DOCUMENT PRJ

**Subject:** PRJ301-Java Web Application Development

**Class:** SE1707

**Group:** 5

**Final Project Topic:** Website Coffee Shop

**Group members and contribution rates:**

|  |  |  |
| --- | --- | --- |
| **Group Members** | **Contribution** | **Assigned Work** |
| Hứa Khánh Huy | 100% | Admin + Home + Menu |
| Nguyễn Phước Lộc | 100% | Staff page + Login + About |
| Bùi Chí Cường | 100% | Viết document + Database + Blog |
| Nguyễn Tiến Đạt | 100% | Viết document + Database + Navbar |
| Trương Minh Khải | 100% | Viết document + Thiết kế Slide + Navbar |

1. **Problem Definition && Solution:**
   1. **Problem Definition:**

**Thực Trạng Của Quán Cafe Truyền Thống:**

Trong ngữ cảnh của ngày nay, quán cafe truyền thống thường gặp phải các thách thức lớn trong việc quản lý dữ liệu khách hàng, sản phẩm, nhân viên và quản lý đơn đặt hàng tại cửa hàng. Các phương pháp quản lý truyền thống thường bao gồm việc sử dụng bản ghi giấy và các hệ thống tạm thời, dẫn đến việc dễ xảy ra lỗi, mất mát dữ liệu, và gây ra sự bất tiện cho cả khách hàng và nhân viên.

**Sự Khác Biệt Giữa Quán Cafe Truyền Thống và Quán Cafe Sử Dụng Trang Web để Quản Lý:**

Trong khi đó, các quán cafe có trang web quản lý thường trải qua một quy trình quản lý hiệu quả hơn. Họ có khả năng:

Quản Lý Dữ Liệu Khách Hàng và Sản Phẩm Một Cách Chính Xác: Dữ liệu của khách hàng và thông tin sản phẩm được lưu trữ một cách chính xác và an toàn trong hệ thống, giúp tránh mất mát thông tin và đảm bảo tính xác thực của dữ liệu.

Tăng Sự Thuận Tiện Cho Khách Hàng: Khách hàng có thể xem menu, đặt hàng và thậm chí thanh toán trực tuyến, giúp tăng cường trải nghiệm mua sắm và giảm thời gian chờ đợi.

Quản Lý Nhân Viên và Đơn Đặt Hàng Tại Cửa Hàng: Nhân viên có thể sử dụng hệ thống để quản lý các đơn đặt hàng tại cửa hàng, giúp họ xử lý đơn hàng một cách chính xác và hiệu quả hơn.

* 1. **Solution:**

**Sử Dụng Trang Web để Quản Lý Tất Cả Các Khía Cạnh Của Quán Cafe:**

Chúng tôi đã phát triển một trang web đặc biệt cho quán cafe của bạn, tập trung vào việc giải quyết các vấn đề mà quán cafe truyền thống thường gặp phải. Dưới đây là cách trang web của chúng tôi giúp đỡ quán cafe của bạn

**Quản Lý Dữ Liệu Khách Hàng và Sản Phẩm:**

Trang web cho phép quán cafe lưu trữ thông tin chi tiết về khách hàng, bao gồm lịch sử đơn hàng và các thông tin về khách hàng.

Danh sách sản phẩm được hiển thị một cách rõ ràng, bao gồm mô tả, hình ảnh và giá cả, giúp quản lý dễ dàng cập nhật thông tin và giữ cho menu luôn được cập nhật.

**Quản Lý Nhân Viên và Đơn Đặt Hàng Tại Cửa Hàng:** Nhân viên có thể sử dụng trang web để xem danh sách các đơn đặt hàng và thông tin liên quan đến khách hàng.

**Hỗ Trợ Tương Tác và Phản Hồi**

Bằng cách sử dụng trang web của chúng tôi, quán cafe của bạn sẽ không chỉ giải quyết các vấn đề hàng ngày một cách hiệu quả mà còn tạo ra một trải nghiệm tốt hơn cho cả nhân viên và khách hàng, giúp nâng cao chất lượng dịch vụ và tăng cường sự hài lòng từ phía khách hàng.

1. **Software Development Model:**

**Waterfall Model:**

1. **Use case && Database:**
2. **Management:**
3. **Application Interface:**
4. **Final Project:**

**II. 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.

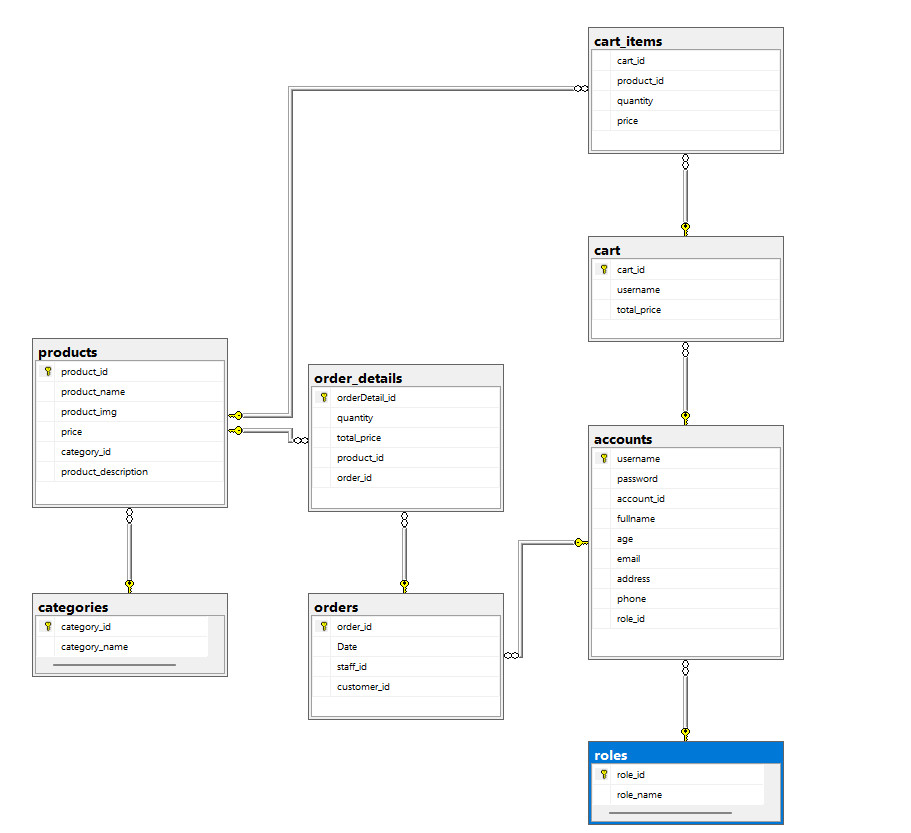
Tạo tài khoản thành viên.

**iii. Khách hàng**

Xem menu sản phẩm.

Xem lịch sử đơn hàng.

**IV. B. CSơ 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.

**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.

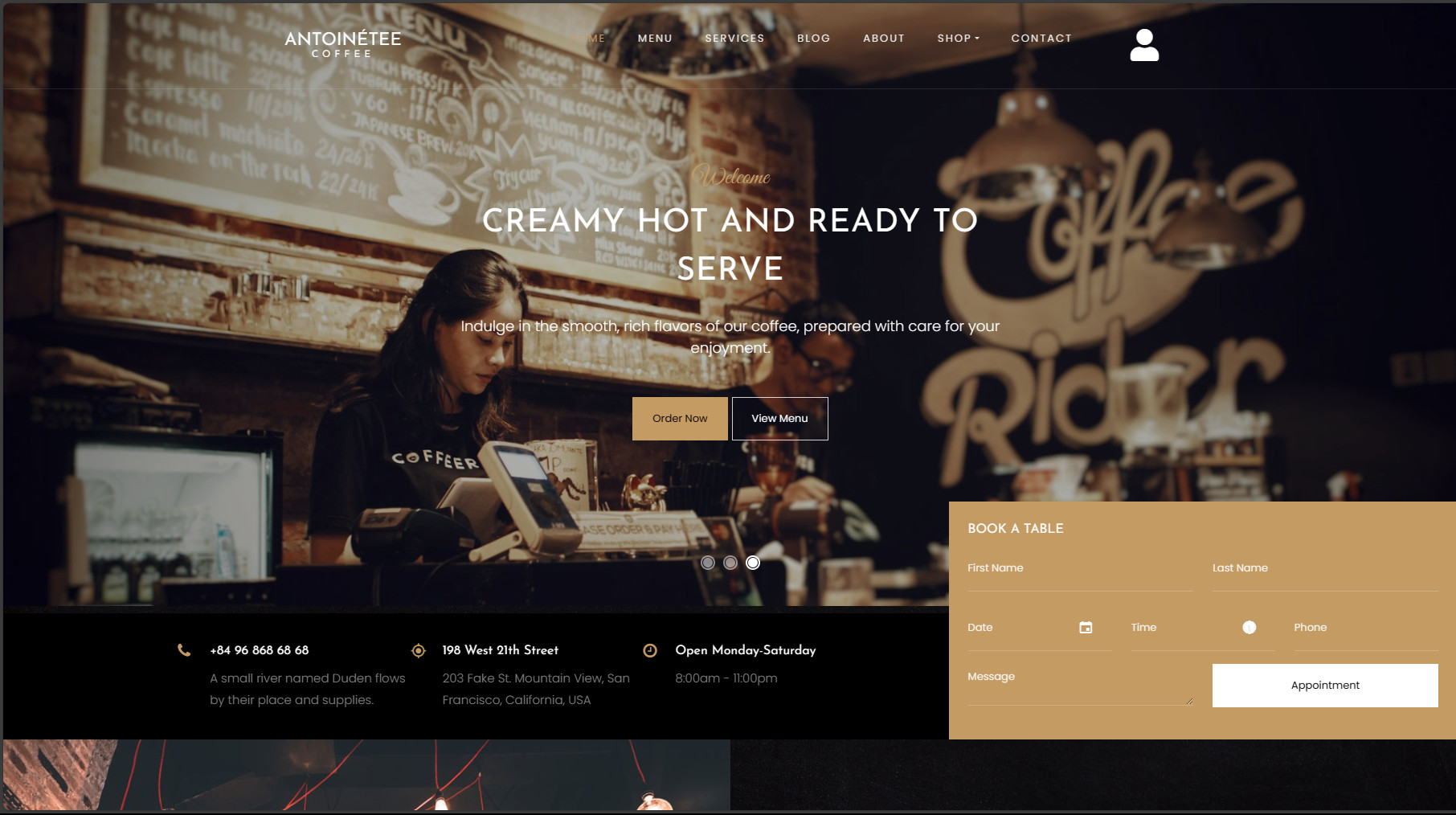
**Pull Request (Yêu Cầu Gộp)**: Đối với các thay đổi lớn hoặc tính năng mới, bạn tạo một pull request để cho phép các thành viên khác xem xét và kiểm tra sự thay đổi trước khi được gộp vào mã nguồn chính.

**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.

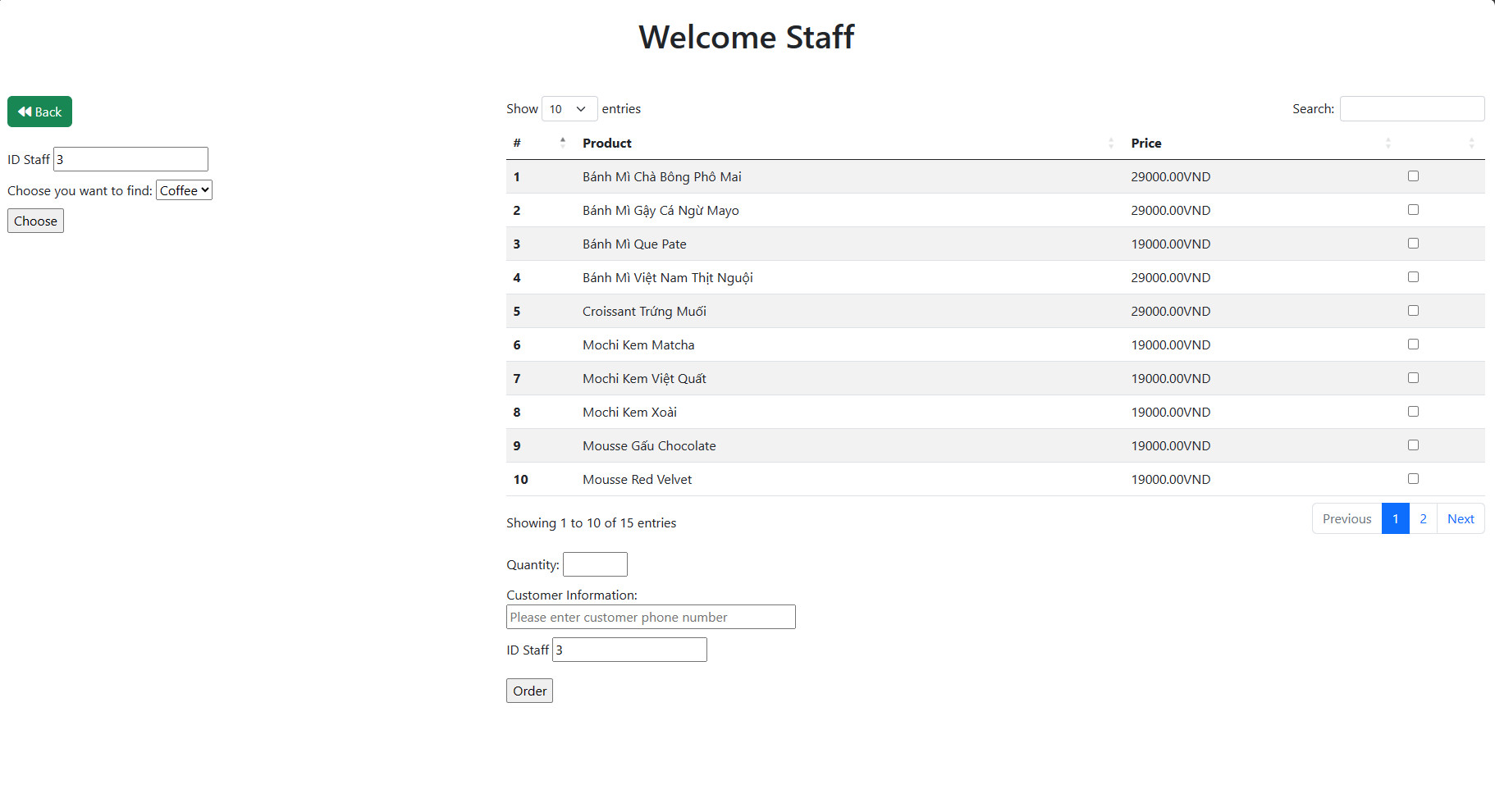
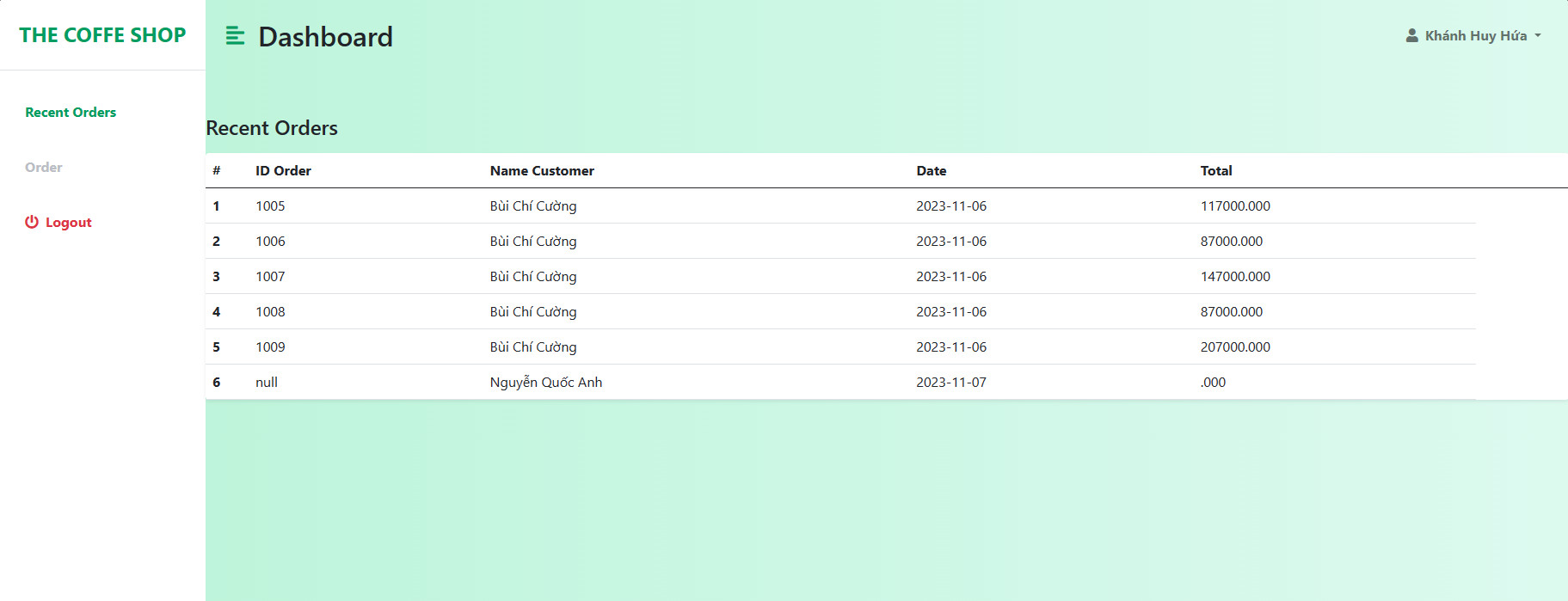
**Lịch Sử Thay Đổi Chi Tiết**: GitHub theo dõi từng thay đổi trong lịch sử mã nguồn, giúp xác định ai đã thực hiện thay đổi gì và khi nào.

**VI. Giao diện ứng dụng:**

**I. Home page**



**II. Staff page**



**III. Admin page**

