Chủ đề: Website Coffee Shop

Thành viên nhóm:

|  |  |  |
| --- | --- | --- |
| Thành viên | Phần trăm đóng góp | Phân công công việc |
| Hứa Khánh Huy | 100% | Database + Admin page |
| Bùi Chí Cường | 100% | Staff page + Thiết kế Slide |
| Nguyên Tiến Đạt | 100% | Viết document + Database |
| Nguyên Phước Lộc | 100% | Home page + Viết document |
| Trương Minh Khải | 100% | Viết document + Thiết kế Slide |

**I. Định nghĩa vấn đề:**

Coffee Shop là một ngành kinh doanh phát triển mạnh mẽ trong thời gian gần đây. Tuy nhiên, để thành công trong lĩnh vực này, các cửa hàng cà phê phải đối mặt với nhiều thách thức. Vấn đề chính của ngành coffee shop bao gồm:

**Cạnh tranh khốc liệt**: Sự cạnh tranh ngày càng gia tăng trong ngành coffee shop đòi hỏi các cửa hàng phải tạo ra những sản phẩm và trải nghiệm khách hàng độc đáo để thu hút và duy trì khách hàng.

**Quản lý nhân viên**: Coffee shop cần duy trì một đội ngũ nhân viên chất lượng và hiệu quả, đồng thời đảm bảo họ luôn duy trì mức độ phục vụ tốt để giữ chân khách hàng.

**Tiếp cận thị trường**: Việc tiếp cận thị trường cũng đóng vai trò quan trọng để mở rộng khách hàng và tăng doanh số bán hàng. Chúng tôi nhận thấy rằng nhiều quán coffee vẫn chưa tận dụng hết tiềm năng của mình để tiếp cận khác hàng và thông tin về sản phẩm, dịch vụ của họ chưa đến được với người tiêu dùng

**Xây dựng và thúc đẩy thương hiệu**: Xây dựng và thúc đẩy thương hiệu không chỉ giúp tạo ra sự nhận biết mạnh mẽ mà còn đóng vai trò quan trọng trong việc xây dựng một cơ sở khách hàng trung thành và đáng tin cậy cho coffee shop là chìa khóa để thành công trong ngành dịch vụ coffee shop.

**II. Giải pháp:**

Để giải quyết các vấn đề nêu trên và đạt được sự thành công trong ngành coffee shop, cần áp dụng các giải pháp sau đây:

**Nghiên cứu và phân tích thị trường**: Hiểu rõ đối tượng khách hàng, xu hướng thị trường và đối thủ cạnh tranh giúp coffee shop điều chỉnh chiến lược kinh doanh và tạo ra sản phẩm và dịch vụ phù hợp.

**Tối ưu hóa quá trình quản lý tồn kho**: Sử dụng phần mềm quản lý tồn kho để theo dõi và điều chỉnh tồn kho một cách hiệu quả, giảm thiểu lãng phí và tối ưu hóa lợi nhuận.

**Đào tạo và phát triển nhân viên**: Đầu tư vào đào tạo và phát triển nhân viên, đảm bảo họ có kiến thức và kỹ năng cần thiết để cung cấp dịch vụ tốt cho khách hàng.

**Quản lý thương hiệu**: Xây dựng thương hiệu thông qua một chiến dịch tiếp thị sáng tạo, kết hợp với việc tăng cường sự hiện diện thương hiệu, duy trì vị thế của cửa hàng trong lòng khách hàng.

**Tiếp thị và khả năng tiếp cận qua phương tiện xã hội**: Việc có một trang web giúp giúp khách hàng dễ dàng truy cập và tiếp cận menu cũng như quảng cáo các thức ăn, thức uống, không gian shop đến khách hàng truy cập.

**III. Waterfall Model**

Mô hình phát triển phần mềm **Waterfall Model** (mô hình thác nước) là một trong những mô hình phát triển phần mềm cổ điển, đặc biệt hiệu quả trong những dự án mà yêu cầu độ tin cậy và quy trình cố định. Dưới đây là sự giới thiệu của mô hình **Waterfall Model** và lý do vì sao nó có thể được áp dụng trong ngành coffee shop.

**Mô hình Waterfall Model:**

Mô hình Waterfall Model chia quá trình phát triển phần mềm thành các giai đoạn tuần tự, mỗi giai đoạn phụ thuộc vào giai đoạn trước đó và không có sự chuyển tiếp ngược lại.

Lý do sử dụng mô hình **Waterfall Model** trong coffee shop:

Mô hình **Waterfall Model** có thể áp dụng trong ngành coffee shop với các lý do sau:

**Độ tin cậy**: Mô hình Waterfall Model đảm bảo rằng từng giai đoạn được hoàn thành một cách cẩn thận trước khi tiến hành giai đoạn tiếp theo, giúp đảm bảo tính đáng tin cậy của phần mềm.

**Yêu cầu cố định**: Trong ngành coffee shop, quản lý nhân viên, quản lý sản phẩm, và quản lý hóa đơn thường ít thay đổi, làm cho mô hình Waterfall phù hợp hơn.

**Tính tuần tự**: Việc áp dụng mô hình Waterfall Model giúp chủ cửa hàng cà phê tập trung vào từng pha một mà không bị xao lạc bởi việc thay đổi trong quá trình phát triển.

**IV. Use Case và Sơ đồ Thiết kế Cơ sở Dữ liệu**

**IV. A. Use Case:**

**i. Admin**

Quản lý danh sách sản phẩm (đồ uống, thực phẩm).

Quản lý danh sách nhân viên (thêm, sửa, xoá).

**ii. Nhân viên**

Ghi đơn hàng cho khách hàng.

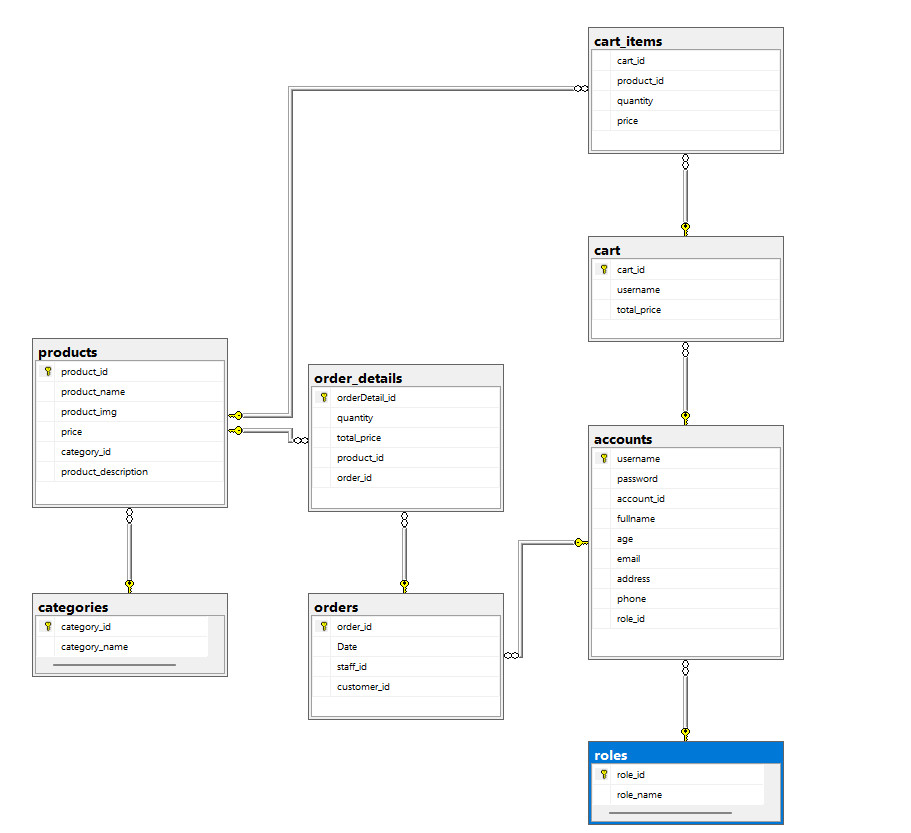
Tạo tài khoản thành viên.

**iii. Khách hàng**

Xem menu sản phẩm.

Xem lịch sử đơn hàng.

**IV. B. Sơ đồ Thiết kế Cơ sở Dữ liệu**



**V. Phương pháp quản lý phiên bản:**

Phương pháp quản lý sự thay đổi của tài liệu và mã nguồn là rất quan trọng trong quá trình phát triển và duy trì một dự án.

**V. A. Quản lý tài liệu:**

**Tạo Tài Liệu Chia Sẻ**: Sử dụng Google Drive để tạo tài liệu và chia sẻ nó với thành viên trong dự án. Điều này giúp mọi người cùng làm việc trên tài liệu cùng một lúc và theo dõi các thay đổi.

**Kiểm Tra Lịch Sử Thay Đổi**: Google Drive cung cấp tính năng theo dõi lịch sử thay đổi của tài liệu. Bạn có thể xem ai đã thực hiện thay đổi, khi nào, và điều gì đã thay đổi.

**Bản Sao Lưu Tài Liệu**: Định kỳ tạo bản sao lưu của tài liệu để đảm bảo rằng bạn có thể khôi phục lại phiên bản trước đó nếu cần thiết.

**Xử lý Phản Hồi và Phê Duyệt**: Sử dụng tính năng ô bình luận để cho phép các thành viên trong nhóm gửi phản hồi và đề xuất sửa đổi trực tiếp trên tài liệu.

**V. B. Quản lý mã nguồn:**

Nhóm em đã chọn và sử dụng Git cụ thể hơn là dịch vụ GitHub như một là hệ thống kiểm soát các phiên bản phân tán mã nguồn mở

Kết hợp **GitHub** với Hệ Thống Quản Lý Phiên Bản (VCS): GitHub là một nền tảng dựa trên Git, một hệ thống quản lý phiên bản phổ biến. Sử dụng VCS giúp theo dõi sự thay đổi của mã nguồn.

**Tạo Repository (Kho Lưu Trữ)**: Tạo một kho lưu trữ trên GitHub để lưu trữ mã nguồn. Mỗi dự án thường có một repository riêng.

**Commit và Push Thay Đổi**: Khi bạn hoàn thành một phần của mã nguồn hoặc thực hiện sự thay đổi, bạn commit (lưu trạng thái hiện tại) và push (đẩy) chúng lên repository.

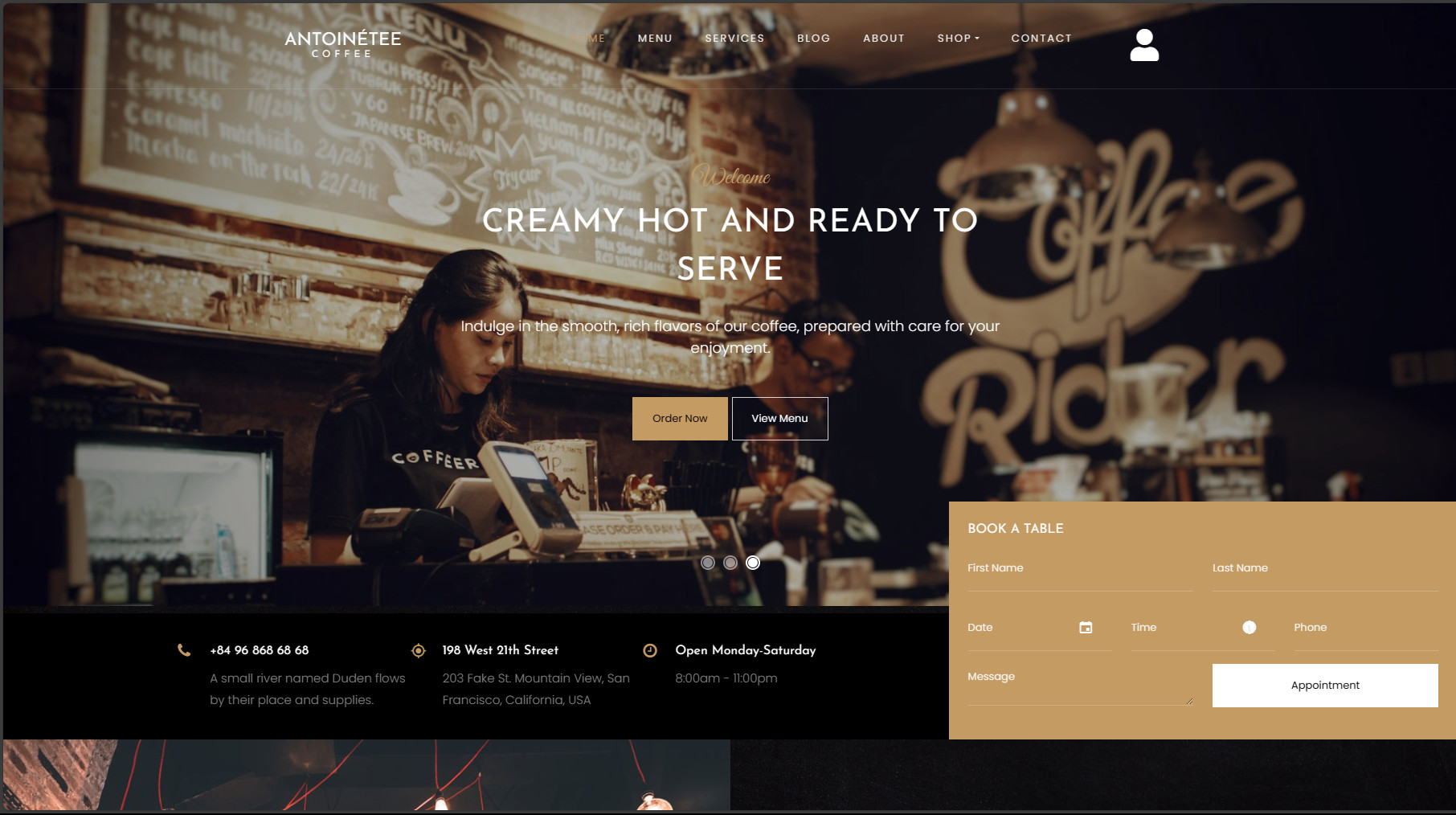
**Pull Request (Yêu Cầu Gộp)**: Đối với các thay đổi lớn hoặc tính năng mới, bạn tạo một pull request để cho phép các thành viên khác xem xét và kiểm tra sự thay đổi trước khi được gộp vào mã nguồn chính.

**Xử Lý Xung Đột**: Nếu có xung đột khi nhiều người cùng thay đổi cùng một phần của mã nguồn, GitHub cung cấp các công cụ giúp giải quyết xung đột.

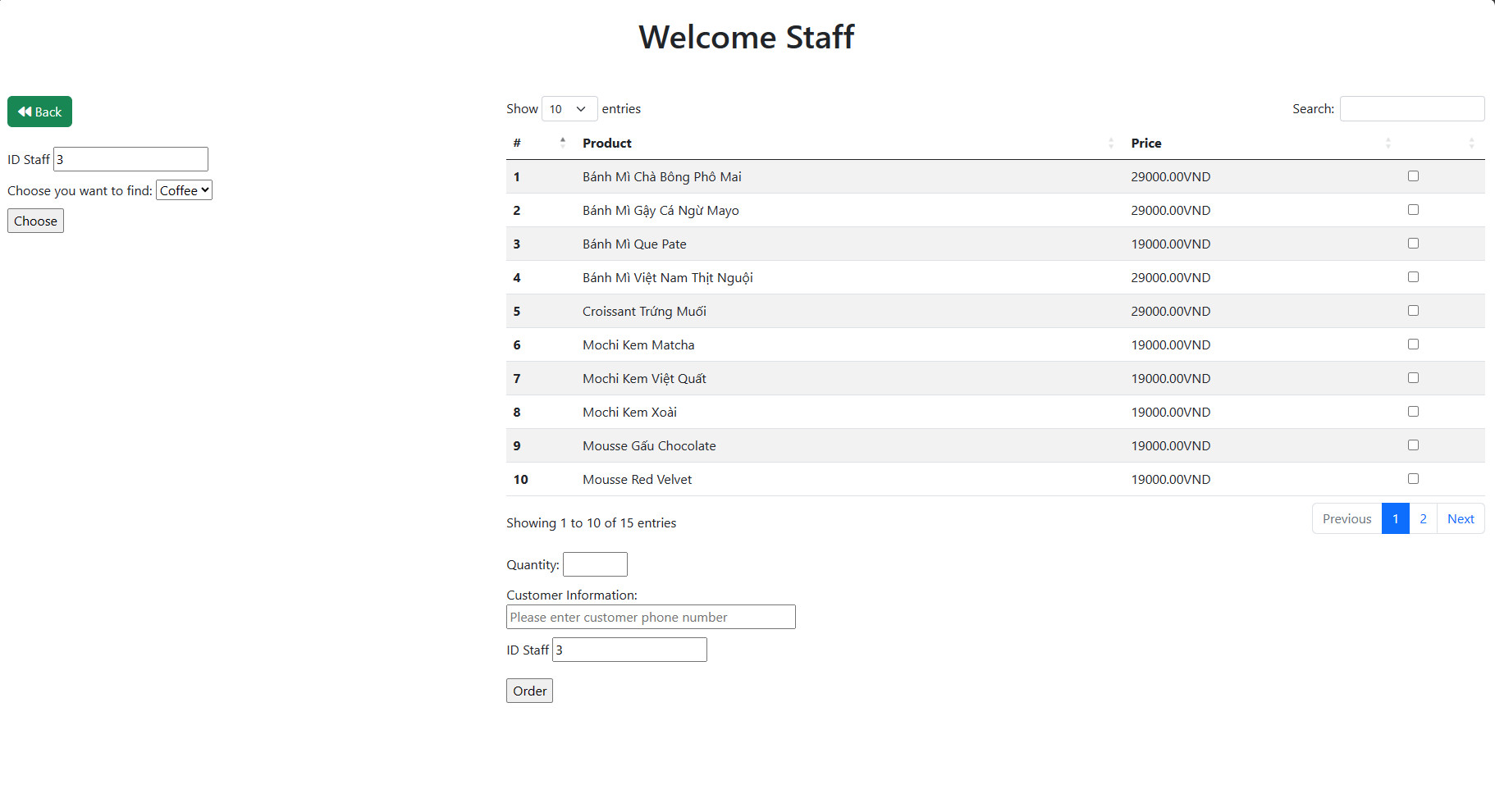
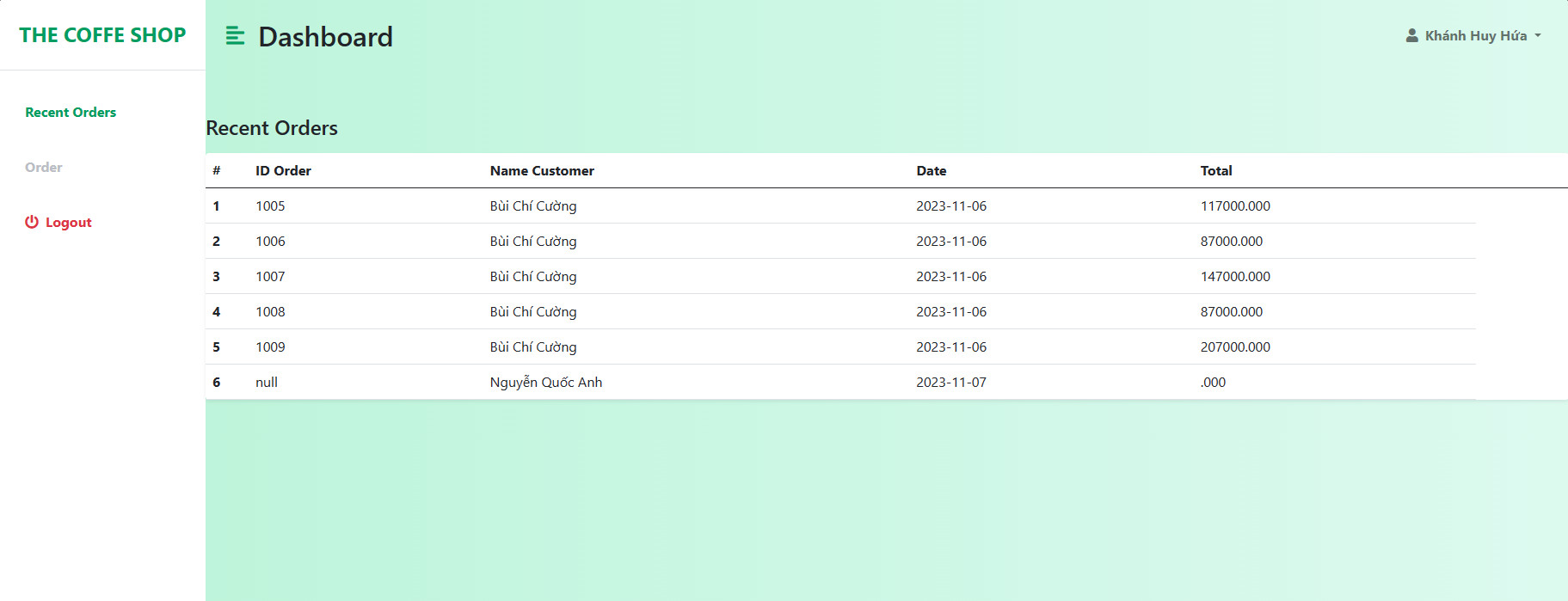
**Lịch Sử Thay Đổi Chi Tiết**: GitHub theo dõi từng thay đổi trong lịch sử mã nguồn, giúp xác định ai đã thực hiện thay đổi gì và khi nào.

**VI. Giao diện ứng dụng:**

**I. Home page**



**II. Staff page**



**III. Admin page**

