. Chi phí chất lượng phần mềm

# CHƯƠNG 3: MÔ TẢ QUÁ TRÌNH THỰC HIỆN VÀ CÁC HOẠT ĐỘNG TRONG VIỆC TRIỂN KHAI KIỂM THỬ VÀ ĐÁNH GIÁ CHẤT LƯỢNG PHẦN MỀM

**3.1. Công cụ sử dụng**

Để thực hiện kiểm thử và đảm bảo chất lượng phần mềm, tôi đã sử dụng các công cụ và thư viện sau:

**3.1.1. PHPUnit**

* PHPUnit là framework kiểm thử phổ biến cho PHP, được sử dụng để viết và chạy các unit test.
* Các tính năng nổi bật:
  + Cho phép so sánh kết quả thực tế với kết quả mong đợi thông qua các phương thức như assertEquals, assertTrue, v.v.
  + Hỗ trợ kiểm tra từng thành phần nhỏ của phần mềm, đảm bảo logic hoạt động đúng như mong muốn.
* **Ứng dụng trong dự án**:
  + Viết các unit test để kiểm tra logic xử lý dữ liệu trong các model (như CartItem, OrderDetail, Laptop, v.v.).
  + Đảm bảo rằng các mô hình hoạt động đúng và tương thích với cơ sở dữ liệu.

**3.1.2. Laravel Testing Features**

* Laravel tích hợp sẵn các tính năng hỗ trợ kiểm thử mạnh mẽ, giúp tăng tốc độ phát triển và kiểm tra phần mềm.
* **Các đặc điểm nổi bật**:
  + **Phân loại kiểm thử**:
    - Unit Test: Kiểm tra logic riêng lẻ trong từng class hoặc method.
    - Feature Test: Kiểm tra các chức năng lớn, bao gồm cả logic xử lý và tương tác với cơ sở dữ liệu.
  + **Hỗ trợ cơ sở dữ liệu**:
    - Laravel cung cấp trait RefreshDatabase, đảm bảo cơ sở dữ liệu được làm mới trước mỗi bài kiểm tra, giúp tránh xung đột dữ liệu.
  + **Hàm kiểm tra hữu ích**:
    - assertDatabaseHas: Kiểm tra dữ liệu có tồn tại trong bảng.
    - assertInstanceOf: Kiểm tra một đối tượng có đúng kiểu mong muốn.
    - assertEquals: So sánh giá trị thực tế với giá trị mong đợi.
* **Ứng dụng trong dự án**:
  + Kiểm tra mối quan hệ giữa các mô hình (belongsTo, hasMany, belongsToMany) như Laptop và Attribute, OrderDetail và Order.
  + Kiểm tra tính chính xác của dữ liệu trong cơ sở dữ liệu khi thực hiện các thao tác CRUD.

**3.1.3. Laravel Dusk**

* Laravel Dusk là công cụ kiểm thử tự động giao diện người dùng (UI Testing) được tích hợp trong Laravel.
* **Các tính năng chính**:
  + Cho phép mô phỏng hành vi của người dùng trên trình duyệt, như click, nhập liệu, chuyển trang, v.v.
  + Hỗ trợ tạo các bài kiểm tra giao diện một cách trực quan với API dễ sử dụng.
  + Có thể sử dụng với các trình duyệt không giao diện (headless browser) như ChromeDriver.
* **Ứng dụng trong dự án**:
  + Kiểm tra các luồng giao diện chính của ứng dụng (như thêm sản phẩm vào giỏ hàng, đặt hàng).
* Đảm bảo các chức năng giao diện hoạt động ổn định, không xảy ra lỗi khi tương tác thực tế.

## 3.2. Unit Testing

Unit test là một giai đoạn quan trọng trong kiểm thử phần mềm, tập trung kiểm tra từng đơn vị nhỏ nhất của mã nguồn, thường là các hàm hoặc phương thức. Các đơn vị này được kiểm tra độc lập để đảm bảo rằng chúng hoạt động đúng theo yêu cầu đã định nghĩa. Trong quá trình kiểm thử đơn vị, các ca kiểm thử được xây dựng để xác minh đầu ra của mỗi đơn vị dựa trên các đầu vào khác nhau, giúp phát hiện lỗi sớm trước khi tích hợp các module lại với nhau. Unit test không chỉ đảm bảo tính chính xác của từng phần nhỏ mà còn giúp giảm thiểu chi phí sửa lỗi trong giai đoạn sau của quy trình phát triển.

**3.2.1. Nhân sự**

|  |  |
| --- | --- |
| **Thành viên** | **Module đảm nhận kiểm thử** |
| Nguyễn Thế Nghĩa | Model User  Model Laptop  Model Chi tiết đơn hàng  Model Đơn hàng  Model Giỏ hàng |

**3.2.2. Phạm vi kiểm thử**

|  |  |
| --- | --- |
| **Module kiểm thử** | **Phạm vi** |
| Model User | **Các thuộc tính (Attributes):**   * Kiểm tra các thuộc tính chính xác và được ánh xạ đúng với cơ sở dữ liệu:   + id, name, email, password, role, created\_at, updated\_at.   **Các ràng buộc (Validation Rules):**   * name không được trống và có độ dài tối đa (ví dụ: 255 ký tự). * email:   + Bắt buộc (not null).   + Định dạng đúng (email hợp lệ).   + Không trùng lặp (unique). * password:   + Bắt buộc.   + Có độ dài tối thiểu (ví dụ: 8 ký tự). * role:   + Mặc định là **user** hoặc **admin**.   + Giá trị nằm trong tập hợp hợp lệ (['user', 'admin']).   **Quan hệ (Relationships):**   * User có thể:   + Có nhiều Orders (Đơn hàng).   + Có nhiều Carts (Giỏ hàng).   **Phương thức (Methods):**   * **Kiểm tra mật khẩu:**   + checkPassword(): Đảm bảo phương thức so sánh mật khẩu nhập vào và mật khẩu mã hóa chính xác. * **Kiểm tra vai trò người dùng:**   + isAdmin(): Trả về true nếu vai trò là **admin**, ngược lại là false. |
| Model Laptop | **Các thuộc tính (Attributes):**   * Kiểm tra các thuộc tính được ánh xạ chính xác:   + id, name, price, brand, description, quantity, image, created\_at, updated\_at.   **Các ràng buộc (Validation Rules):**   * name: Bắt buộc, không trùng lặp, độ dài tối đa. * price:   + Bắt buộc.   + Giá trị phải là số dương. * brand: Bắt buộc. * quantity:   + Không âm.   + Giá trị mặc định là 0 nếu không được cung cấp. * description: Có thể rỗng (nullable). * image: Đảm bảo lưu đường dẫn hợp lệ.   **Quan hệ (Relationships):**   * Laptop có thể:   + Thuộc về nhiều **Orders** thông qua bảng chi tiết đơn hàng (Order Details).   + Được thêm vào nhiều **Carts**.   **Phương thức (Methods):**   * **Tính năng giảm giá:**   + getDiscountPrice(): Trả về giá sau khi áp dụng khuyến mãi nếu có. * **Kiểm tra tồn kho:**   + isAvailable(): Trả về true nếu số lượng lớn hơn 0, ngược lại là false. |
| Model Chi tiết đơn hàng | **Các thuộc tính (Attributes):**   * id, order\_id, laptop\_id, quantity, price, created\_at, updated\_at.   **Các ràng buộc (Validation Rules):**   * order\_id: Bắt buộc, khóa ngoại liên kết với bảng **Orders**. * laptop\_id: Bắt buộc, khóa ngoại liên kết với bảng **Laptops**. * quantity:   + Bắt buộc.   + Là số nguyên dương. * price:   + Bắt buộc.   + Là số dương.   **Quan hệ (Relationships):**   * Thuộc về một **Order**. * Thuộc về một **Laptop**.   **Phương thức (Methods):**   * **Tính tổng giá trị chi tiết đơn hàng:**   + getTotalPrice(): Trả về giá trị bằng quantity \* price. |
| Model Đơn hàng | **Các thuộc tính (Attributes):**   * id, user\_id, status, total\_price, created\_at, updated\_at.   **Các ràng buộc (Validation Rules):**   * user\_id: Bắt buộc, khóa ngoại liên kết với bảng **Users**. * status:   + Bắt buộc.   + Giá trị nằm trong các trạng thái hợp lệ (ví dụ: pending, processing, completed, cancelled). * total\_price:   + Bắt buộc.   + Là số dương.   **Quan hệ (Relationships):**   * Thuộc về một **User**. * Có nhiều **Order Details**.   **Phương thức (Methods):**   * **Cập nhật trạng thái:**   + updateStatus(newStatus): Đảm bảo trạng thái được cập nhật hợp lệ. * **Tính tổng giá trị đơn hàng:**   + calculateTotalPrice(): Tính toán tổng giá trị từ các chi tiết đơn hàng. |
| Model Giỏ hàng | **Các thuộc tính (Attributes):**   * id, user\_id, laptop\_id, quantity, created\_at, updated\_at.   **Các ràng buộc (Validation Rules):**   * user\_id: Bắt buộc, khóa ngoại liên kết với bảng **Users**. * laptop\_id: Bắt buộc, khóa ngoại liên kết với bảng **Laptops**. * quantity:   + Bắt buộc.   + Là số nguyên dương.   **Quan hệ (Relationships):**   * Thuộc về một **User**. * Thuộc về một **Laptop**.   **Phương thức (Methods):**   * **Kiểm tra tồn kho trước khi thêm vào giỏ hàng:**   + canAddToCart(laptop, quantity): Trả về true nếu số lượng yêu cầu không vượt quá số lượng còn trong kho. * **Tính tổng giá trị giỏ hàng:**   + getTotalCartValue(): Tính tổng giá trị của tất cả sản phẩm trong giỏ hàng. |

## 3.3. Integration Testing

Integration Test là một giai đoạn trong kiểm thử phần mềm. Mỗi module phần mềm riêng biệt được kết hợp lại và kiểm thử theo nhóm. Kiểm thử tích hợp nhận các module đầu vào đã được kiểm thử đơn vị, nhóm chúng vào các tập hợp lớn hơn, áp dụng các ca kiểm thử đã được định nghĩa trong kế hoạch kiểm thử tích hợp vào tập hợp đó, và cung cấp đầu ra cho hệ thống tích hợp.

Feature Test là một loại kiểm thử được Laravel cung cấp, nhằm kiểm tra hành vi của các tính năng cụ thể trong ứng dụng từ góc nhìn của người dùng. Mục tiêu chính của Feature Test là đảm bảo rằng một tính năng cụ thể hoạt động đúng từ đầu đến cuối.

**3****.3.1. Nhân sự**

|  |  |
| --- | --- |
| **Thành viên** | **Module đảm nhận kiểm thử** |
| Đinh Thành Nam | [ Client ] Đăng ký/ Đăng nhập/ Xác nhận/ Lấy lại mật khẩu  [ Client ] Trang chủ và trang Flash Sale  [ Admin ] Đăng nhập  [ Admin ] Quản lý khuyến mãi / Flash Sale  [ Admin ] Tìm kiếm |
| Nguyễn Thế Nghĩa | [ Client ] Trang chi tiết sản phẩm  [ Client ] Trang chuyên mục sản phẩm  [ Client ] Tìm kiếm & bộ lọc  [ Admin ] Dashboard báo cáo  [ Admin ] Quản lý slider (carousel) |
| Nguyễn Duy Hưng | [ Client ] Thêm giỏ hàng & Quản lý giỏ hàng  [ Client ] Thanh toán / Các hình thức thanh toán  [ Client ] Quản lý tài khoản người dùng  [ Client ] Quản lý đơn hàng cá nhân & Lịch sử mua hàng  [ Admin ] Quản lý sản phẩm  [ Admin ] Quản lý đơn hàng  [ Admin ] Quản lý khuyến mãi / Flash Sale |

**3.3.2. Phạm vi kiểm thử**

|  |  |
| --- | --- |
| **Module kiểm thử** | **Phạm vi** |
| [ Client ] Đăng ký/ Đăng nhập/ Xác nhận/ Lấy lại mật khẩu | **Integration test:**  Test người dùng đăng ký thành công  Test đăng ký yêu cần mọi trường dữ liệu input  Test đăng ký yêu cầu email hợp lệ  Test đăng nhập hợp lệ  Test đăng nhập không hợp lệ  Test đăng nhập với người dùng bình thường  Test đăng nhập khi không nhập mật khẩu hoặc email  Test người dùng đăng nhập được hướng đến trang chủ  **Feature test:**  Test form đăng ký nên được hiển thị  Test form đăng nhập nên được hiển thị  Test form yêu cầu mật khẩi nên được hiển thị  Test người dùng chưa xác nhận không thể xem trang tài khoản cá nhân  Test người dùng không phải admin không thể xem admin dashboard  Test người dùng đã xác thực có thể xem trang tài khoản cá nhân |
| [ Admin ] Đăng nhập | **Integration test:**  Test đăng nhập với người dùng là admin  **Feature test:**  Test người dùng admin có thể xem admin dashboard |
| [ Client ] Trang chủ và trang Flash Sale | **Integration test:**  Test trang chủ có đường dẫn đến chính xác danh mục  Kiểm tra flash sale item có thời gian còn lại đúng  Kiểm tra flash sale item có % giảm giá đúng  **Feature test:**  Test trang flash sale hiển thị  Test trang flash sale có sự thay đổi Test trang chủ hiển thị  Test trang chủ có sự thay đổi  Test trang chủ có slides |
| [ Admin ] Quản lý khuyến mãi / Flash Sale | **Integration test:**  Kiểm tra thêm sản phẩm sale thành công |
| [ Client ] Trang chi tiết sản phẩm | **Integration test:**  Test hiển thị đúng chi tiết sản phẩm  Test trả về 404 nếu sản phẩm không tồn tại  Test trả về 404 nếu thuộc tính sản phẩm không hợp lệ |
| [ Client ] Trang chuyên mục sản phẩm | **Integration test:**  Test hiển thị chuyên mục laptop gaming thành công |
| [ Client ] Tìm kiếm & bộ lọc | **Integration test:**  Test hiển thị đúng chi tiết sản phẩm được tìm kiếm  Test laptop on top nên được hiển thị  Test phân trang laptop có 13 bản ghi  **Feature test:**  Test hiển thị đúng bộ lọc sản phẩm  Test hiển thị 404 nếu sản phẩm trong bộ lọc không tồn tại  Test laptop có 27 bản ghi |
| [ Client ] Thêm giỏ hàng & Quản lý giỏ hàng | **Integration test:**  Test hiển thị giỏ hàng trống sản phẩm  Test hiển thị giỏ hàng có sản phẩm  Test thêm sản phẩm vào giỏ hàng  Test thêm sản phẩm tương tự vào giỏ hàng và tăng số lượng sản phẩm  Test xóa sản phẩm khỏi giỏ hàng  Test mua ngay sản phẩm  Test mua ngay sản phẩm và cập nhật số lượng |
| [ Client ] Thanh toán / Các hình thức thanh toán | **Integration test:**  Test thông tin đặt hàng hiển thị đúng sản phẩm  Test đặt hàng thành công  Test đặt hàng với phương thức chuyển khoản ngân hàng thành công |
| [ Client ] Quản lý tài khoản người dùng  [ Client ] Quản lý đơn hàng cá nhân & Lịch sử mua hàng | **Integration test:**  Test người dùng có thể xem đơn hàng của họ  Test trả về 404 nếu đơn hàng không tồn tại  Test hiển thị chi tiết đơn hàng nếu đơn hàng tồn tại  Test người dùng có thể sửa thông tin tài khoản của họ |
| [ Admin ] Quản lý sản phẩm | **Integration test:**  Test hiển thị tất cả sản phẩm  Test thêm sản phẩm mới thành công  Test sửa sản phẩm thành công  Test xóa sản phẩm thành công  Test tìm kiểm sản phẩm thành công  Test tìm kiếm sản phẩm không có kết quả |
| [ Admin ] Quản lý đơn hàng | **Integration test:**  Test hiển thị tất cả đơn hàng  Test sửa thông tin đơn hàng thành công |

## 3.4. System Testing

## 3.5. Acceptance Testing

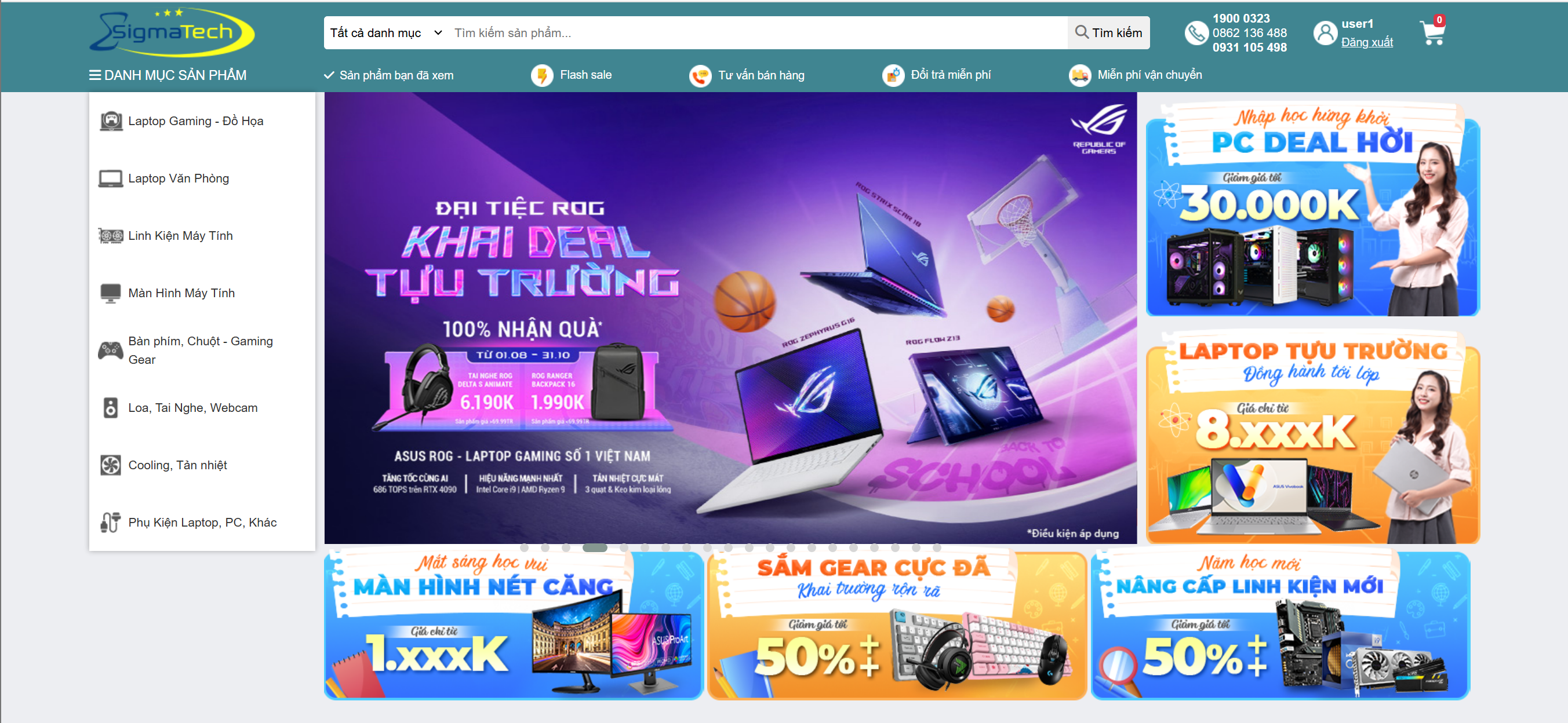
# CHƯƠNG 4: ĐÁNH GIÁ CHẤT LƯỢNG PHẦN MỀM

Để đảm bảo chất lượng tổng thể của phần mềm, chương này tập trung vào đánh giá các khía cạnh quan trọng trong hệ thống. Từ đó, nhóm xác định được mức độ hoàn thiện của sản phẩm và khả năng đáp ứng yêu cầu sử dụng thực tế của người dùng.

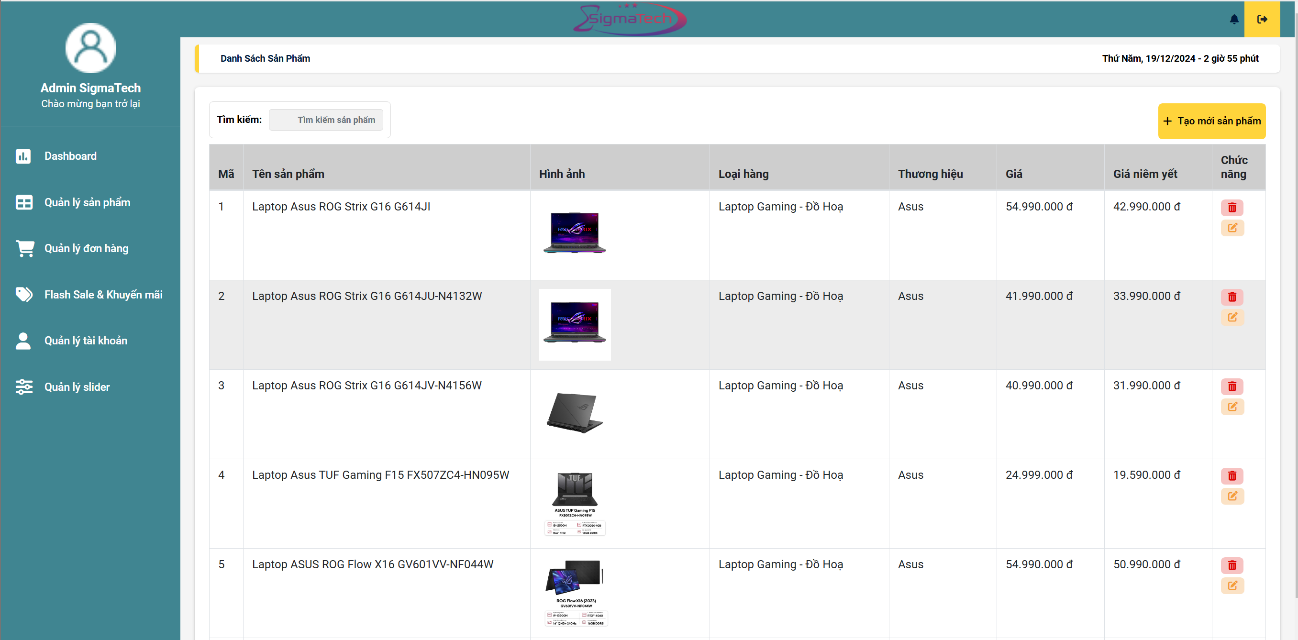
**4.1. Tính đúng**

Tính đúng của hệ thống là yếu tố đảm bảo rằng phần mềm hoạt động chính xác theo yêu cầu được đề ra. Nhóm thực hiện các kiểm thử sau:

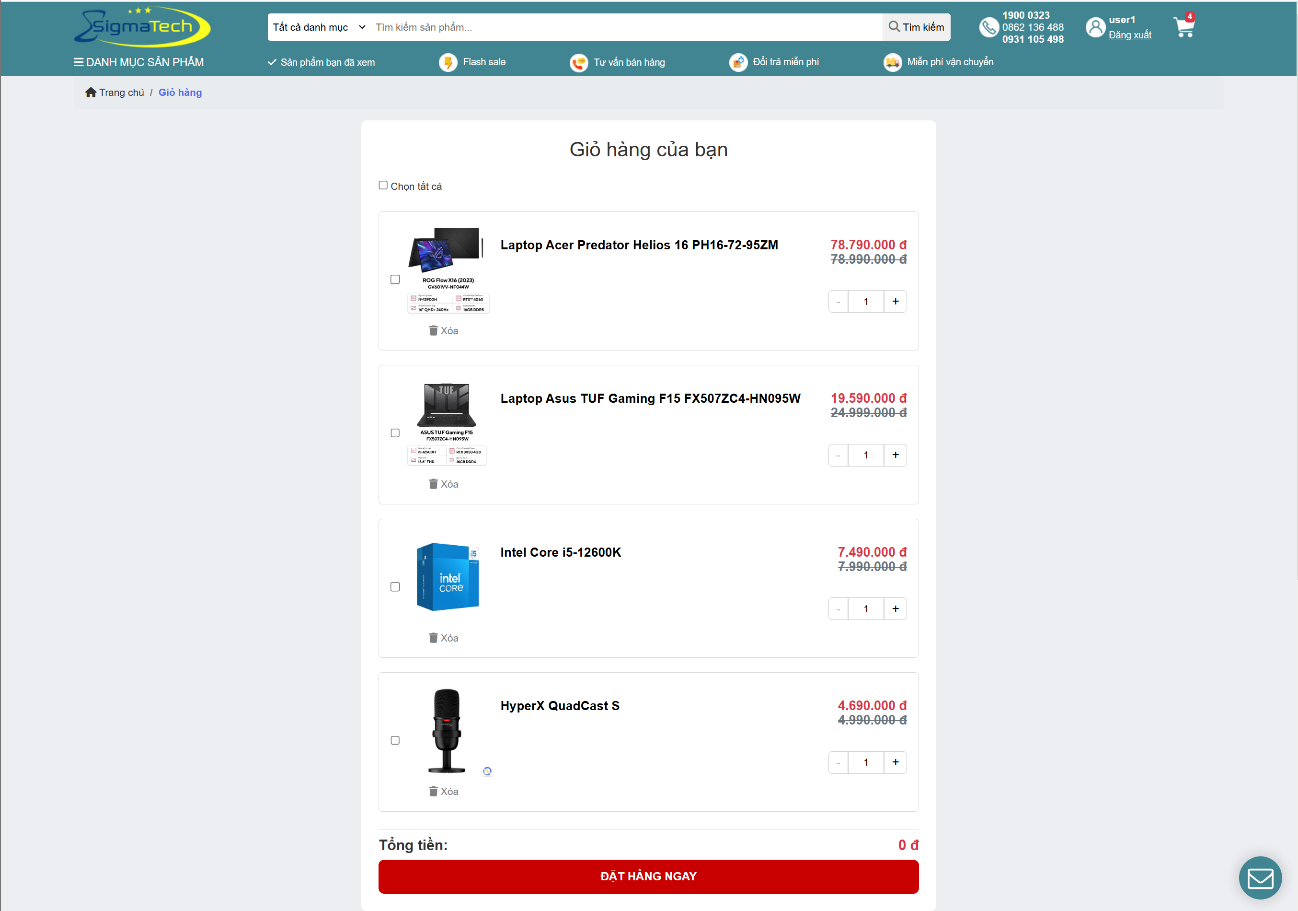
* **Kiểm thử đăng ký tài khoản:**
  + Kiểm tra khả năng xử lý các trường hợp:
    1. Thông tin nhập hợp lệ: Tạo tài khoản thành công.
    2. Thông tin không đầy đủ (thiếu trường thông tin bắt buộc như email, mật khẩu).
    3. Email không hợp lệ.
  + Kết quả: Hệ thống phản hồi chính xác với thông báo lỗi rõ ràng hoặc xác nhận đăng ký thành công.
* **Kiểm thử đăng nhập:**
  + Kiểm tra khả năng xử lý các trường hợp:
    1. Thông tin đăng nhập chính xác.
    2. Thông tin không đầy đủ (thiếu tên đăng nhập hoặc mật khẩu).
    3. Thông tin không hợp lệ (tên người dùng không tồn tại hoặc mật khẩu sai).
  + Đặc biệt kiểm tra khả năng phân quyền:
    1. **Người dùng thông thường:** Được điều hướng đến giao diện người dùng cơ bản.



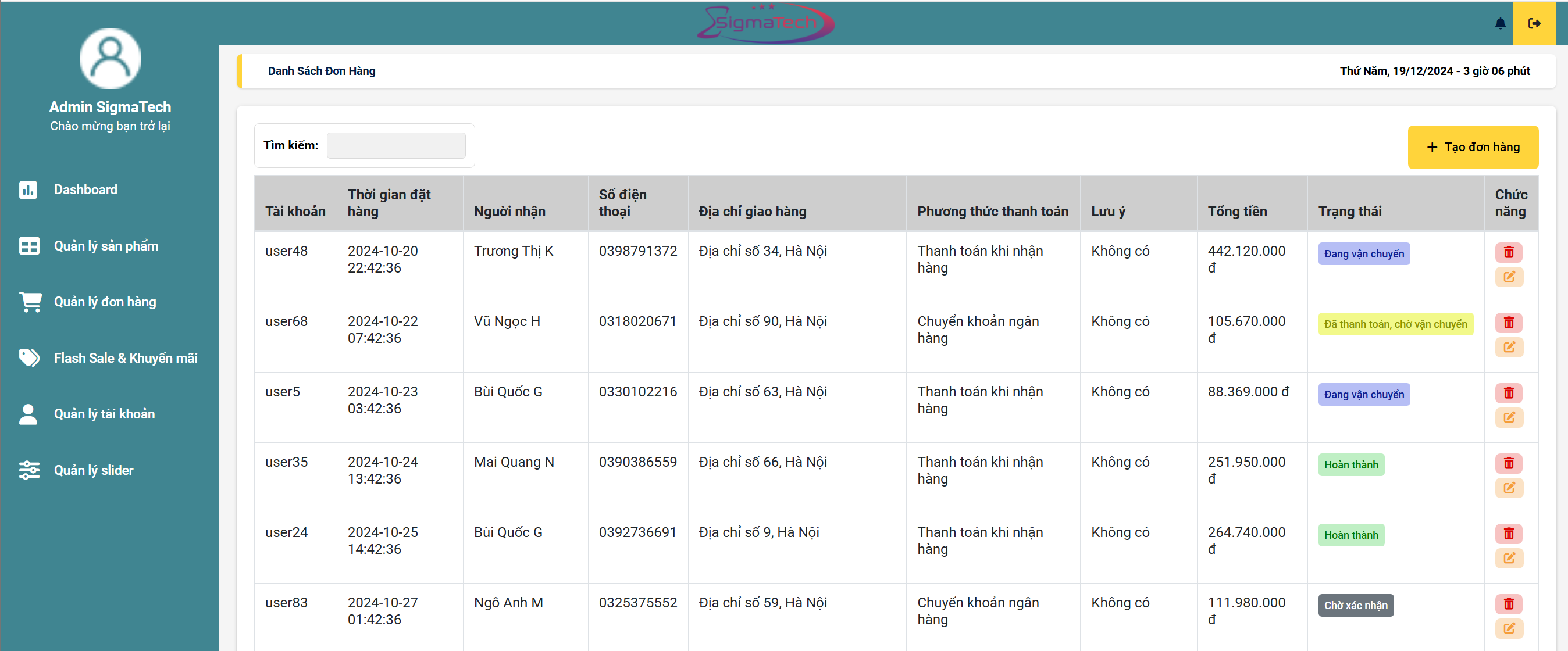
* + 1. **Quản trị viên:** Được điều hướng đến giao diện quản trị.
  + Kết quả: Hệ thống xử lý đúng các trường hợp và hiển thị thông báo phù hợp.



* **Kiểm thử giỏ hàng:**
  + Thực hiện các kiểm thử liên quan đến chức năng giỏ hàng, bao gồm:
    1. Thêm sản phẩm vào giỏ.
    2. Hiển thị chính xác thông tin sản phẩm trong giỏ (tên, giá, số lượng).
    3. Cập nhật số lượng sản phẩm.
    4. Xóa sản phẩm khỏi giỏ.
    5. Chức năng "mua ngay": Đảm bảo hệ thống cập nhật số lượng sản phẩm và chuyển sang bước thanh toán.
  + Kết quả: Các thao tác trên giỏ hàng được thực hiện chính xác, dữ liệu cập nhật đầy đủ.



* **Kiểm thử đơn hàng:**
  + Kiểm tra các luồng đặt hàng:
    1. Đặt hàng bằng phương thức COD (thanh toán khi nhận hàng).
    2. Đặt hàng qua thanh toán trực tuyến.
  + Đảm bảo thông tin đơn hàng (tên khách hàng, sản phẩm, địa chỉ giao hàng) được lưu trữ chính xác trong cơ sở dữ liệu.
  + Kết quả: Hệ thống ghi nhận đầy đủ thông tin đơn hàng và điều hướng đúng đến trang kết quả.



**4.2. Tính kiểm thử được**

Tính kiểm thử được của hệ thống phản ánh khả năng dễ dàng kiểm tra và phát hiện lỗi trong các luồng xử lý.

* **Dữ liệu mô phỏng:**
  + Sử dụng các factory của Laravel để tạo dữ liệu giả lập (người dùng, sản phẩm, giỏ hàng, đơn hàng). Điều này giúp nhóm kiểm thử từng trường hợp cụ thể mà không cần dữ liệu thực tế.
* **Cơ chế kiểm tra kết quả:**
  + Sử dụng các lệnh **assert** trong PHPUnit để kiểm tra:
    1. Trạng thái HTTP trả về từ server.
    2. Nội dung phản hồi.
    3. Dữ liệu trong cơ sở dữ liệu.
    4. Các session được tạo trong quá trình xử lý.
* **Phân tách logic:**
  + Các chức năng như giỏ hàng, đặt hàng, và xác thực người dùng được thiết kế độc lập. Điều này giúp từng module có thể kiểm thử riêng biệt, tăng hiệu quả phát hiện lỗi.

**4.3. Tính an toàn**

Tính an toàn tập trung vào bảo vệ dữ liệu người dùng và ngăn chặn các hành vi không mong muốn.

* **Kiểm tra đăng nhập:**
  + Đảm bảo tài khoản không bị truy cập trái phép khi thông tin đăng nhập không hợp lệ.
  + Người dùng không đăng nhập không được phép truy cập các thông tin nhạy cảm.
* **Kiểm thử đầu vào:**
  + Kiểm tra chặt chẽ các trường nhập liệu như:
    1. Email và mật khẩu.
    2. Thông tin đặt hàng.
  + Loại bỏ các rủi ro tấn công như SQL Injection và XSS (Cross-site Scripting).
* **Phân quyền:**
  + Xác minh rằng chỉ tài khoản quản trị viên mới có thể truy cập các chức năng quản trị, và tài khoản người dùng thường chỉ được sử dụng các chức năng cơ bản.

**4.4. Tính toàn vẹn**

Tính toàn vẹn đảm bảo rằng dữ liệu trong hệ thống không bị mất mát hoặc sai lệch trong quá trình sử dụng.

* **Dữ liệu giỏ hàng và đơn hàng:**
  + Đảm bảo các thao tác như thêm, cập nhật, và xóa sản phẩm không gây ra lỗi mất dữ liệu.
* **Luồng giao dịch:**
  + Đảm bảo tính chính xác của các thông tin quan trọng như:
    1. Tổng giá trị đơn hàng.
    2. Phương thức thanh toán.
    3. Thông tin người nhận hàng.

**4.5. Tính tiêu chuẩn và tính chuẩn**

Hệ thống tuân thủ các tiêu chuẩn phát triển và kiểm thử phần mềm:

* **Framework Laravel:**
  + Áp dụng các tiêu chuẩn MVC của Laravel trong tổ chức mã nguồn.
  + Sử dụng factory để tạo dữ liệu giả lập cho kiểm thử.
  + Kết hợp PHPUnit và Laravel Dusk để tự động hóa kiểm thử.
* **RESTful API:**
  + Các endpoint được thiết kế theo chuẩn RESTful, đảm bảo tính thống nhất trong giao tiếp giữa client và server.

**4.6. Tính độc lập**

Tính độc lập đảm bảo rằng các bài kiểm thử không phụ thuộc lẫn nhau, giúp tăng tính chính xác và đáng tin cậy của quá trình kiểm thử.

* **Reset cơ sở dữ liệu:**
  + Sử dụng cơ chế **RefreshDatabase** để khôi phục trạng thái ban đầu cho từng bài kiểm thử.
* **Dữ liệu giả lập:**
  + Các thông tin như tài khoản người dùng và sản phẩm được tạo giả lập để tránh phụ thuộc vào dữ liệu thực tế.

**4.7. Các tiêu chuẩn phần mềm và tính ứng dụng**

* **Tiêu chuẩn phần mềm:**
  + Hệ thống sử dụng các công cụ kiểm thử như PHPUnit và Laravel Dusk, kết hợp với các quy trình kiểm thử hiện đại để đảm bảo tính chuyên nghiệp và hiệu quả.
* **Tính ứng dụng:**
  + Các chức năng chính (đăng ký, đăng nhập, xử lý giỏ hàng, đặt hàng) được kiểm tra đầy đủ, đảm bảo người dùng có thể thao tác dễ dàng và hiệu quả trong môi trường thực tế.

# KẾT LUẬN

Hệ thống báo cáo và dashboard mà nhóm xây dựng đã đạt được nhiều kết quả tích cực, cụ thể:

* **Hiệu quả hỗ trợ quản lý:** Các báo cáo trực quan và dễ hiểu, hỗ trợ các BI Users như Content Team, Finance Team, và Original Programming Team đưa ra các quyết định chiến lược nhanh chóng và hiệu quả hơn.
* **Ý nghĩa đối với doanh nghiệp:** Hệ thống giúp Netflix hiểu rõ hơn về thói quen người dùng, tối ưu danh mục nội dung, và cải thiện ROI từ các khoản đầu tư sản xuất nội dung. Điều này không chỉ nâng cao hiệu quả hoạt động mà còn cải thiện năng lực cạnh tranh.
* **Bài học thu được:** Nhóm đã nắm bắt quy trình ETL, cách xây dựng kho dữ liệu và tầm quan trọng của việc lựa chọn nền tảng BI phù hợp như Power BI.

**Khuyến nghị**

1. **Nâng cao chất lượng dữ liệu:**
   * Áp dụng các biện pháp làm sạch dữ liệu tự động để giảm lỗi.
   * Xây dựng cơ chế kiểm tra và đồng bộ dữ liệu từ các nguồn đầu vào khác nhau.
2. **Khai thác giá trị dữ liệu hiệu quả hơn:**
   * Kết hợp AI và machine learning để dự đoán xu hướng thị trường và tối ưu thuật toán gợi ý nội dung.
   * Sử dụng các báo cáo chi tiết hơn để theo dõi hiệu quả của từng nội dung tại từng khu vực.
3. **Tăng cường an toàn và bảo mật:**
   * Sử dụng mã hóa dữ liệu và quản lý quyền truy cập chặt chẽ.
   * Đảm bảo tuân thủ các quy định bảo mật quốc tế, như GDPR.
   * Cập nhật thường xuyên các biện pháp phòng chống rủi ro an ninh mạng.

# TÀI LIỆU THAM KHẢO

Marketing, P. (không ngày tháng). Được truy lục từ https://pamarketing.vn/mo-hinh-kinh-doanh-cua-netflix-p1/

*Netflix 10+ Year Stock Data [2002-2020]*. (2024). Được truy lục từ Kaggle: https://www.kaggle.com/datasets/aayushmishra1512/netflix-stock-data

*Netflix data with IMDB scores*. (2024). Được truy lục từ Kaggle: https://www.kaggle.com/datasets/sarahjeeeze/imdbfile