**TRƯỜNG ĐẠI HỌC CÔNG NGHIỆP TP HỒ CHÍ MINH**

**KHOA CÔNG NGHỆ THÔNG TIN**

****

**ĐỒ ÁN CUỐI KÌ**

**PHÁT TRIỂN ỨNG DỤNG**

**ĐỀ TÀI: PORTFOLIO MANAGEMENT SYSTEM**

*Người thực hiện*: **NGUYỄN CHÍ TRUNG – 22719231**

**ĐÀO TIẾN SANG – 22705971**

**HOA XUÂN HOÀN - 22689381**

**NGUYỄN TRƯỜNG VƯƠNG - 22642961**

Lớp **: 420300350103**

Khoá  **: 18**

*Người hướng dẫn*: **TS TRƯƠNG VĨNH LINH**

**KS TRẦN TẤN THÀNH**

**THÀNH PHỐ HỒ CHÍ MINH, NĂM 2025**

**TRƯỜNG ĐẠI HỌC CÔNG NGHIỆP TP HỒ CHÍ MINH**

**KHOA CÔNG NGHỆ THÔNG TIN**

****

**ĐỒ ÁN CUỐI KÌ**

**PHÁT TRIỂN ỨNG DỤNG**

**ĐỀ TÀI: PORTFOLIO MANAGEMENT SYSTEM**

Người thực hiện: **NGUYỄN CHÍ TRUNG - 22719231**

**ĐÀO TIẾN SANG - 22705971**

**HOA XUÂN HOÀN - 22689381**

**NGUYỄN TRƯỜNG VƯƠNG - 22642961**

Lớp **: 420300350103**

Khoá  **: 18**

*Người hướng dẫn*: **TS TRƯƠNG VĨNH LINH**

**KS TRẦN TẤN THÀNH**

**THÀNH PHỐ HỒ CHÍ MINH, NĂM 2025**

**LỜI CẢM ƠN**

Trước tiên, em xin bày tỏ lòng biết ơn chân thành đến Thầy ThS. Trương Vĩnh Linh – KS. Trần Tấn Thành người đã tận tình hướng dẫn, hỗ trợ và định hướng chuyên môn cho em trong suốt quá trình thực hiện bài luận. Sự tận tâm giảng dạy, những góp ý quý báu và sự đồng hành của Thầy đã giúp em hoàn thành đề tài một cách nghiêm túc và hiệu quả hơn.

Em cũng xin gửi lời cảm ơn sâu sắc đến quý Thầy, Cô thuộc Khoa Công nghệ Thông tin – Trường Đại học Công nghiệp TP. Hồ Chí Minh, những người đã luôn tận tâm giảng dạy và truyền đạt cho em nền tảng kiến thức vững chắc trong suốt quá trình học tập. Những bài học và kinh nghiệm quý báu từ quý Thầy, Cô là hành trang quan trọng để em hoàn thiện bài luận này cũng như phát triển trong tương lai.

Em xin chân thành cảm ơn!

**ĐỒ ÁN ĐƯỢC HOÀN THÀNH**

**TẠI TRƯỜNG ĐẠI HỌC CÔNG NGHIỆP TP HỒ CHÍ MINH**

Nhóm xin cam đoan đây là sản phẩm đồ án của riêng chúng em và được sự hướng dẫn của ThS. Trương Vĩnh Linh - KS. Trần Tấn Thành. Các nội dung nghiên cứu, kết quả trong đề tài này là trung thực và chưa công bố dưới bất kỳ hình thức nào trước đây. Những số liệu trong các bảng biểu phục vụ cho việc phân tích, nhận xét, đánh giá được chính chúng em thu thập từ các nguồn khác nhau có ghi rõ trong phần tài liệu tham khảo.

Ngoài ra, trong đồ án còn sử dụng một số nhận xét, đánh giá cũng như số liệu của các tác giả khác, cơ quan tổ chức khác đều có trích dẫn và chú thích nguồn gốc.

**Nếu phát hiện có bất kỳ sự gian lận nào chúng em xin hoàn toàn chịu trách nhiệm về nội dung đồ án của mình.** Trường đại học Công nghiệp TP Hồ Chí Minh không liên quan đến những vi phạm tác quyền, bản quyền do chúng em gây ra trong quá trình thực hiện (nếu có).

*TP. Hồ Chí Minh, ngày tháng năm 2025*

*Tác giả*

*Đào Tiến Sang*

*Nguyễn Chí Trung*

*Hòa Xuân Hoàn*

*Nguyễn Trường Vương*

**PHẦN ĐÁNH GIÁ CỦA GIẢNG VIÊN**

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Tp. Hồ Chí Minh, ngày tháng năm

(kí và ghi họ tên)

**TÓM TẮT**

**MỤC LỤC**

1. **Giới Thiệu Chung**1.1. Bối cảnh và nhu cầu phát triển hệ thống  
   1.2. Mục tiêu của dự án  
   1.3. Phạm vi chức năng  
   1.4. Tầm quan trọng của dự án
2. **Phân Tích Yêu Cầu**2.1. Yêu cầu chức năng  
   2.1.1. Quản lý người dùng  
   2.1.2. Quản lý ví điện tử  
   2.1.3. Quản lý tài khoản ngân hàng liên kết  
   2.1.4. Quản lý danh mục đầu tư  
   2.1.5. Giao dịch tài sản  
   2.1.6. Theo dõi thị trường chứng khoán  
   2.1.7. Tương tác với trợ lý AI  
   2.2. Yêu cầu phi chức năng  
   2.2.1. Giao diện và trải nghiệm người dùng  
   2.2.2. Hiệu năng hệ thống  
   2.2.3. Bảo mật thông tin  
   2.2.4. Khả năng mở rộng và bảo trì  
   2.2.5. Địa hóa cho thị trường Việt Nam
3. **Thiết Kế Hệ Thống**3.1. Kiến trúc hệ thống  
   3.1.1. Lớp trình diễn (Presentation Layer)  
   3.1.2. Lớp ứng dụng (Application Layer)  
   3.1.3. Lớp dữ liệu (Data Layer)  
   3.1.4. Tích hợp dịch vụ bên ngoài  
   3.2. Thiết kế cơ sở dữ liệu  
   3.2.1. Sơ đồ quan hệ thực thể  
   3.2.2. Mô tả chi tiết các bảng dữ liệu  
   3.3. Thiết kế giao diện người dùng (UI/UX)  
   3.3.1. Nguyên tắc thiết kế  
   3.3.2. Phân tích các giao diện chính
4. **Hiện Thực Hệ Thống**4.1. Môi trường và công cụ phát triển  
   4.2. Cấu trúc thư mục dự án  
   4.3. Ứng dụng Portfolio - Thành phần trung tâm  
   4.3.1. Models - Tầng dữ liệu  
   4.3.2. Views - Tầng xử lý logic  
   4.3.3. Templates và Static Files  
   4.4. Các module hỗ trợ chuyên biệt  
   4.4.1. VNStock Services Module  
   4.4.2. Utils Module  
   4.5. Hiện thực các luồng nghiệp vụ chính  
   4.5.1. Luồng giao dịch mua bán cổ phiếu  
   4.5.2. Luồng nạp tiền qua ngân hàng  
   4.5.3. Luồng cập nhật dữ liệu thị trường  
   4.5.4. Luồng tương tác AI Chatbot
5. **Kiểm Thử**5.1. Kế hoạch kiểm thử  
   5.2. Mô tả các kịch bản kiểm thử quan trọng  
   5.3. Kết quả kiểm thử và quy trình khắc phục lỗi
6. **Triển Khai và Hướng Dẫn Sử Dụng**6.1. Yêu cầu hệ thống và môi trường triển khai  
   6.2. Quy trình cài đặt chi tiết  
   6.3. Hướng dẫn sử dụng toàn diện các chức năng hệ thống
7. **Đánh Giá và Hướng Phát Triển**7.1. Đánh giá những điểm đã đạt được và ưu điểm của hệ thống  
   7.2. Phân tích những hạn chế còn tồn tại  
   7.3. Đề xuất các hướng phát triển và hoàn thiện hệ thống trong tương lai
8. **Kết Luận**8.1. Tóm tắt những kết quả chính của dự án  
   8.2. Khẳng định giá trị của hệ thống
9. **Tài Liệu Tham Khảo**

6

**DANH MỤC KÍ HIỆU VÀ CHỮ VIẾT TẮT**

**CÁC KÝ HIỆU VÀ CÁC CHỮ VIẾT TẮT**

*API Programming Interface (Giao diện lập trình ứng dụng)*

*Auth0 Dịch vụ xác thực và quản lý danh tính*

*CSS3 Cascading Style Sheets phiên bản 3*

*ERD Entity Relationship Diagram (Sơ đồ quan hệ thực thể)*

*HTML5 HyperText Markup Language phiên bản 5*

*MVC Model-View-Controller (Mô hình-View-Bộ điều khiển)*

*ORM Object-Relational Mapping (Ánh xạ quan hệ đối tượng)*

*UI/UX User Interface/User Experience*

*UAT User Acceptance Testing*

*VNSTOCK Thư viện thu thập dữ liệu thị trường chứng khoán Việt Nam*

*WSGI Web Server Gateway Interface*

*2FA Two-Factor Authentication*

**DANH MỤC CÁC HÌNH VẼ**

Hình 1. Sơ đồ quan hệ thực thể (ERD) của hệ thống quản lý danh mục đầu tư

Hình 2.1 Giao diện trang đăng nhập và đăng ký

Hình 2.2 Giao diện Dashboard tổng quan

Hình 2.3 Giao diện quản lý danh mục đầu tư

Hình 2.4 Giao diện giao dịch cổ phiếu

Hình 2.5 Giao diện theo dõi thị trường

Hình 2.6 Giao diện quản lý ví điện tử

**DANH MỤC CÁC BẢNG**

Bảng 1 Cấu trúc bảng User (portfolio.User)

Bảng 2 Cấu trúc bảng Wallet (portfolio.Wallet)

Bảng 3 Cấu trúc bảng BankAccount (portfolio.BankAccount)

Bảng 4 Cấu trúc bảng BankTransaction (portfolio.BankTransaction)

Bảng 5 Cấu trúc bảng Portfolio (portfolio.Portfolio)

Bảng 6 Cấu trúc bảng PortfolioSymbol (portfolio.PortfolioSymbol)

Bảng 7 Cấu trúc bảng StockTransaction (portfolio.StockTransaction)

Bảng 8 Cấu trúc bảng Assets (portfolio.Assets)

[1](#_rnwziriw7ufg)

**PORTFOLO MANAGEMENT SYSTEM**

1. **Giới Thiệu Chung**

Trong bối cảnh thị trường tài chính ngày càng phát triển mạnh mẽ và thu hút sự quan tâm rộng rãi của các nhà đầu tư cá nhân, nhu cầu về một công cụ hỗ trợ quản lý danh mục đầu tư một cách hiệu quả, minh bạch và thân thiện với người dùng đã trở nên vô cùng cấp thiết. Đặc biệt tại Việt Nam, với sự phát triển nhanh chóng của thị trường chứng khoán và xu hướng đầu tư cá nhân ngày càng gia tăng, việc có một hệ thống quản lý đầu tư chuyên nghiệp nhưng dễ tiếp cận trở thành yếu tố quyết định đến thành công của các nhà đầu tư.

Chính từ thực tiễn này, dự án "Hệ Thống Quản Lý Danh Mục Đầu Tư" đã được khởi xướng và triển khai bởi một đội ngũ phát triển tâm huyết, bao gồm các thành viên: Hoa Xuân Hoàn (22689381), Đào Tiến Sang (22705971), Nguyễn Chí Trung (22719231), và Nguyễn Trường Vương (22642961). Nhóm tác giả nhận thức được rằng, trong thời đại công nghệ 4.0, việc ứng dụng các giải pháp số hóa vào quản lý tài chính cá nhân không chỉ là xu thế tất yếu mà còn là chìa khóa để tối ưu hóa hiệu quả đầu tư.

Mục tiêu cốt lõi mà dự án hướng đến là xây dựng một nền tảng ứng dụng web toàn diện, cung cấp cho người dùng cá nhân một giải pháp công nghệ để theo dõi, quản lý các tài sản tài chính, thực hiện giao dịch mua bán cổ phiếu, và quản lý dòng tiền một cách thuận tiện và an toàn. Hơn nữa, hệ thống được kỳ vọng sẽ hỗ trợ nhà đầu tư trong việc đưa ra các quyết định đầu tư sáng suốt hơn thông qua việc cung cấp thông tin thị trường real-time và các công cụ phân tích cơ bản.

Về phạm vi chức năng, hệ thống được thiết kế để bao quát các nghiệp vụ thiết yếu trong quản lý đầu tư cá nhân. Điều này bao gồm việc quản lý tài khoản người dùng với các tính năng bảo mật cao, quản lý ví điện tử cá nhân với khả năng nạp rút tiền linh hoạt, khởi tạo và theo dõi chi tiết các danh mục đầu tư đa dạng, thực hiện giao dịch mua bán cổ phiếu trên thị trường một cách trực quan, và cập nhật thông tin thị trường chứng khoán theo thời gian thực. Bên cạnh những chức năng cơ bản này, dự án cũng tiên phong trong việc xem xét tiềm năng tích hợp các công cụ hỗ trợ thông minh như trợ lý AI, nhằm nâng cao trải nghiệm người dùng và cung cấp các tư vấn đầu tư cơ bản.

Tầm quan trọng của dự án này không chỉ dừng lại ở việc giải quyết các vấn đề kỹ thuật, mà còn góp phần vào việc phổ cập kiến thức tài chính và đầu tư trong cộng đồng. Thông qua việc cung cấp một công cụ dễ sử dụng và đáng tin cậy, hệ thống mong muốn thu hẹp khoảng cách giữa các nhà đầu tư chuyên nghiệp và những người mới bắt đầu, từ đó thúc đẩy sự phát triển bền vững của thị trường tài chính Việt Nam.

1. **Phân Tích Yêu Cầu**

Để đảm bảo hệ thống được phát triển đáp ứng đúng và đủ nhu cầu của người dùng cuối, giai đoạn phân tích yêu cầu đã được thực hiện một cách cẩn trọng và chi tiết. Quá trình này được tiến hành thông qua việc nghiên cứu các hệ thống tương tự trên thị trường, khảo sát nhu cầu người dùng tiềm năng, và phân tích các thách thức kỹ thuật có thể gặp phải. Kết quả của giai đoạn này là việc xác định rõ ràng các yêu cầu về mặt chức năng cũng như các yêu cầu phi chức năng đối với hệ thống.

**2.1 Yêu cầu chức năng**

Các yêu cầu chức năng định nghĩa một cách cụ thể những gì hệ thống phải thực hiện được, đồng thời mô tả các nghiệp vụ mà người dùng có thể tương tác trực tiếp. Những yêu cầu này được phân loại theo các nhóm chức năng chính, tạo nên một hệ thống logic và dễ quản lý.

Đầu tiên, về khía cạnh quản lý người dùng, hệ thống cần cung cấp khả năng cho phép người dùng mới thực hiện đăng ký tài khoản một cách đơn giản nhưng an toàn. Quy trình đăng ký này phải bao gồm việc xác thực email và các biện pháp bảo mật cơ bản để ngăn chặn các tài khoản giả mạo. Đối với những người dùng đã có tài khoản, chức năng đăng nhập và đăng xuất an toàn là yêu cầu bắt buộc, đi kèm với các cơ chế như xác thực hai yếu tố nếu cần thiết. Sau khi đăng nhập thành công, người dùng phải có khả năng quản lý và cập nhật toàn bộ thông tin cá nhân của mình, bao gồm không chỉ các thông tin cơ bản như họ tên, địa chỉ, số điện thoại, mà còn các chi tiết khác như ảnh đại diện và tùy chọn bảo mật cá nhân.

Tiếp theo, đối với khía cạnh quản lý ví điện tử, hệ thống phải hỗ trợ người dùng trong việc nạp tiền vào ví cá nhân thông qua các phương thức thanh toán đa dạng và thuận tiện, trong đó ưu tiên các phương thức phổ biến như chuyển khoản ngân hàng. Để đảm bảo tính minh bạch và đáng tin cậy, hệ thống cần có cơ chế kiểm tra và xác nhận các giao dịch nạp tiền một cách tự động hoặc bán tự động. Bên cạnh đó, chức năng rút tiền từ ví về tài khoản ngân hàng đã liên kết cũng cần được cung cấp với các biện pháp bảo mật tương ứng. Quan trọng không kém, người dùng phải có khả năng xem số dư ví hiện tại một cách real-time và truy cập được lịch sử chi tiết của mọi giao dịch đã thực hiện trên ví, bao gồm cả thông tin về thời gian, số tiền, và trạng thái giao dịch.

Để hỗ trợ các giao dịch tài chính một cách hiệu quả, chức năng quản lý tài khoản ngân hàng liên kết trở thành yếu tố không thể thiếu. Hệ thống cần cho phép người dùng thêm mới, chỉnh sửa thông tin, hoặc xóa các tài khoản ngân hàng đã liên kết với tài khoản của họ trên hệ thống. Đặc biệt, một tính năng quan trọng là khả năng thiết lập một tài khoản ngân hàng làm tài khoản mặc định cho các giao dịch nạp và rút tiền, giúp tối ưu hóa trải nghiệm người dùng và giảm thiểu các bước thao tác không cần thiết.

Trái tim của toàn bộ hệ thống chính là chức năng quản lý danh mục đầu tư, đây là nơi thể hiện giá trị cốt lõi của sản phẩm. Người dùng cần được phép tạo mới các danh mục đầu tư với khả năng tùy chỉnh cao, bao gồm việc cung cấp các thông tin như tên danh mục, mô tả chi tiết về chiến lược đầu tư, mục tiêu đầu tư dự kiến về mặt tài chính, và mức độ chấp nhận rủi ro cá nhân. Hệ thống phải có khả năng hiển thị danh sách tất cả các danh mục mà người dùng đang sở hữu một cách trực quan và dễ quản lý. Quan trọng hơn cả, người dùng cần có thể xem được chi tiết đầy đủ của từng danh mục, bao gồm danh sách các cổ phiếu đang nắm giữ, tỷ trọng của từng cổ phiếu trong tổng danh mục, giá trị thị trường hiện tại của từng mã cổ phiếu, và quan trọng nhất là tổng lãi lỗ của toàn bộ danh mục được tính toán một cách chính xác và cập nhật real-time. Ngoài ra, các thao tác chỉnh sửa thông tin danh mục và xóa danh mục cũng cần được hỗ trợ, với điều kiện hệ thống phải kiểm tra xem danh mục có còn chứa cổ phiếu hay không trước khi cho phép xóa, nhằm đảm bảo tính toàn vẹn của dữ liệu.

Liên quan trực tiếp và không thể tách rời với quản lý danh mục là khả năng thực hiện giao dịch tài sản, cụ thể là giao dịch cổ phiếu. Hệ thống phải cho phép người dùng thực hiện lệnh mua cổ phiếu vào một danh mục cụ thể, đồng thời tự động kiểm tra số dư trong ví điện tử để đảm bảo người dùng có đủ tài chính trước khi cho phép giao dịch diễn ra. Tương tự, khi người dùng muốn bán cổ phiếu từ một danh mục, hệ thống phải có khả năng kiểm tra số lượng cổ phiếu đang sở hữu để đảm bảo tính hợp lệ của giao dịch. Đặc biệt quan trọng, lịch sử của tất cả các giao dịch cổ phiếu phải được lưu trữ một cách đầy đủ và chi tiết, với khả năng cho phép người dùng xem toàn bộ lịch sử hoặc lọc theo từng danh mục cụ thể, giúp họ có cái nhìn tổng quan về các hoạt động đầu tư của mình.

Để cung cấp thông tin hỗ trợ cho việc đưa ra các quyết định đầu tư sáng suốt, chức năng theo dõi thị trường chứng khoán trở thành một yếu tố không thể thiếu. Hệ thống cần có khả năng hiển thị bảng giá cổ phiếu được cập nhật thường xuyên, lý tưởng là theo thời gian thực hoặc gần như thời gian thực. Bên cạnh đó, người dùng cần có khả năng tìm kiếm thông tin chi tiết của một mã cổ phiếu cụ thể, bao gồm các thông tin như giá hiện tại, biến động trong ngày, khối lượng giao dịch, và các chỉ số kỹ thuật cơ bản. Hơn nữa, việc hiển thị biểu đồ lịch sử biến động giá của từng mã cổ phiếu cũng là một tính năng quan trọng, giúp người dùng phân tích xu hướng và đưa ra các quyết định đầu tư dựa trên dữ liệu lịch sử.

Cuối cùng, một tính năng tiềm năng được xem xét để nâng cao trải nghiệm người dùng và tạo ra sự khác biệt cạnh tranh là khả năng tương tác với trợ lý AI. Chức năng này cho phép người dùng đặt các câu hỏi liên quan đến đầu tư, phân tích thị trường, hoặc các vấn đề tài chính cá nhân, và nhận được các phản hồi tư vấn cơ bản từ một hệ thống chatbot thông minh. Mặc dù đây không phải là tư vấn tài chính chuyên nghiệp, nhưng có thể giúp người dùng có thêm thông tin tham khảo và nâng cao kiến thức đầu tư của họ.

**2.2 Yêu Cầu Phi Chức Năng**

Bên cạnh các chức năng cụ thể mà hệ thống cần thực hiện, các yêu cầu phi chức năng đóng vai trò quyết định trong việc đảm bảo chất lượng tổng thể, hiệu suất hoạt động, và trải nghiệm người dùng của hệ thống. Những yêu cầu này thường khó đo lường một cách định lượng nhưng lại có tác động trực tiếp đến sự thành công của sản phẩm.

Về khía cạnh giao diện người dùng và trải nghiệm người dùng, hệ thống phải được thiết kế với triết lý "user-centric", nghĩa là đặt người dùng làm trung tâm của mọi quyết định thiết kế. Giao diện phải thân thiện, trực quan và dễ sử dụng, ngay cả đối với những người dùng không có nhiều kinh nghiệm về công nghệ hoặc đầu tư tài chính. Điều này đòi hỏi việc áp dụng các nguyên tắc thiết kế UX/UI hiện đại, sử dụng các màu sắc phù hợp với lĩnh vực tài chính, bố cục rõ ràng và logic, cũng như đảm bảo tính nhất quán trong toàn bộ hệ thống. Trải nghiệm người dùng tổng thể cần đảm bảo sự mượt mà và tự nhiên, với các luồng thao tác được thiết kế logic và dễ hiểu, giúp người dùng có thể hoàn thành các tác vụ một cách nhanh chóng và hiệu quả.

Hiệu năng hệ thống là một yếu tố then chốt, đặc biệt quan trọng trong lĩnh vực tài chính nơi mà tốc độ có thể quyết định đến thành công hay thất bại của một giao dịch. Hệ thống phải hoạt động ổn định với thời gian phản hồi nhanh chóng đối với mọi yêu cầu từ người dùng, lý tưởng là dưới 2 giây cho hầu hết các thao tác thông thường và dưới 5 giây cho các thao tác phức tạp như tính toán danh mục hoặc tải biểu đồ. Quan trọng hơn, hệ thống phải duy trì được hiệu năng này ngay cả khi có nhiều người dùng truy cập đồng thời, đòi hỏi việc thiết kế kiến trúc có khả năng mở rộng và tối ưu hóa hiệu quả.

Bảo mật thông tin người dùng và dữ liệu giao dịch phải được đặt lên hàng đầu, đây là yếu tố quyết định đến sự tin tưởng của người dùng đối với hệ thống. Điều này bao gồm việc áp dụng các biện pháp mã hóa dữ liệu tiên tiến cho cả dữ liệu được truyền tải và dữ liệu được lưu trữ, thiết lập hệ thống kiểm soát truy cập chặt chẽ với các cấp độ phân quyền khác nhau, và triển khai các cơ chế phòng chống các loại tấn công phổ biến như SQL injection, cross-site scripting (XSS), và cross-site request forgery (CSRF). Bên cạnh đó, hệ thống cũng cần có khả năng audit trail để theo dõi và ghi lại mọi hoạt động quan trọng, giúp phát hiện và ứng phóng với các mối đe dọa bảo mật tiềm ẩn.

Khả năng mở rộng và tính bảo trì là những yếu tố quan trọng để đảm bảo tính bền vững lâu dài của hệ thống. Hệ thống cần được thiết kế với kiến trúc modular và scalable, cho phép dễ dàng bổ sung các tính năng mới hoặc nâng cấp hiệu năng trong tương lai mà không gây ảnh hưởng lớn đến kiến trúc hiện tại hoặc làm gián đoạn hoạt động của hệ thống. Đồng thời, mã nguồn phải được tổ chức một cách rõ ràng và tuân thủ các best practices trong lập trình, đảm bảo tính dễ bảo trì, dễ hiểu, và dễ mở rộng cho các developer khác trong tương lai.

Cuối cùng, để phục vụ tốt nhất cho đối tượng người dùng mục tiêu tại Việt Nam, toàn bộ giao diện và các thông báo trong hệ thống sẽ sử dụng ngôn ngữ Tiếng Việt. Điều này không chỉ bao gồm việc dịch thuật đơn thuần mà còn đòi hỏi việc địa hóa (localization) các tính năng cho phù hợp với thói quen sử dụng, văn hóa, và quy định pháp lý tại Việt Nam, chẳng hạn như định dạng số, định dạng ngày tháng, và các quy định về giao dịch tài chính.

1. **Thiết Kế Hệ Thống**

Sau giai đoạn phân tích và đặc tả yêu cầu một cách chi tiết và toàn diện, công đoạn thiết kế hệ thống được triển khai với mục tiêu xây dựng một kiến trúc kỹ thuật vững chắc và một cấu trúc dữ liệu hợp lý. Giai đoạn này đóng vai trò như cầu nối quan trọng giữa ý tưởng và hiện thực, tạo ra nền tảng vững chắc cho quá trình phát triển và triển khai sản phẩm cuối cùng.

**3.1 Kiến Trúc Hệ Thống**

Hệ thống quản lý danh mục đầu tư được đề xuất xây dựng dựa trên mô hình kiến trúc Client-Server ba lớp, một mô hình đã được chứng minh là hiệu quả và phù hợp cho các ứng dụng web có độ phức tạp trung bình đến cao. Mô hình này không chỉ đảm bảo tính ổn định và bảo mật mà còn cung cấp khả năng mở rộng tốt cho tương lai.

Lớp đầu tiên là Lớp Trình diễn (Presentation

Layer), thường được gọi là Front-End, đây chính là nơi người dùng tương tác trực tiếp với hệ thống. Lớp này được phát triển sử dụng các công nghệ web chuẩn bao gồm HTML5 cho cấu trúc nội dung, CSS3 cho việc tạo kiểu và layout, và JavaScript cho các tương tác động. Để đảm bảo tính nhất quán và chuyên nghiệp trong thiết kế, nhóm phát triển đã lựa chọn thư viện Bootstrap, một framework CSS nổi tiếng, giúp xây dựng giao diện người dùng responsive và thẩm mỹ. Việc sử dụng Bootstrap không chỉ tiết kiệm thời gian phát triển mà còn đảm bảo giao diện hoạt động tốt trên nhiều thiết bị khác nhau, từ desktop đến mobile.

Lớp thứ hai là Lớp Ứng dụng (Application Layer), hay Back-End, đây là trái tim của toàn bộ hệ thống nơi chứa đựng toàn bộ logic nghiệp vụ và xử lý các yêu cầu từ người dùng. Để xây dựng lớp này, nhóm đã lựa chọn Django, một web framework mạnh mẽ và mature của Python. Django được chọn không chỉ vì tính ổn định và bảo mật cao mà còn vì khả năng phát triển nhanh chóng và cộng đồng hỗ trợ rộng lớn. Framework này cung cấp nhiều tính năng built-in như ORM (Object-Relational Mapping), authentication system, admin interface, và các biện pháp bảo mật tự động, giúp đẩy nhanh quá trình phát triển đồng thời đảm bảo chất lượng code.

Lớp thứ ba và cũng là lớp cuối cùng là Lớp Dữ liệu (Data Layer), bao gồm cơ sở dữ liệu và các hệ thống lưu trữ. Sau quá trình nghiên cứu và so sánh các giải pháp, nhóm đã quyết định sử dụng PostgreSQL, một hệ quản trị cơ sở dữ liệu quan hệ mã nguồn mở có hiệu năng cao và độ tin cậy tốt. PostgreSQL được lựa chọn vì khả năng xử lý các truy vấn phức tạp, hỗ trợ tốt cho các kiểu dữ liệu đa dạng, và tính năng backup/recovery mạnh mẽ, những yếu tố quan trọng trong một hệ thống tài chính.

Bên cạnh ba lớp chính, hệ thống còn tích hợp với nhiều dịch vụ bên ngoài để nâng cao chức năng và trải nghiệm người dùng. Đầu tiên là dịch vụ xác thực, nhóm đang cân nhắc sử dụng Auth0, một dịch vụ quản lý danh tính và xác thực đáng tin cậy. Auth0 không chỉ giúp đơn giản hóa quy trình đăng nhập/đăng ký mà còn cung cấp các tính năng bảo mật nâng cao như multi-factor authentication và social login.

Để cung cấp dữ liệu thị trường chứng khoán Việt Nam một cách chính xác và kịp thời, hệ thống tích hợp với thư viện vnstock, một công cụ mạnh mẽ được phát triển đặc biệt cho thị trường Việt Nam. Thư viện này cho phép thu thập dữ liệu giá cổ phiếu, thông tin công ty, và các chỉ số thị trường một cách tự động và đáng tin cậy.

Một tính năng đổi mới mà nhóm muốn tích hợp là trợ lý AI, sử dụng Gemini API của Google. API này sẽ được sử dụng để xây dựng một chatbot thông minh có khả năng trả lời các câu hỏi cơ bản về đầu tư và tài chính, giúp nâng cao trải nghiệm người dùng và cung cấp giá trị gia tăng cho sản phẩm.

Cuối cùng, để hỗ trợ chức năng nạp tiền vào ví điện tử, hệ thống sử dụng một giải pháp sáng tạo thông qua Google Script API. Giải pháp này cho phép tạo ra các endpoint API tùy chỉnh để xác minh các giao dịch chuyển khoản ngân hàng, đảm bảo tính chính xác và tự động hóa trong quy trình nạp tiền.

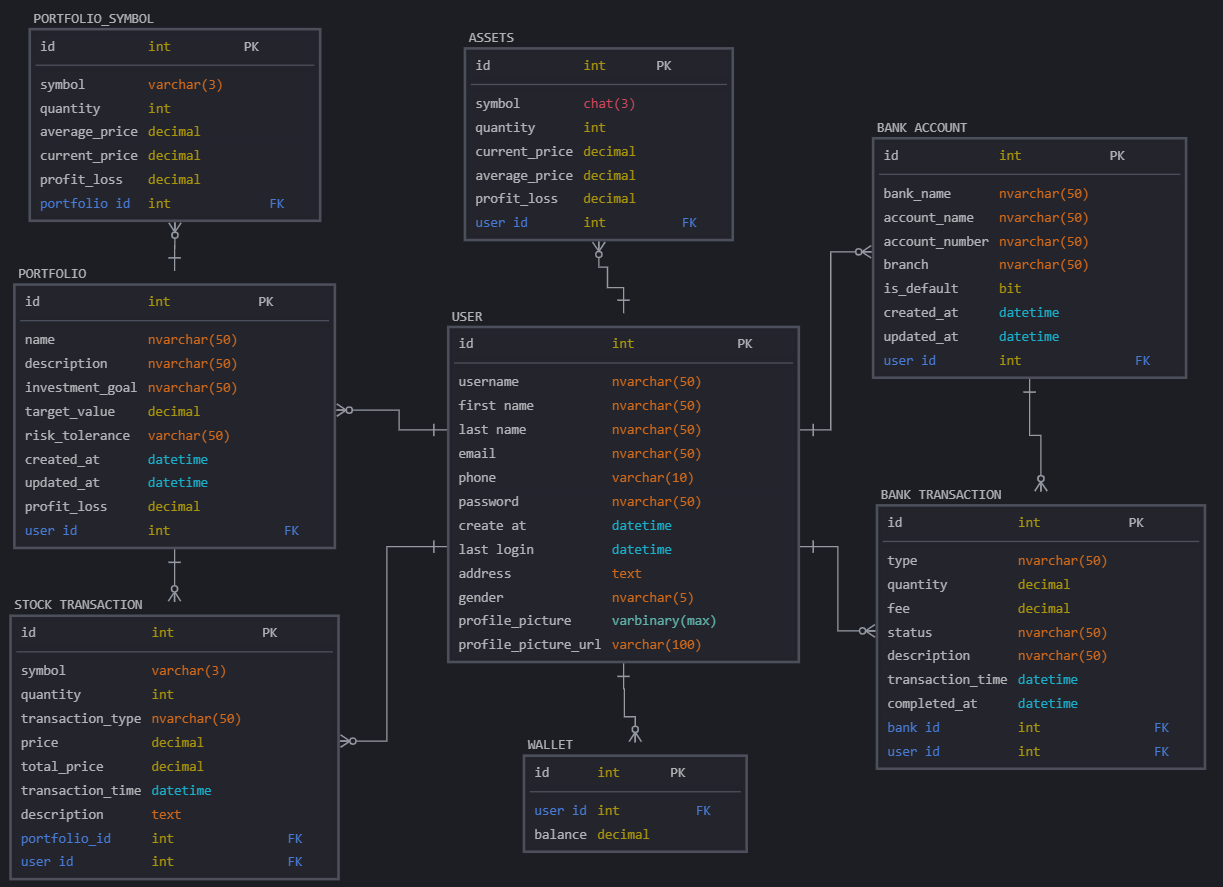
**3.2 Thiết kế cơ sở dữ liệu (Database Design)**

**3.2.1 Sơ đồ quan hệ trực thể**

Thiết kế cơ sở dữ liệu đóng vai trò nền tảng quan trọng trong việc đảm bảo tính toàn vẹn, hiệu quả và khả năng mở rộng của hệ thống quản lý danh mục đầu tư. Quá trình thiết kế này được thực hiện dựa trên các nguyên tắc chuẩn hóa cơ sở dữ liệu quan hệ nhằm loại bỏ sự dư thừa dữ liệu và đảm bảo tính nhất quán thông tin.

**Sơ Đồ Quan Hệ Thực Thể (Entity Relationship Diagram):**

Để mô tả trực quan cấu trúc dữ liệu và mối quan hệ giữa các thực thể trong hệ thống, sơ đồ ERD đã được xây dựng chi tiết. Sơ đồ này thể hiện rõ ràng các thực thể chính bao gồm người dùng (User), ví điện tử (Wallet), danh mục đầu tư (Portfolio), giao dịch cổ phiếu (StockTransaction), tài khoản ngân hàng (BankAccount), và các tài sản (Assets), cùng với các mối quan hệ một-một, một-nhiều, và nhiều-nhiều giữa chúng.



*Hình 2: Sơ đồ quan hệ thực thể (ERD) của hệ thống quản lý danh mục đầu tư*

Sơ đồ ERD cho thấy tính logic và tối ưu của thiết kế, trong đó mỗi thực thể đều có vai trò rõ ràng và các mối quan hệ được thiết lập phù hợp với nghiệp vụ thực tế. Điều này đảm bảo việc truy xuất dữ liệu hiệu quả và duy trì tính toàn vẹn tham chiếu giữa các bảng.

**Mô Tả Chi Tiết Các Bảng Dữ Liệu**

Cấu trúc cơ sở dữ liệu được thiết kế với 8 bảng chính, mỗi bảng đảm nhận một nhóm chức năng cụ thể và có mối liên kết chặt chẽ với nhau.

**Bảng User (portfolio.User)**

Bảng User đóng vai trò trung tâm trong hệ thống, lưu trữ toàn bộ thông tin cá nhân và thông tin xác thực của người dùng. Bảng này bao gồm các trường dữ liệu như id (khóa chính), username và password (đã được mã hóa để đảm bảo bảo mật), email, first\_name, last\_name để lưu thông tin cá nhân cơ bản. Ngoài ra, các trường phone, address, gender cung cấp thông tin chi tiết hơn về người dùng.

Đặc biệt, hệ thống hỗ trợ hai phương thức lưu trữ ảnh đại diện thông qua profile\_picture (kiểu ImageField lưu đường dẫn file trên server) và profile\_picture\_url (kiểu URLField lưu đường dẫn URL). Trường auth0\_user\_id được thiết kế để tích hợp với dịch vụ xác thức Auth0, trong khi các trường date\_joined, last\_login, is\_staff, is\_active quản lý trạng thái và quyền hạn của người dùng.

|  |  |  |  |
| --- | --- | --- | --- |
| **Tên trường** | **Kiểu dữ liệu** | **Ràng buộc** | **Mô tả** |
| id | Integer | PK, Auto Increment | Khóa chính |
| username | CharField(150) | Unique, Not Null | Tên đăng nhập |
| password | CharField(128) | Not Null | Mật khẩu đã mã hóa |
| email | EmailField(254) | Unique, Not Null | Email |
| first\_name | CharField(150) | Null | Tên |
| last\_name | CharField(150) | Null | Họ |
| phone | CharField(15) | Null | Số điện thoại |
| address | TextField | Null | Địa chỉ |
| gender | CharField(10) | Null | Giới tính |
| profile\_picture | ImageField | Null | Ảnh đại diện (file) |
| profile\_picture\_url | URLField | Null | URL ảnh đại diện |
| auth0\_user\_id | CharField(100) | Null, Unique | ID từ Auth0 |
| date\_joined | DateTimeField | Auto Add Now | Ngày tham gia |
| last\_login | DateTimeField | Null | Lần đăng nhập cuối |
| is\_staff | BooleanField | Default False | Quyền admin |
| is\_active | BooleanField | Default True | Trạng thái hoạt động |

**Bảng Wallet (portfolio.Wallet)**

Bảng Wallet thiết lập mối quan hệ một-một với bảng User, đảm bảo mỗi người dùng chỉ có một ví điện tử duy nhất. Cấu trúc đơn giản nhưng hiệu quả này bao gồm id (khóa chính), user (khóa ngoại liên kết với User), và balance (kiểu Decimal để đảm bảo độ chính xác trong các phép tính tài chính). Việc sử dụng kiểu Decimal thay vì Float là một quyết định thiết kế quan trọng nhằm tránh các lỗi làm tròn có thể xảy ra trong các giao dịch tài chính.

|  |  |  |  |
| --- | --- | --- | --- |
| **Tên trường** | **Kiểu dữ liệu** | **Ràng buộc** | **Mô tả** |
| id | Integer | PK,Auto Increment | Khóa chính |
| user | ForeignKey(User) | One-to-One, Not Null | Liên kết người dùng |
| balance | DecimalField(15,2) | Default 0, >= 0 | Số dư ví |
| created\_at | DateTimeField | Auto Add Now | Ngày tạo |
| updated\_at | DateTimeField | Auto Update | Ngày cập nhật cuối |

**Bảng BankAccount (portfolio.BankAccount)**

Nhằm hỗ trợ các giao dịch nạp và rút tiền, bảng BankAccount lưu trữ thông tin các tài khoản ngân hàng mà người dùng liên kết với hệ thống. Cấu trúc bảng bao gồm id (khóa chính), user (khóa ngoại tham chiếu đến User tạo mối quan hệ một-nhiều), bank\_name, account\_name, account\_number, và branch để lưu thông tin chi tiết về tài khoản ngân hàng.

Trường is\_default (kiểu Boolean) cho phép người dùng thiết lập tài khoản ngân hàng mặc định, tối ưu hóa trải nghiệm giao dịch. Các trường created\_at và updated\_at theo dõi thời gian tạo và cập nhật thông tin, hỗ trợ việc kiểm soát phiên bản và audit trail.

|  |  |  |  |
| --- | --- | --- | --- |
| **Tên trường** | **Kiểu dữ liệu** | **Ràng buộc** | **Mô tả** |
| id | Integer | PK, AutoIncrement | Khóa chính |
| user | ForeignKey(User) | Not Null, ON DELETE CASCADE | Chủ tài khoản |
| bank\_name | CharField(100) | Not Null | Tên ngân hàng |
| account\_name | CharField(200) | Not Null | Tên chủ tài khoản |
| account\_number | CharField(50) | Not Null | Số tài khoản |
| branch | CharField(200) | Null | Chi nhánh |
| is\_default | BooleanField | Default False | TK mặc định |
| created\_at | DateTimeField | Auto Add Now | Ngày tạo |
| updated\_at | DateTimeField | Auto Update | Ngày cập nhật |

**Bảng BankTransaction (portfolio.BankTransaction)**

Bảng BankTransaction ghi lại đầy đủ lịch sử các giao dịch tài chính giữa ví điện tử và tài khoản ngân hàng. Thiết kế này bao gồm id (khóa chính), user (khóa ngoại tham chiếu User), bank\_account (khóa ngoại tham chiếu BankAccount, cho phép null để xử lý các trường hợp đặc biệt), type (CharField phân biệt loại giao dịch như 'deposit' hoặc 'withdraw'), quantity và fee (kiểu Decimal cho số tiền giao dịch và phí).

Trường status (CharField) quản lý trạng thái giao dịch với các giá trị như 'pending', 'completed', 'failed', hoặc 'cancelled', cho phép theo dõi tiến trình xử lý giao dịch. Các trường transaction\_time và completed\_at (cho phép null) ghi nhận thời điểm khởi tạo và hoàn thành giao dịch, cung cấp thông tin quan trọng cho việc đối soát và báo cáo.

|  |  |  |  |
| --- | --- | --- | --- |
| **Tên trường** | **Kiểu dữ liệu** | **Ràng buộc** | **Mô tả** |
| id | Integer | PK, Auto Increment | Khóa chính |
| user | ForeignKey(User) | Not Null | Người thực hiện |
| bank\_account | ForeignKey(BankAccount) | Null, ON DELETE SET NULL | TK ngân hàng |
| type | CharField(20) | Not Null | Loại GD (deposit/withdraw) |
| amount | DecimalField(15,2) | Not Null, > 0 | Số tiền |
| fee | DecimalField(10,2) | Default 0, >= 0 | Phí giao dịch |
| status | CharField(20) | Default 'pending' | Trạng thái |
| description | TextField | Null | Mô tả |
| transaction\_code | CharField(50) | Null, Unique | Mã giao dịch |
| transaction\_time | DateTimeField | Auto Add Now | Thời gian yêu cầu |
| completed\_at | DateTimeField | Null | Thời gian hoàn thành |

**Bảng Portfolio (portfolio.Portfolio)**

Là trung tâm của chức năng quản lý danh mục đầu tư, bảng Portfolio lưu trữ thông tin về các danh mục mà người dùng tạo lập. Cấu trúc bao gồm id (khóa chính), user (khóa ngoại tạo mối quan hệ một-nhiều với User), name và description mô tả danh mục. Trường investment\_goal lưu trữ mục tiêu đầu tư, target\_value (kiểu Decimal, cho phép null) xác định giá trị mục tiêu, và risk\_tolerance (CharField) định nghĩa mức độ chấp nhận rủi ro.

Các trường created\_at và updated\_at theo dõi thời gian tạo và cập nhật danh mục. Đặc biệt, trường profit\_loss (kiểu Decimal, được đặt tên là profiss\_loss trong model để tránh từ khóa dành riêng) tính toán và lưu trữ tổng lãi lỗ của danh mục, hỗ trợ việc đánh giá hiệu quả đầu tư.

|  |  |  |  |
| --- | --- | --- | --- |
| **Tên trường** | **Kiểu dữ liệu** | **Ràng buộc** | **Mô tả** |
| id | Integer | PK, Auto Increment | Khóa chính |
| user | ForeignKey(User) | Not Null | Chủ danh mục |
| name | CharField(200) | Not Null | Tên danh mục |
| description | TextField | Null | Mô tả |
| investment\_goal | CharField(100) | Null | Mục tiêu đầu tư |
| target\_value | DecimalField(15,2) | Null, > 0 | Giá trị mục tiêu |
| risk\_tolerance | CharField(20) | Default 'medium' | Mức chấp nhận rủi ro |
| created\_at | DateTimeField | Auto Add Now | Ngày tạo |
| updated\_at | DateTimeField | Auto Update | Ngày cập nhật |
| total\_value | DecimalField(15,2) | Default 0 | Tổng giá trị hiện tại |
| profit\_loss | DecimalField(15,2) | Default 0 | Lãi/ lỗ |

**Bảng PortfolioSymbol (portfolio.PortfolioSymbol)**

Bảng PortfolioSymbol chi tiết hóa các cổ phiếu cụ thể trong từng danh mục đầu tư, tạo mối quan hệ một-nhiều với bảng Portfolio. Cấu trúc bao gồm id (khóa chính), portfolio (khóa ngoại), symbol (CharField với độ dài 3 ký tự theo chuẩn mã cổ phiếu Việt Nam), quantity (Integer) cho số lượng cổ phiếu đang nắm giữ.

Các trường average\_price, current\_price, và profit\_loss (kiểu Decimal) lưu trữ giá mua trung bình, giá hiện tại, và lãi lỗ của từng mã cổ phiếu. Thiết kế này cho phép tính toán hiệu quả tỷ trọng và hiệu suất của từng cổ phiếu trong danh mục tổng thể.

|  |  |  |  |
| --- | --- | --- | --- |
| **Tên trường** | **Kiểu dữ liệu** | **Ràng buộc** | **Mô tả** |
| id | Integer | PK, Auto Increment | Khóa chính |
| portfolio | ForeignKey(Portfolio) | Not Null, ON DELETE CASCADE | Danh mục |
| symbol | CharField(10) | Not Null | Mã cổ phiếu |
| quantity | Integer | Not Null, >= 0 | Số lượng |
| average\_price | DecimalField(10,2) | Not Null, > 0 | Giá trung bình |
| current\_price | DecimalField(10,2) | Default 0 | Giá hiện tại |
| profit\_loss | DecimalField(15,2) | Default 0 | Lãi/lỗ |
| profit\_loss\_percent | DecimalField(5,2) | Default 0 | % Lãi/lỗ |
| last\_updated | DateTimeField | Auto Update | Cập nhật cuối |

**Bảng StockTransaction (portfolio.StockTransaction)**

Nhằm duy trì tính minh bạch và khả năng truy xuất lịch sử giao dịch, bảng StockTransaction ghi lại mọi giao dịch mua bán cổ phiếu. Cấu trúc bao gồm id (khóa chính), user (khóa ngoại), portfolio (khóa ngoại, cho phép null để xử lý các giao dịch đặc biệt), symbol (CharField 3 ký tự), quantity (Integer), và transaction\_type (CharField phân biệt 'buy' hoặc 'sell').

Các trường price và total\_price (kiểu Decimal) lưu giá giao dịch và tổng giá trị, transaction\_time ghi nhận thời điểm thực hiện, và description cho phép ghi chú bổ sung. Thiết kế này đảm bảo tính audit trail hoàn chỉnh cho mọi hoạt động giao dịch.

|  |  |  |  |
| --- | --- | --- | --- |
| **Tên trường** | **Kiểu dữ liệu** | **Ràng buộc** | **Mô tả** |
| id | Integer | PK, Auto Increment | Khóa chính |
| user | ForeignKey(User) | Not Null | Người thực hiện |
| portfolio | ForeignKey(Portfolio) | Null, ON DELETE SET NULL | Danh mục |
| symbol | CharField(10) | Not Null | Mã cổ phiếu |
| quantity | Integer | Not Null, > 0 | Số lượng |
| transaction\_type | CharField(10) | Not Null | Loại GD (buy/sell) |
| price | DecimalField(10,2) | Not Null, > 0 | Giá |
| total\_amount | DecimalField(15,2) | Not Null | Tổng tiền |
| fee | DecimalField(10,2) | Default 0 | Phí giao dịch |
| transaction\_time | DateTimeField | Auto Add Now | Thời gian GD |
| description | TextField | Null | Ghi chú |
| status | CharField(20) | Default 'completed' | Trạng thái |

**Bảng Assets (portfolio.Assets)**

Bảng Assets quản lý thông tin tổng hợp về các mã cổ phiếu trong hệ thống, đóng vai trò như một registry trung tâm. Mối quan hệ với User thông qua khóa ngoại user được thiết kế để theo dõi người dùng nào đã từng tương tác hoặc quan tâm đến các mã tài sản cụ thể, khác biệt với PortfolioSymbol thể hiện sự sở hữu thực tế trong một danh mục.

Cấu trúc bao gồm id (khóa chính), user (khóa ngoại), symbol (CharField 10 ký tự để hỗ trợ các mã dài hơn), name (tên công ty), quantity (Integer có thể đại diện cho tổng số lượng được hệ thống theo dõi), current\_price, average\_price, và profit\_loss (kiểu Decimal). Dữ liệu trong bảng này được đồng bộ hóa và cập nhật tự động từ thư viện vnstock, đảm bảo tính chính xác và kịp thời của thông tin thị trường.

|  |  |  |  |
| --- | --- | --- | --- |
| **Tên trường** | **Kiểu dữ liệu** | **Ràng buộc** | **Mô tả** |
| id | Integer | PK, Auto Increment | Khóa chính |
| symbol | CharField(10) | Unique, Not Null | Mã cổ phiếu |
| company\_name | CharField(200) | Not Null | Tên công ty |
| exchange | CharField(10) | Not Null | Sàn giao dịch |
| sector | CharField(100) | Null | Ngành |
| current\_price | DecimalField(10,2) | Default 0 | Giá hiện tại |
| previous\_close | DecimalField(10,2) | Default 0 | Giá đóng cửa trước |
| change\_percent | DecimalField(5,2) | Default 0 | % Thay đổi |
| volume | BigInteger | Default 0 | Khối lượng |
| market\_cap | DecimalField(20,2) | Null | Vốn hóa |
| last\_updated | DateTimeField | Auto Update | Cập nhật cuối |

**3.3 Thiết kế giao diện người dùng(UI/UX)**

Thiết kế giao diện người dùng đóng vai trò then chốt trong việc quyết định sự thành công của một ứng dụng quản lý đầu tư. Mục tiêu chính của quá trình thiết kế này không chỉ dừng lại ở việc tạo ra một giao diện đẹp mắt, mà còn phải đảm bảo trải nghiệm người dùng trực quan, hiệu quả và dễ tiếp cận. Trong bối cảnh thị trường tài chính đầy biến động, người dùng cần một công cụ cho phép họ thực hiện các quyết định đầu tư một cách nhanh chóng và chính xác, điều này đòi hỏi giao diện phải được thiết kế với sự chú ý đặc biệt đến tính khoa học và tâm lý người dùng.

**3.3.1. Nguyên Tắc Thiết Kế**

Việc xây dựng một hệ thống giao diện hiệu quả cần tuân thủ các nguyên tắc thiết kế cơ bản, trong đó tính nhất quán đóng vai trò nền tảng. Nguyên tắc nhất quán (Consistency) không chỉ thể hiện qua việc sử dụng thống nhất màu sắc, font chữ, kích thước nút bấm và biểu tượng trên toàn bộ ứng dụng, mà còn bao gồm cả cách thức tương tác và luồng điều hướng. Khi người dùng làm quen với một mẫu tương tác ở một phần của ứng dụng, họ có thể dự đoán và áp dụng kinh nghiệm này cho các phần khác, từ đó giảm thiểu thời gian học tập và nâng cao hiệu quả sử dung.

Đồng thời, nguyên tắc đơn giản (Simplicity) yêu cầu loại bỏ những yếu tố thừa thãi có thể gây phân tán sự chú ý của người dùng. Trong lĩnh vực tài chính, thông tin có tính chất phức tạp và nhạy cảm, do đó việc trình bày cần tập trung vào những dữ liệu quan trọng nhất và các chức năng cốt lõi. Điều này không có nghĩa là loại bỏ hoàn toàn các tính năng nâng cao, mà là tổ chức chúng theo cấu trúc phân cấp hợp lý, cho phép người dùng từng bước tiếp cận các chức năng phức tạp hơn khi cần thiết.

Hơn nữa, nguyên tắc phản hồi (Feedback) đặc biệt quan trọng trong ứng dụng tài chính, nơi mỗi hành động đều có thể tác động trực tiếp đến tài sản của người dùng. Hệ thống cần cung cấp phản hồi rõ ràng, kịp thời cho mọi thao tác, từ những tương tác đơn giản như nhấp chuột đến các giao dịch phức tạp như mua bán cổ phiếu. Phản hồi này không chỉ bao gồm thông báo thành công hay thất bại, mà còn cần thể hiện trạng thái tải dữ liệu, tiến trình xử lý và các cảnh báo về rủi ro khi cần thiết.

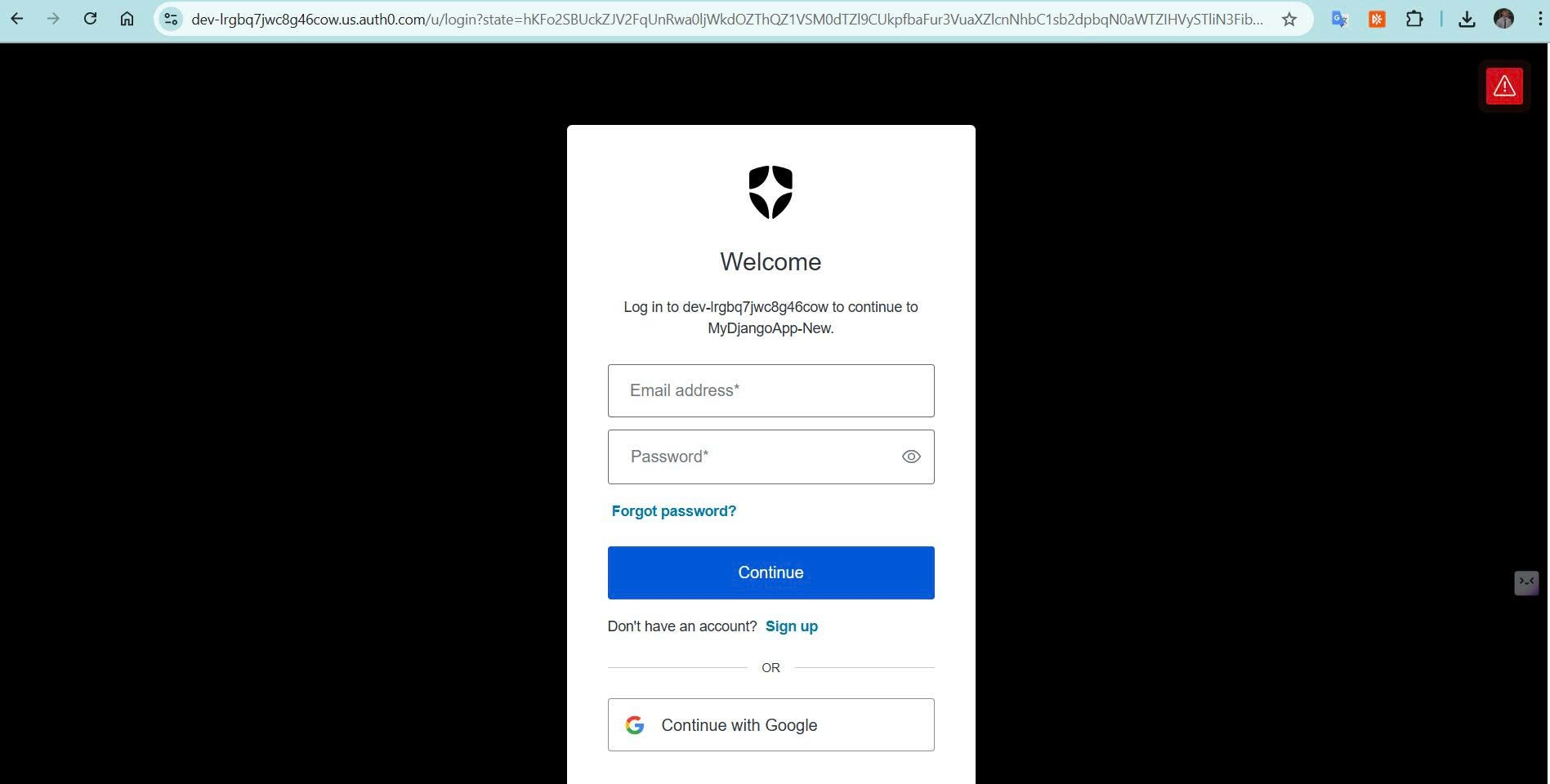
Bên cạnh đó, khả năng khám phá (Discoverability) đảm bảo rằng các chức năng và thông tin quan trọng có thể được tìm thấy một cách tự nhiên thông qua thiết kế giao diện. Điều này đạt được bằng cách sử dụng các quy ước thiết kế phổ biến, tổ chức menu logic và cung cấp các gợi ý trực quan để hướng dẫn người dùng khám phá các tính năng mới.

Cuối cùng, trong thời đại công nghệ di động phát triển mạnh mẽ, thiết kế responsive trở thành yêu cầu bắt buộc. Giao diện phải thích ứng hoàn hảo với mọi kích thước màn hình, từ màn hình desktop rộng lớn đến smartphone nhỏ gọn, đảm bảo trải nghiệm người dùng nhất quán và hiệu quả trên mọi thiết bị.

**3.3.2. Phân Tích Các Giao Diện Chính**

**Hệ Thống Xác Thực Người Dùng**

Trang đăng nhập và đăng ký đóng vai trò cổng chính để người dùng truy cập vào hệ thống, do đó thiết kế của nó cần tạo ấn tượng chuyên nghiệp và đáng tin cậy ngay từ lần tiếp xúc đầu tiên.



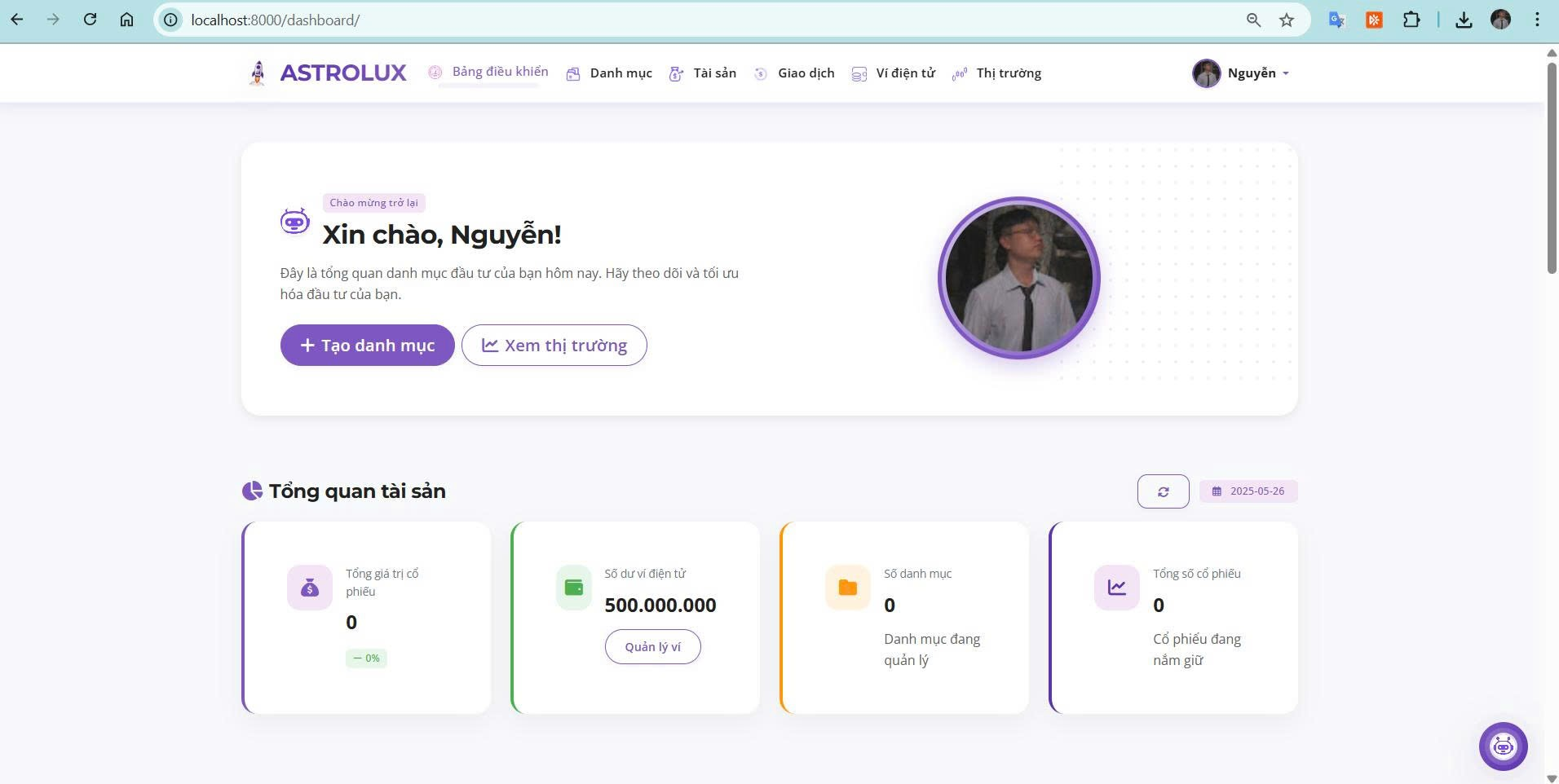
*Hình 2.1: Giao diện trang đăng nhập và đăng ký*

Form đăng nhập được thiết kế tối giản với hai trường chính là email/tên đăng nhập và mật khẩu, kèm theo checkbox "Ghi nhớ đăng nhập" để cải thiện trải nghiệm người dùng thường xuyên. Đặc biệt, việc tích hợp tùy chọn đăng nhập qua Auth0 không chỉ nâng cao bảo mật mà còn giảm thiểu rào cản cho người dùng mới. Liên kết "Quên mật khẩu?" được đặt ở vị trí dễ nhận biết, trong khi nút chuyển đổi sang form đăng ký được thiết kế rõ ràng để hướng dẫn người dùng chưa có tài khoản.

Form đăng ký yêu cầu thông tin cơ bản bao gồm tên đăng nhập, email, mật khẩu và xác nhận mật khẩu. Để nâng cao trải nghiệm, hệ thống tích hợp tính năng kiểm tra độ mạnh mật khẩu theo thời gian thực và xác thực email ngay lập tức để tránh lỗi nhập liệu. Điều khoản sử dụng và chính sách bảo mật được trình bày dưới dạng checkbox bắt buộc, đảm bảo tính minh bạch và tuân thủ quy định pháp lý.

**Bảng Điều Khiển Tổng Quan**

Dashboard đóng vai trò là trung tâm thông tin, nơi người dùng có thể nắm bắt tổng quan về tình hình tài chính và đầu tư của mình trong một cái nhìn duy nhất.



*Hình 2.2: Giao diện Dashboard tổng quan*

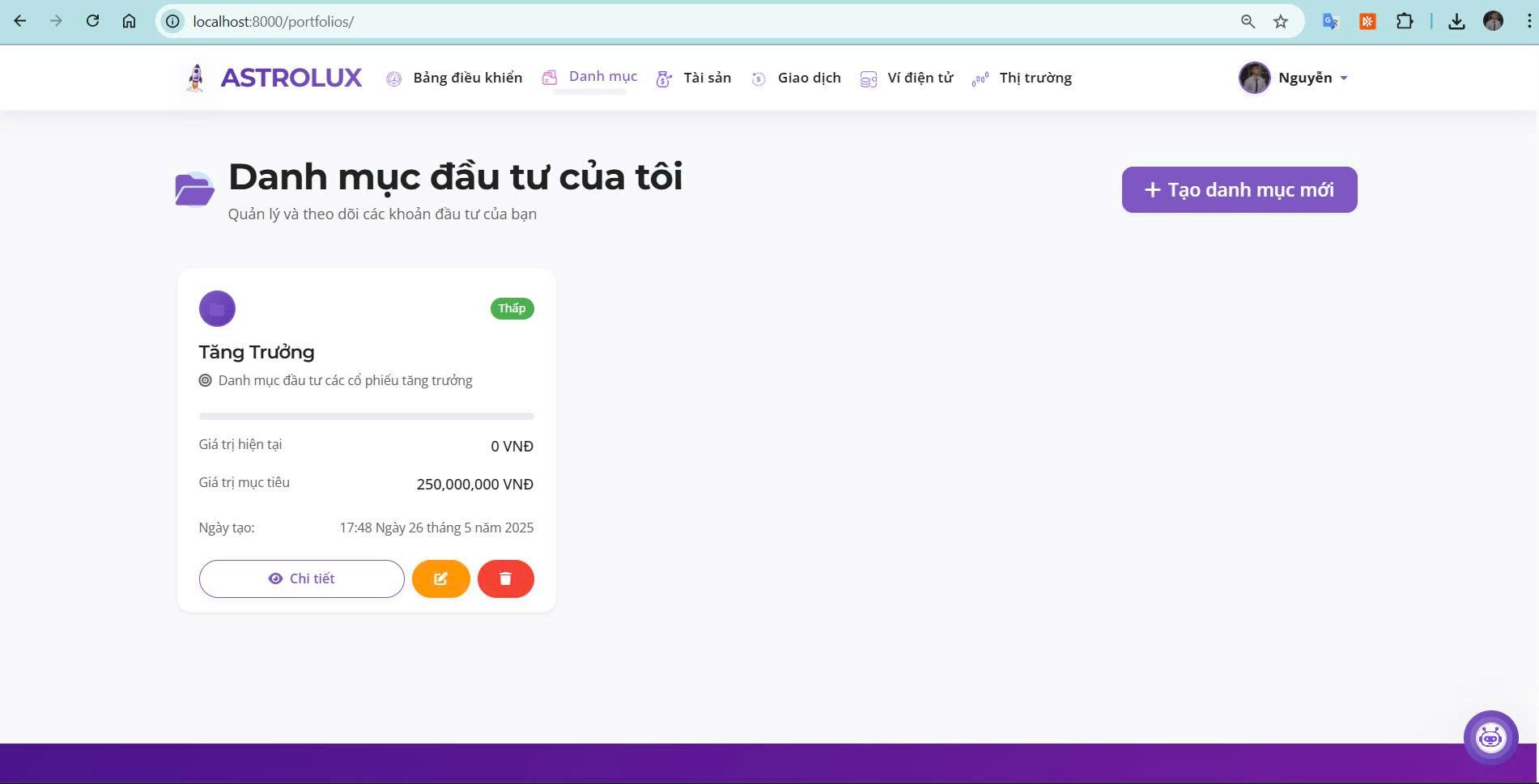
Khu vực trên cùng hiển thị số dư ví hiện tại với định dạng rõ ràng, kèm theo thông tin tổng giá trị tài sản và tỷ lệ thay đổi so với kỳ trước. Những con số này được làm nổi bật bằng màu sắc phù hợp - xanh lá cho tăng trưởng tích cực và đỏ cho giảm giá trị - giúp người dùng nhanh chóng đánh giá tình hình.

Biểu đồ phân bổ tài sản được thiết kế dưới dạng biểu đồ tròn tương tác, cho phép người dùng xem tỷ trọng của từng loại tài sản như tiền mặt, cổ phiếu, và các khoản đầu tư khác. Khi người dùng hover hoặc nhấp vào từng phần, chi tiết sẽ được hiển thị trong tooltip hoặc panel bên cạnh.

Danh sách các danh mục đầu tư được trình bày dưới dạng card layout, mỗi card chứa thông tin tóm tắt về tên danh mục, tổng giá trị, số lượng cổ phiếu và tỷ lệ lãi/lỗ. Thiết kế này không chỉ tiết kiệm không gian mà còn cho phép quét thông tin nhanh chóng.

**Hệ Thống Quản Lý Danh Mục Đầu Tư**

Giao diện quản lý danh mục đầu tư được thiết kế theo mô hình master-detail, trong đó panel bên trái hiển thị danh sách các danh mục và panel chính thể hiện chi tiết danh mục được chọn.



*Hình 2.3: Giao diện quản lý danh mục đầu tư*

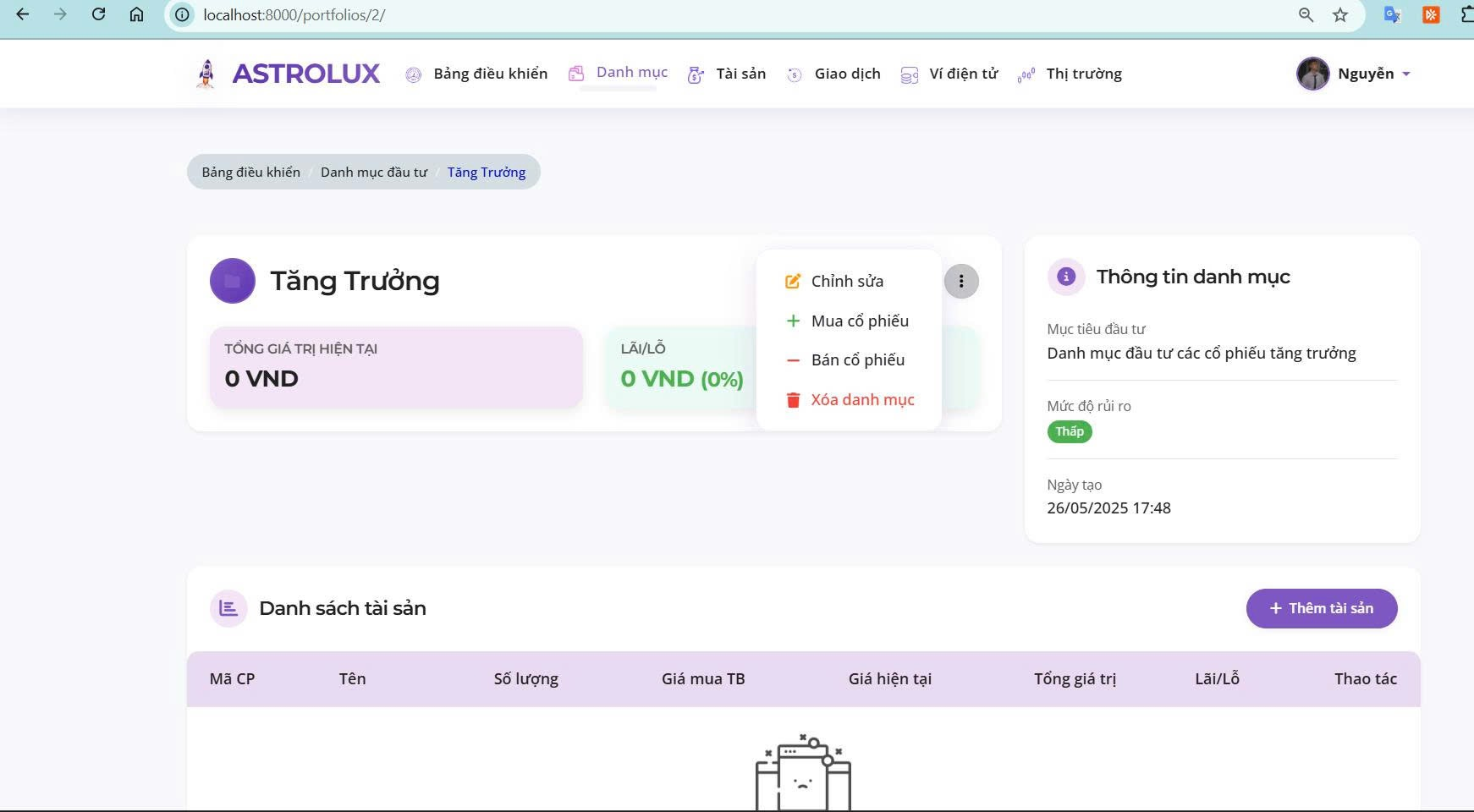
Sidebar bên trái chứa danh sách tất cả danh mục đã tạo, mỗi item hiển thị tên danh mục, tổng giá trị và biểu tượng trạng thái. Nút "Tạo danh mục mới" được đặt ở đầu danh sách với thiết kế nổi bật để khuyến khích hành động. Khi người dùng chọn một danh mục, nội dung chi tiết sẽ được tải vào khu vực chính.

Khu vực chi tiết bao gồm header với thông tin cơ bản của danh mục như tên, mô tả và mục tiêu đầu tư. Bên dưới là bảng chi tiết các cổ phiếu trong danh mục, hiển thị mã cổ phiếu, số lượng nắm giữ, giá mua trung bình, giá thị trường hiện tại, tổng giá trị và tỷ lệ lãi/lỗ. Mỗi dòng trong bảng được tô màu khác nhau để phân biệt giữa các cổ phiếu có lãi và lỗ.

Thanh công cụ phía trên bảng chứa các nút hành động chính như "Thêm cổ phiếu", "Chỉnh sửa danh mục" và "Xóa danh mục", cùng với các công cụ lọc và sắp xếp để người dùng dễ dàng tìm kiếm thông tin cần thiết.

**Giao Diện Giao Dịch Cổ Phiếu**

Trang giao dịch cần thiết kế đặc biệt cẩn thận vì đây là nơi người dùng thực hiện các quyết định tài chính quan trọng. Giao diện được chia thành hai phần chính: form nhập lệnh và panel thông tin hỗ trợ.



*Hình 2.4: Giao diện giao dịch cổ phiếu*

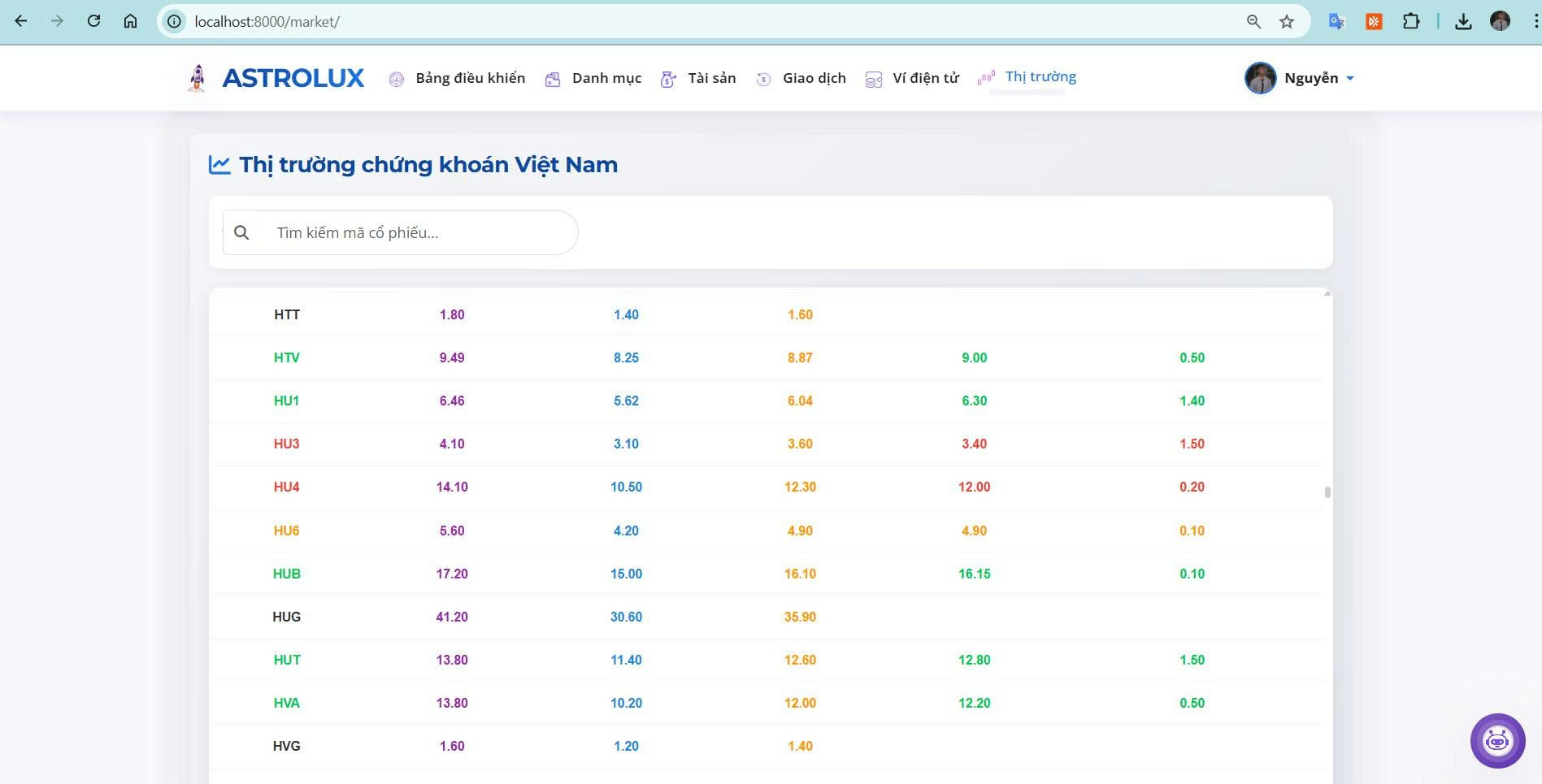
Form giao dịch được thiết kế theo luồng logic từ trên xuống dưới: đầu tiên là lựa chọn loại giao dịch (mua/bán), tiếp theo là chọn danh mục đích, nhập mã cổ phiếu với tính năng autocomplete, xác định số lượng và loại lệnh. Đối với lệnh giới hạn, trường nhập giá sẽ xuất hiện tự động. Hệ thống validation theo thời gian thực sẽ kiểm tra tính hợp lệ của từng trường và hiển thị cảnh báo khi cần thiết.

Panel thông tin bên cạnh hiển thị số dư ví khả dụng (đối với lệnh mua) hoặc số lượng cổ phiếu có thể bán (đối với lệnh bán), giá thị trường hiện tại của cổ phiếu được chọn và ước tính tổng giá trị giao dịch. Những thông tin này được cập nhật tự động khi người dùng thay đổi các tham số giao dịch.

Nút "Xác nhận giao dịch" được thiết kế nổi bật với màu sắc phù hợp (xanh lá cho mua, đỏ cho bán) và chỉ kích hoạt khi tất cả thông tin bắt buộc đã được nhập đúng. Trước khi thực hiện giao dịch, hệ thống sẽ hiển thị dialog xác nhận với tóm tắt chi tiết về giao dịch sắp thực hiện.

**Hệ Thống Theo Dõi Thị Trường**

Giao diện theo dõi thị trường cần cung cấp thông tin real-time về giá cổ phiếu một cách trực quan và dễ tiếp cận. Layout được thiết kế theo mô hình chia đôi với bảng giá bên trái và biểu đồ chi tiết bên phải.



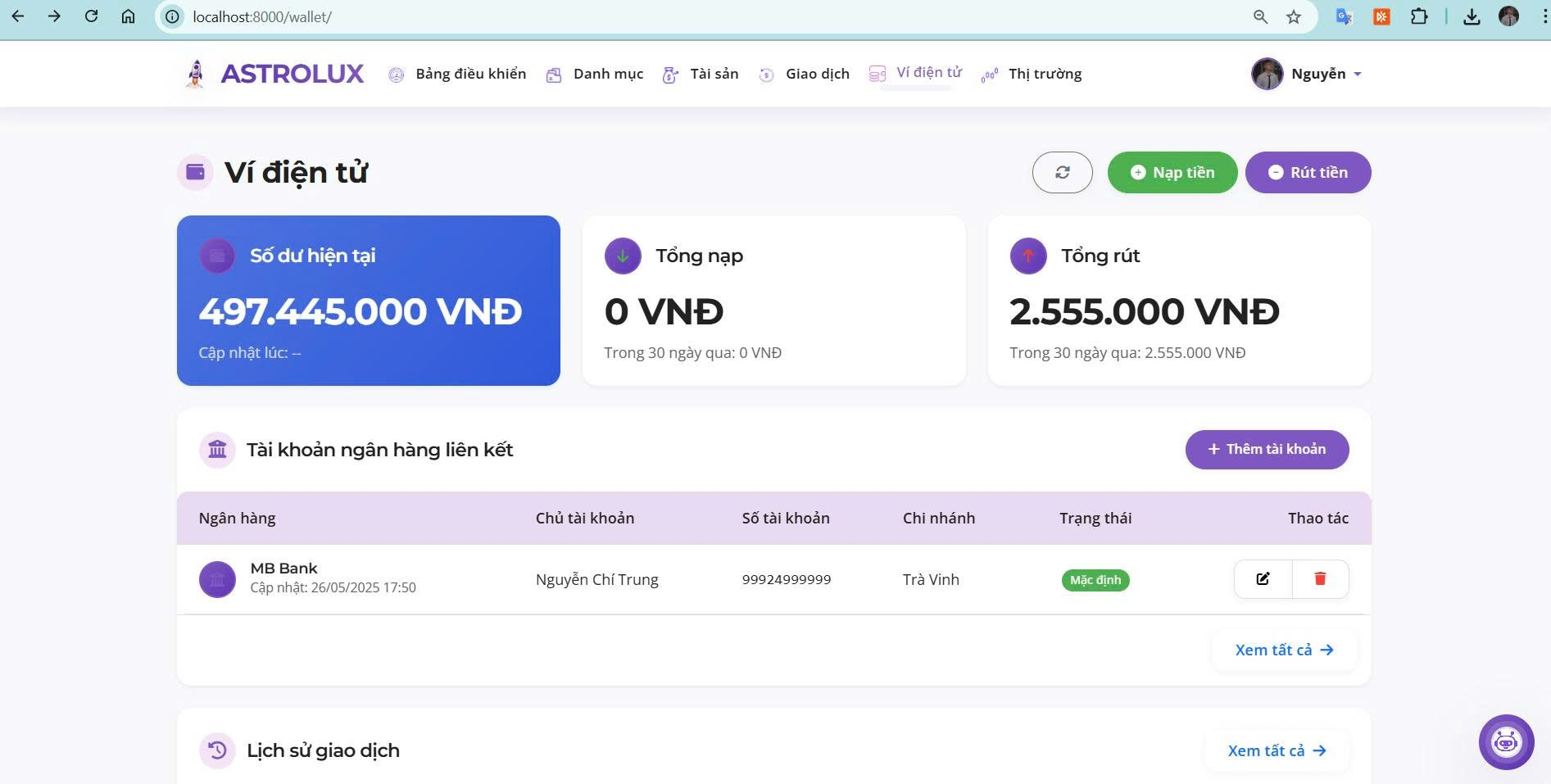
*Hình 2.5: Giao diện theo dõi thị trường*

Bảng giá cổ phiếu hiển thị các thông tin cơ bản như mã, giá hiện tại, thay đổi trong ngày, khối lượng giao dịch và các chỉ số kỹ thuật cơ bản. Màu sắc được sử dụng để phân biệt cổ phiếu tăng giá (xanh lá), giảm giá (đỏ) và đứng giá (vàng). Chức năng tìm kiếm và lọc theo sàn giao dịch được tích hợp ở đầu bảng để người dùng dễ dàng tìm kiếm cổ phiếu quan tâm.

Khi người dùng chọn một mã cổ phiếu, panel bên phải sẽ hiển thị biểu đồ giá lịch sử với khả năng điều chỉnh khung thời gian.

**Hệ Thống Quản Lý Ví Điện Tử**

Giao diện quản lý ví được thiết kế để cung cấp cái nhìn toàn diện về tình hình tài chính và lịch sử giao dịch của người dùng. Layout được tổ chức theo cấu trúc tab để phân chia rõ ràng các chức năng khác nhau.



*Hình 2.6: Giao diện quản lý ví điện tử*

Tab "Tổng quan" hiển thị số dư ví hiện tại với định dạng rõ ràng, kèm theo biểu đồ thống kê dòng tiền vào/ra trong khoảng thời gian gần đây. Các nút chức năng "Nạp tiền" và "Rút tiền" được thiết kế nổi bật và dễ tiếp cận.

Form nạp tiền cung cấp nhiều phương thức thanh toán khác nhau, bao gồm chuyển khoản ngân hàng với thông tin tài khoản được hiển thị rõ ràng và QR code để quét nhanh. Hướng dẫn chi tiết về quy trình nạp tiền được trình bày theo từng bước một cách trực quan.

Form rút tiền yêu cầu người dùng chọn tài khoản ngân hàng đã liên kết từ danh sách dropdown và nhập số tiền muốn rút. Hệ thống sẽ kiểm tra số dư khả dụng và hiển thị phí giao dịch (nếu có) trước khi cho phép xác nhận.

Tab "Lịch sử giao dịch" hiển thị tất cả các giao dịch nạp/rút tiền theo thứ tự thời gian, bao gồm thông tin về số tiền, phương thức, trạng thái và thời gian thực hiện. Chức năng lọc theo thời gian và loại giao dịch giúp người dùng dễ dàng tìm kiếm thông tin cần thiết.

Tổng thể, thiết kế giao diện của hệ thống tuân thủ các nguyên tắc thiết kế hiện đại, tập trung vào trải nghiệm người dùng và đảm bảo tính nhất quán trên toàn bộ ứng dụng. Việc sử dụng responsive design và các patterns phổ biến sẽ giúp người dùng dễ dàng làm quen và sử dụng hiệu quả các chức năng của hệ thống.

1. **Hiện Thực Hệ Thống**

Giai đoạn hiện thực hóa hệ thống là quá trình chuyển đổi các thiết kế lý thuyết thành một sản phẩm phần mềm hoạt động thực tế. Quá trình này đòi hỏi sự kết hợp chặt chẽ giữa kiến thức lý thuyết và kỹ năng thực hành, đồng thời cần tuân thủ các tiêu chuẩn phát triển phần mềm hiện đại.

**4.1 Môi trường và công cụ phát triển**

Việc lựa chọn môi trường và công cụ phát triển phù hợp đóng vai trò quyết định đến hiệu quả và chất lượng của quá trình hiện thực. Hệ thống được phát triển dựa trên nền tảng công nghệ đa dạng, mỗi thành phần được lựa chọn cẩn thận để đáp ứng các yêu cầu cụ thể.

Đối với phần back-end, Python được chọn làm ngôn ngữ lập trình chính với framework Django. Quyết định này dựa trên tính ổn định, bảo mật cao, và hệ sinh thái thư viện phong phú của Python. Django cung cấp các tính năng sẵn có như ORM (Object-Relational Mapping), hệ thống xác thực, và admin interface, giúp tăng tốc quá trình phát triển và đảm bảo tính nhất quán của mã nguồn.

Phần front-end được xây dựng bằng HTML5, CSS3, và JavaScript ES6+ kết hợp với framework Bootstrap. Sự kết hợp này đảm bảo giao diện người dùng responsive, tương thích đa nền tảng và thiết bị. Bootstrap cung cấp hệ thống grid linh hoạt và các component sẵn có, giúp rút ngắn thời gian phát triển giao diện đồng thời duy trì tính nhất quán về thiết kế.

Hệ quản trị cơ sở dữ liệu PostgreSQL được sử dụng nhờ khả năng xử lý giao dịch ACID đáng tin cậy, hỗ trợ các tính năng nâng cao như JSON fields, full-text search, và khả năng mở rộng tốt. Điều này đặc biệt quan trọng đối với một hệ thống tài chính đòi hỏi tính toàn vẹn dữ liệu cao.

Các thư viện Python chuyên biệt được tích hợp bao gồm vnstock cho việc thu thập dữ liệu thị trường chứng khoán Việt Nam, requests cho các API call, django-crispy-forms và crispy-bootstrap5 để tối ưu hóa giao diện form, và authlib để hỗ trợ tích hợp Auth0 nếu cần thiết.

**4.2 Cấu trúc thư mục dự án**

Cấu trúc thư mục dự án được tổ chức theo chuẩn Django, đảm bảo tính rõ ràng, dễ bảo trì và khả năng mở rộng. Kiến trúc này tuân thủ nguyên tắc separation of concerns, phân chia rõ ràng các thành phần theo chức năng và trách nhiệme

*Hình 4: Sơ đồ cấu trúc thư mục dự án Django*

Thư mục gốc của dự án chứa các file cấu hình quan trọng như manage.py (file điều khiển chính của Django), requirements.txt (danh sách các dependencies), .env (file cấu hình biến môi trường), và docker-compose.yml (cấu hình container hóa). Cấu trúc này đảm bảo việc triển khai và bảo trì dự án được thực hiện một cách nhất quán và đáng tin cậy.

Thư mục config chứa các file cấu hình Django bao gồm settings.py (cài đặt chính), urls.py (định tuyến URL cấp cao), wsgi.py và asgi.py (cấu hình web server). Việc tách biệt cấu hình thành module riêng giúp quản lý các môi trường khác nhau (development, staging, production) một cách hiệu quả.

**4.3 Ứng Dụng Portfolio - Thành Phần Trung Tâm**

Ứng dụng portfolio đóng vai trò hạt nhân của hệ thống, chứa đựng toàn bộ logic nghiệp vụ liên quan đến quản lý danh mục đầu tư. Cấu trúc của ứng dụng này được thiết kế theo mô hình MVC (Model-View-Controller) của Django.

**Models - Tầng Dữ Liệu**

File models.py định nghĩa cấu trúc cơ sở dữ liệu thông qua các lớp Python, mỗi lớp tương ứng với một bảng trong cơ sở dữ liệu. Các model này không chỉ định nghĩa cấu trúc dữ liệu mà còn bao gồm các phương thức nghiệp vụ và validation logic.

Model User kế thừa từ AbstractUser của Django, mở rộng thêm các trường cần thiết như phone, address, gender, và profile\_picture. Thiết kế này cho phép tích hợp seamless với hệ thống xác thực có sẵn của Django đồng thời mở rộng được theo nhu cầu cụ thể.

Model Wallet được thiết kế với mối quan hệ OneToOneField với User, đảm bảo mỗi người dùng chỉ có một ví duy nhất. Các phương thức như deposit(), withdraw(), và get\_balance() được định nghĩa để xử lý logic nghiệp vụ liên quan đến giao dịch ví.

Model Portfolio và PortfolioSymbol thiết lập mối quan hệ phức tạp để quản lý danh mục đầu tư. Portfolio chứa thông tin tổng quan về danh mục, trong khi PortfolioSymbol chi tiết từng cổ phiếu trong danh mục. Các phương thức như calculate\_total\_value(), get\_portfolio\_performance(), và update\_portfolio\_statistics() được triển khai để tính toán các chỉ số tài chính quan trọng.

**Views - Tầng Xử Lý Logic**

File views.py chứa các hàm xử lý request HTTP, đóng vai trò trung gian giữa model và template. Các view được tổ chức theo nhóm chức năng và tuân thủ nguyên tắc single responsibility.

Nhóm views xử lý xác thực người dùng bao gồm register\_view, login\_view, logout\_view, và profile\_view. Các view này tích hợp với Django's authentication system và xử lý các case đặc biệt như tích hợp Auth0.

Nhóm views quản lý danh mục gồm portfolio\_list\_view, portfolio\_detail\_view, create\_portfolio\_view, edit\_portfolio\_view, và delete\_portfolio\_view. Mỗi view được thiết kế để xử lý một chức năng cụ thể, sử dụng các form validation và error handling phù hợp.

Views giao dịch cổ phiếu như buy\_stock\_view và sell\_stock\_view sử dụng database transaction để đảm bảo tính ACID. Việc sử dụng @transaction.atomic decorator đảm bảo các thao tác phức tạp được thực hiện an toàn.

**Templates và Static Files**

Hệ thống template được xây dựng dựa trên template inheritance của Django, với base.html làm template gốc chứa layout chung. Các template con như dashboard.html, portfolio\_list.html, portfolio\_detail.html kế thừa và mở rộng template gốc.

Các file static bao gồm CSS, JavaScript, và hình ảnh được tổ chức trong thư mục static, tuân thủ cấu trúc chuẩn của Django. File CSS chính được viết bằng SCSS và compile thành CSS, sử dụng các CSS custom properties và flexbox để tạo layout responsive.

JavaScript được tổ chức theo module, sử dụng ES6+ features và async/await pattern. Các module như portfolio.js, transaction.js, và chart.js xử lý tương tác phía client và giao tiếp với API backend thông qua AJAX.

**4.4 Các Module Hỗ Trợ Chuyên Biệt**

### VNStock Services Module

File vnstock\_services.py đóng gói các hàm tương tác với thư viện vnstock, cung cấp interface thống nhất để thu thập dữ liệu thị trường. Module này bao gồm các hàn như get\_stock\_price(), get\_stock\_info(), sync\_market\_data(), và update\_portfolio\_prices().

Hàm sync\_vnstock\_to\_assets() thực hiện đồng bộ hóa dữ liệu từ vnstock vào bảng Assets, sử dụng bulk\_update() và bulk\_create() của Django ORM để tối ưu hiệu suất. Error handling được implement để xử lý các trường hợp API không phản hồi hoặc dữ liệu không hợp lệ.

**Utils Module**

File utils.py tập hợp các hàm tiện ích được sử dụng xuyên suốt hệ thống. Các hàm quan trọng bao gồm:

Hàm generate\_qr\_code() sử dụng thư viện qrcode để tạo mã QR cho giao dịch chuyển khoản, tích hợp thông tin tài khoản ngân hàng và số tiền giao dịch.

Hàm check\_paid() tương tác với Google Script API để xác minh giao dịch ngân hàng. Hàm này gửi request POST đến endpoint API với thông tin giao dịch và nhận response xác nhận trạng thái thanh toán.

Hàm get\_ai\_response() gọi Gemini API để cung cấp tính năng chatbot tư vấn đầu tư. Hàm này xử lý prompt engineering, rate limiting, và error handling cho các tương tác AI.

**4.5. Hiện Thực Các Luồng Nghiệp Vụ Chính**

### Luồng Giao Dịch Mua Bán Cổ Phiếu

Luồng mua cổ phiếu bắt đầu từ khi người dùng submit form tại giao diện. Request được xử lý bởi buy\_stock\_view, thực hiện các bước validation đầu vào, kiểm tra số dư ví, lấy giá cổ phiếu realtime từ vnstock, và tính toán tổng giá trị giao dịch.

Sau khi validation thành công, hệ thống sử dụng database transaction để thực hiện một loạt thao tác atomic: cập nhật số dư ví, tạo record StockTransaction, cập nhật hoặc tạo mới PortfolioSymbol, và recalculate các chỉ số danh mục. Nếu bất kỳ bước nào thất bại, toàn bộ transaction được rollback.

Luồng bán cổ phiếu tương tự nhưng có thêm kiểm tra số lượng cổ phiếu đang sở hữu. Hệ thống sử dụng FIFO (First In, First Out) để tính toán lãi/lỗ khi bán cổ phiếu.

### Luồng Nạp Tiền Qua Ngân Hàng

Quá trình nạp tiền được thiết kế theo mô hình asynchronous để xử lý việc xác nhận giao dịch ngân hàng. Khi người dùng yêu cầu nạp tiền, hệ thống tạo một BankTransaction với status 'pending' và sinh mã QR chứa thông tin chuyển khoản.

Google Script API được sử dụng để polling giao dịch ngân hàng theo intervalic. Khi phát hiện giao dịch matching, webhook được trigger để cập nhật trạng thái BankTransaction thành 'completed' và cộng tiền vào ví người dùng.

Cơ chế retry và timeout được implement để xử lý các trường hợp API không phản hồi hoặc giao dịch bị delay.

### Luồng Cập Nhật Dữ Liệu Thị Trường

Hệ thống sử dụng Celery periodic task để cập nhật dữ liệu thị trường theo định kỳ. Task sync\_market\_data\_task được scheduled chạy mỗi 5 phút trong giờ giao dịch, gọi đến vnstock API để lấy dữ liệu mới nhất.

Dữ liệu được validate và normalize trước khi update vào database. Cơ chế caching được áp dụng để giảm số lượng API call và tăng hiệu suất hệ thống.

### Luồng Tương Tác AI Chatbot

Tính năng chatbot được implement thông qua WebSocket connection để đảm bảo tương tác realtime. Khi người dùng gửi tin nhắn, hệ thống thực hiện preprocessing để extract context và intent, sau đó gọi Gemini API với prompt được optimize cho domain tài chính.

Response từ AI được post-process để format phù hợp và filter các thông tin có thể gây hiểu lầm về tư vấn đầu tư. Rate limiting được áp dụng để tránh abuse và quản lý cost API.

## Kiểm Thử

Công tác kiểm thử đóng một vai trò thiết yếu và không thể tách rời trong quy trình phát triển phần mềm, đặc biệt đối với một hệ thống đòi hỏi tính chính xác và ổn định cao như "Hệ Thống Quản Lý Danh Mục Đầu Tư". Giai đoạn này được tiến hành nhằm mục tiêu cao nhất là đảm bảo chất lượng sản phẩm, xác minh tính đúng đắn của các chức năng so với yêu cầu đã đặt ra, đồng thời phát hiện và khắc phục các lỗi tiềm ẩn trước khi hệ thống được chính thức đưa vào vận hành.

### 5.1. Kế hoạch kiểm thử

Để đạt được các mục tiêu trên, một kế hoạch kiểm thử toàn diện đã được xây dựng theo phương pháp tiếp cận nhiều tầng, từ mức độ chi tiết nhỏ nhất đến tổng thể. Quá trình này bắt đầu từ việc **kiểm thử đơn vị (Unit Test)**, nơi từng thành phần mã nguồn nhỏ nhất như các hàm xử lý logic trong views.py, các phương thức trong models.py, hay các hàm tiện ích trong utils.py được kiểm tra một cách độc lập. Lập trình viên đã sử dụng framework kiểm thử tích hợp của Django để viết các kịch bản kiểm thử, đảm bảo mỗi đơn vị hoạt động chính xác với các loại dữ liệu đầu vào khác nhau, bao gồm cả các trường hợp biên và ngoại lệ.

Tiếp nối kiểm thử đơn vị là **kiểm thử tích hợp (Integration Test)**, tập trung vào việc xác minh sự tương tác trơn tru và đúng đắn giữa các module khi được kết nối với nhau. Chẳng hạn, một kịch bản kiểm thử tích hợp điển hình là kiểm tra luồng dữ liệu từ khi người dùng thực hiện giao dịch mua cổ phiếu. Trong trường hợp này, hệ thống phải cập nhật chính xác số dư trong Wallet, ghi nhận giao dịch vào StockTransaction, và điều chỉnh số lượng cổ phiếu trong Asset thuộc Portfolio tương ứng. Đặc biệt, sự phối hợp giữa hệ thống với các API bên ngoài như vnstock và Google Script API mô phỏng cũng được kiểm tra kỹ lưỡng để đảm bảo tính ổn định của các tích hợp này.

Ở cấp độ cao hơn, **kiểm thử hệ thống (System Test)** được thực hiện trên một môi trường đã được tích hợp hoàn chỉnh, mô phỏng gần như hoàn toàn môi trường vận hành thực tế. Giai đoạn này tập trung vào việc đánh giá toàn bộ ứng dụng thông qua các kịch bản sử dụng từ đầu đến cuối (end-to-end scenarios), đảm bảo hệ thống không chỉ đáp ứng các yêu cầu chức năng mà còn tuân thủ các yêu cầu phi chức năng như hiệu năng, khả năng tương thích trình duyệt và tính dễ sử dụng.

Cuối cùng, **kiểm thử chấp nhận người dùng (User Acceptance Test - UAT)** được thực hiện bởi các thành viên nhóm đóng vai người dùng cuối, nhằm xác nhận rằng hệ thống thực sự đáp ứng được mong đợi và nhu cầu của người dùng mục tiêu trong các tình huống sử dụng thực tế.

### 5.2. Mô tả các kịch bản kiểm thử quan trọng

Trong quá trình kiểm thử, nhiều kịch bản (test case) đã được xây dựng và thực thi để bao phủ các chức năng quan trọng của hệ thống. Đối với chức năng **đăng ký tài khoản người dùng**, một kịch bản kiểm thử toàn diện bao gồm việc người dùng truy cập trang đăng ký, nhập đầy đủ thông tin hợp lệ với các điều kiện như tên đăng nhập và email chưa tồn tại trong hệ thống, mật khẩu đáp ứng các tiêu chí bảo mật, và mật khẩu xác nhận khớp với mật khẩu gốc. Kết quả mong đợi là hệ thống sẽ thông báo đăng ký thành công, chuyển hướng người dùng đến trang đăng nhập, đồng thời tự động tạo một bản ghi người dùng mới cùng với một ví điện tử liên kết trong cơ sở dữ liệu.

Một kịch bản quan trọng khác là **thực hiện giao dịch mua cổ phiếu thành công**. Kịch bản này yêu cầu người dùng đã đăng nhập với ví có đủ số dư, sau đó điều hướng đến chức năng mua cổ phiếu và lựa chọn danh mục đầu tư phù hợp. Người dùng sẽ nhập mã cổ phiếu hợp lệ, số lượng và giá mong muốn. Khi xác nhận giao dịch, hệ thống được kỳ vọng sẽ ghi nhận chính xác giao dịch vào lịch sử, trừ số dư ví tương ứng với tổng giá trị giao dịch, và cập nhật đúng số lượng cũng như giá mua trung bình của cổ phiếu đó trong danh mục được chỉ định.

Đối với các tình huống ngoại lệ, các kịch bản như **mua cổ phiếu khi không đủ số dư** được thiết kế để đảm bảo hệ thống xử lý lỗi một cách duyên dáng. Trong trường hợp này, hệ thống phải hiển thị thông báo lỗi rõ ràng cho người dùng và ngăn chặn việc thực hiện giao dịch không hợp lệ mà không ảnh hưởng đến các dữ liệu hiện có.

Chức năng **nạp tiền vào ví** cũng được kiểm thử một cách tỉ mỉ thông qua việc mô phỏng quy trình chuyển khoản. Người dùng thực hiện các thao tác cập nhật thông tin vào Google Sheet mà Google Script API theo dõi, sau đó hệ thống phải tự động nhận diện và xử lý giao dịch này. Kết quả mong đợi là giao dịch được chuyển sang trạng thái "Hoàn thành" và số dư ví của người dùng được tăng lên chính xác theo số tiền đã nạp. Các trường hợp ngoại lệ như mã giao dịch không tồn tại, thông tin không khớp, hoặc giao dịch trùng lặp cũng được kiểm tra để đảm bảo tính toàn vẹn của hệ thống.

*Ghi chú: Bảng tóm tắt các kịch bản kiểm thử quan trọng có thể được trình bày dưới dạng Hình 5.1 để minh họa rõ hơn.*

### 5.3. Kết quả kiểm thử và quy trình khắc phục lỗi

Quá trình kiểm thử theo kế hoạch đã phát hiện một số vấn đề ở các mức độ khác nhau, chủ yếu tập trung vào ba nhóm chính. Nhóm thứ nhất liên quan đến logic xử lý dữ liệu trong các trường hợp biên, ví dụ như việc xử lý số tiền âm hoặc các giá trị đặc biệt trong tính toán giá trung bình của danh mục. Nhóm thứ hai bao gồm các vấn đề về giao diện người dùng, đặc biệt trên một số kích thước màn hình cụ thể hoặc các trình duyệt khác nhau. Nhóm thứ ba liên quan đến hiệu năng, đặc biệt là thời gian phản hồi chưa được tối ưu khi truy xuất và xử lý lượng dữ liệu lớn.

Tất cả các lỗi được phát hiện đều được ghi nhận một cách có hệ thống vào danh sách quản lý lỗi, kèm theo thông tin chi tiết bao gồm mô tả các bước tái hiện, kết quả thực tế thu được, kết quả mong đợi, và mức độ nghiêm trọng. Sau khi được phân tích và xác định nguyên nhân gốc rễ, các lỗi được phân loại theo mức độ ưu tiên để đội ngũ phát triển có thể tập trung giải quyết các vấn đề quan trọng nhất trước.

Quy trình khắc phục lỗi bao gồm việc điều chỉnh mã nguồn, cập nhật cấu hình hệ thống, hoặc tối ưu hóa các truy vấn cơ sở dữ liệu tùy theo tính chất của từng lỗi. Đặc biệt quan trọng, sau mỗi đợt sửa lỗi, **kiểm thử hồi quy (Regression Testing)** được thực hiện một cách nghiêm ngặt trên các chức năng liên quan và các khu vực có khả năng bị ảnh hưởng, nhằm đảm bảo rằng việc khắc phục không vô tình tạo ra các vấn đề mới.

Nhờ quy trình kiểm thử và khắc phục lỗi được tiến hành một cách có hệ thống này, chất lượng tổng thể của hệ thống đã được cải thiện đáng kể. Các chức năng chính hoạt động ổn định hơn, các lỗi nghiêm trọng đã được loại bỏ hoàn toàn, và trải nghiệm người dùng cũng được nâng cao rõ rệt. Kết quả kiểm thử cuối cùng cho thấy hệ thống đã đáp ứng phần lớn các yêu cầu đặt ra ban đầu và sẵn sàng cho giai đoạn triển khai.

**6. Triển Khai và Hướng Dẫn Sử Dụng**

Sau khi hệ thống đã trải qua các giai đoạn phát triển và kiểm thử kỹ lưỡng, việc triển khai ứng dụng để người dùng cuối có thể truy cập và sử dụng, cùng với việc cung cấp tài liệu hướng dẫn rõ ràng và chi tiết, trở thành những bước tiếp theo mang tính quyết định đến sự thành công của toàn bộ dự án.

### 6.1. Yêu cầu hệ thống và môi trường triển khai

Để "Hệ Thống Quản Lý Danh Mục Đầu Tư" có thể hoạt động ổn định và hiệu quả trong môi trường production, một số yêu cầu cụ thể về hạ tầng và phần mềm cần được đáp ứng một cách đầy đủ. Về phía hệ điều hành, khuyến nghị sử dụng một bản phân phối Linux ổn định và được hỗ trợ lâu dài như Ubuntu Server LTS hoặc CentOS. Cấu hình phần cứng tối thiểu cần được tính toán dựa trên số lượng người dùng dự kiến và khối lượng dữ liệu giao dịch, với khả năng mở rộng linh hoạt khi cần thiết. Cụ thể, máy chủ cần có ít nhất 2 GB RAM cho môi trường nhỏ, nhưng nên có 4-8 GB RAM cho môi trường production thực tế, cùng với CPU đa nhân và dung lượng ổ cứng đủ lớn để lưu trữ cơ sở dữ liệu và các tệp log.

Các thành phần phần mềm thiết yếu bao gồm Python phiên bản 3.8 trở lên để đảm bảo tương thích với Django và các thư viện được sử dụng, hệ quản trị cơ sở dữ liệu PostgreSQL phiên bản 12 trở lên để tận dụng các tính năng hiện đại, một máy chủ ứng dụng WSGI hiệu năng cao như Gunicorn hoặc uWSGI, và một máy chủ web ngược (reverse proxy) như Nginx để xử lý các yêu cầu HTTP, phục vụ tệp tĩnh, và cân bằng tải nếu cần thiết. Ngoài ra, việc đăng ký một tên miền và cài đặt chứng chỉ SSL/TLS (khuyến nghị sử dụng Let's Encrypt miễn phí) để kích hoạt giao thức HTTPS là yêu cầu bắt buộc nhằm đảm bảo an toàn cho dữ liệu truyền tải giữa người dùng và máy chủ.

### 6.2. Quy trình cài đặt chi tiết

Quy trình cài đặt hệ thống được chia thành hai phạm vi chính: môi trường phát triển cục bộ và môi trường production. Đối với **môi trường phát triển**, người dùng cần chuẩn bị sẵn các công cụ cơ bản bao gồm Git để quản lý mã nguồn, Python 3.8+, pip để quản lý gói Python, và PostgreSQL (hoặc có thể sử dụng SQLite để đơn giản hóa trong giai đoạn đầu).

Bước đầu tiên là clone mã nguồn của dự án từ repository thông qua lệnh git clone [repository-url]. Tiếp theo, việc tạo môi trường ảo Python bằng lệnh python -m venv venv và kích hoạt nó bằng source venv/bin/activate (trên Linux/Mac) hoặc venv\Scripts\activate (trên Windows) là cực kỳ quan trọng để cô lập các gói phụ thuộc. Sau đó, tất cả các gói phụ thuộc sẽ được cài đặt thông qua lệnh pip install -r requirements.txt.

Cấu hình cơ sở dữ liệu trong tệp settings.py cần được điều chỉnh cho phù hợp với thông số của PostgreSQL cục bộ, bao gồm tên cơ sở dữ liệu, tên người dùng, mật khẩu và host. Sau khi cấu hình xong, các lệnh python manage.py makemigrations và python manage.py migrate cần được chạy để khởi tạo cấu trúc bảng trong cơ sở dữ liệu theo các model đã định nghĩa. Việc tạo một tài khoản superuser thông qua lệnh python manage.py createsuperuser cho phép truy cập vào trang quản trị Django. Cuối cùng, máy chủ phát triển có thể được khởi chạy bằng lệnh python manage.py runserver, cho phép truy cập ứng dụng thông qua địa chỉ http://localhost:8000.

Đối với **môi trường production**, quy trình phức tạp hơn đáng kể và yêu cầu sự cẩn thận cao hơn. Ngoài các bước cơ bản tương tự như môi trường phát triển, việc triển khai production còn bao gồm cấu hình Gunicorn để phục vụ ứng dụng Django với các tham số tối ưu như số worker processes, thiết lập Nginx làm reverse proxy với các cấu hình cache và compression, cấu hình tên miền và DNS, cài đặt chứng chỉ SSL/TLS, và thiết lập các biện pháp bảo mật như firewall, fail2ban, và các quy tắc bảo mật cho PostgreSQL. Quá trình này thường được tự động hóa thông qua các script deployment hoặc sử dụng các công cụ như Docker để đảm bảo tính nhất quán và giảm thiểu lỗi con người.

### 6.3. Hướng dẫn sử dụng toàn diện các chức năng hệ thống

Để người dùng có thể nhanh chóng làm quen và khai thác hiệu quả tối đa các tính năng của "Hệ Thống Quản Lý Danh Mục Đầu Tư", một tài liệu hướng dẫn sử dụng chi tiết và có cấu trúc rõ ràng là điều không thể thiếu. Hành trình của người dùng thường bắt đầu với việc **đăng ký tài khoản mới** thông qua trang đăng ký, nơi họ cần cung cấp các thông tin cơ bản như tên đăng nhập độc nhất, địa chỉ email hợp lệ, và mật khẩu mạnh đạt yêu cầu bảo mật. Sau khi hoàn tất đăng ký, người dùng có thể **đăng nhập** vào hệ thống bằng thông tin tài khoản vừa tạo.

Khi đã truy cập thành công vào **bảng điều khiển chính (Dashboard)**, người dùng sẽ được chào đón bởi một giao diện tổng quan trực quan, hiển thị các thông tin quan trọng nhất bao gồm tổng giá trị tài sản hiện tại, số dư ví khả dụng, số lượng danh mục đầu tư đang quản lý, và biểu đồ thể hiện xu hướng tăng trưởng của danh mục theo thời gian. Dashboard này được thiết kế để cung cấp cái nhìn tổng quan nhanh chóng và giúp người dùng có thể đưa ra quyết định kịp thời.

Chức năng **quản lý ví điện tử** đóng vai trò trung tâm trong hệ thống, cho phép người dùng theo dõi tất cả các giao dịch tài chính của mình. Trong phần này, người dùng có thể xem lịch sử chi tiết các giao dịch nạp tiền, rút tiền, mua bán cổ phiếu, cùng với thông tin về thời gian, số tiền và trạng thái của từng giao dịch. Tính năng **nạp tiền** được thiết kế với quy trình thuận tiện: hệ thống sẽ cung cấp thông tin tài khoản ngân hàng hoặc mã QR để người dùng thực hiện chuyển khoản, sau đó giao dịch được xác nhận tự động thông qua cơ chế tích hợp với Google Script API. Tương tự, chức năng **rút tiền** cho phép người dùng chuyển số dư từ ví về tài khoản ngân hàng đã liên kết, với các bước xác thực cần thiết để đảm bảo bảo mật.*.*

Việc **tạo và quản lý danh mục đầu tư** là một trong những tính năng cốt lõi và đầy sức mạnh của hệ thống. Người dùng có thể tạo ra nhiều danh mục khác nhau tùy theo chiến lược đầu tư của mình, mỗi danh mục có thể được đặt tên riêng biệt, mô tả chi tiết về mục tiêu đầu tư, xác định mức độ rủi ro chấp nhận được, và thậm chí thiết lập các ngưỡng cảnh báo tự động. Khi xem chi tiết một danh mục cụ thể, hệ thống sẽ hiển thị danh sách đầy đủ các cổ phiếu đang nắm giữ, bao gồm thông tin về số lượng, giá mua trung bình, giá thị trường hiện tại, tỷ trọng trong danh mục, và mức lãi/lỗ cả về số tiền tuyệt đối lẫn tỷ lệ phần trăm. Các biểu đồ trực quan cũng được cung cấp để người dùng có thể nhanh chóng đánh giá hiệu suất và phân bổ tài sản trong danh mục.

Để **thực hiện giao dịch cổ phiếu**, người dùng truy cập vào module giao dịch và lựa chọn giữa hai loại giao dịch chính: mua hoặc bán. Đối với giao dịch mua, người dùng cần chỉ định mã cổ phiếu (hệ thống hỗ trợ tìm kiếm thông minh và gợi ý), số lượng cổ phiếu muốn mua, loại lệnh (thị trường hoặc giới hạn), giá mong muốn (nếu là lệnh giới hạn), và danh mục đầu tư sẽ chứa cổ phiếu này. Trước khi xác nhận, hệ thống sẽ kiểm tra tự động các điều kiện cần thiết như số dư ví có đủ để thực hiện giao dịch hay không, và hiển thị tóm tắt giao dịch để người dùng xem xét lần cuối. Đối với giao dịch bán, quy trình tương tự nhưng hệ thống sẽ kiểm tra số lượng cổ phiếu khả dụng trong danh mục được chọn.

Cuối cùng, chức năng **theo dõi thị trường** cung cấp cái nhìn toàn diện về tình hình thị trường chứng khoán Việt Nam. Người dùng có thể xem bảng giá cập nhật theo thời gian thực của các cổ phiếu quan tâm, sử dụng công cụ tìm kiếm mạnh mẽ để tra cứu thông tin chi tiết về từng mã cổ phiếu bao gồm thông tin công ty, các chỉ số tài chính cơ bản, và biểu đồ lịch sử giá với khả năng điều chỉnh khung thời gian. Điều này giúp người dùng nắm bắt kịp thời các diễn biến thị trường và đưa ra những quyết định đầu tư có căn cứ. Bên cạnh đó, các chức năng quản lý thông tin cá nhân như cập nhật hồ sơ, thay đổi mật khẩu, và cài đặt các tùy chọn thông báo cũng được thiết kế một cách trực quan và dễ sử dụng.

### 7. ĐÁNH GIÁ VÀ HƯỚNG PHÁT TRIỂN

Sau một quá trình nỗ lực từ khâu lên ý tưởng, phân tích, thiết kế, hiện thực đến kiểm thử, việc nhìn lại toàn bộ dự án **"Hệ Thống Quản Lý Danh Mục Đầu Tư"** để đưa ra những đánh giá khách quan về những gì đã đạt được, những ưu điểm nổi bật, cũng như thẳng thắn nhìn nhận những hạn chế còn tồn tại là vô cùng cần thiết. Từ đó, những định hướng phát triển và hoàn thiện hệ thống trong tương lai mới có thể được vạch ra một cách rõ ràng và khả thi.

#### 7.1. Đánh Giá Những Điểm Đã Đạt Được Và Ưu Điểm Của Hệ Thống

Nhìn chung, dự án đã thành công trong việc xây dựng một nền tảng ứng dụng web quản lý danh mục đầu tư với bộ chức năng cốt lõi, đáp ứng được phần lớn các mục tiêu ban đầu đề ra. Hệ thống cho phép người dùng thực hiện các thao tác thiết yếu như: quản lý tài khoản cá nhân, quản lý dòng tiền thông qua ví điện tử với cơ chế nạp tiền mô phỏng, và tạo lập cũng như theo dõi chi tiết nhiều danh mục đầu tư đa dạng.

Đặc biệt, khả năng thực hiện giao dịch mua bán cổ phiếu và cập nhật thông tin thị trường chứng khoán Việt Nam thông qua việc tích hợp thành công thư viện vnstock là một trong những ưu điểm nổi bật, mang lại giá trị thực tiễn cao cho người dùng.

Một điểm mạnh khác của hệ thống là việc lựa chọn kiến trúc công nghệ phù hợp. Nền tảng Django kết hợp với cơ sở dữ liệu PostgreSQL không chỉ cung cấp một môi trường phát triển mạnh mẽ, an toàn mà còn đảm bảo tính module hóa cao, tạo điều kiện thuận lợi cho việc bảo trì và mở rộng các tính năng trong tương lai.

Giao diện người dùng, dù có thể cần cải thiện thêm, đã được thiết kế theo hướng thân thiện và trực quan với ngôn ngữ Tiếng Việt, giúp người dùng trong nước dễ dàng tiếp cận và sử dụng. Hơn nữa, tiềm năng tích hợp với các công nghệ tiên tiến như Gemini API cho trợ lý AI, dù mới ở giai đoạn đầu, cũng mở ra một hướng đi đầy hứa hẹn.

#### 7.2. Phân Tích Những Hạn Chế Còn Tồn Tại

Bên cạnh những thành tựu đáng ghi nhận, hệ thống vẫn còn một số hạn chế cần được nhìn nhận và khắc phục. Thứ nhất, mặc dù cung cấp các chức năng quản lý cơ bản, hệ thống hiện tại còn thiếu các công cụ phân tích danh mục chuyên sâu và báo cáo hiệu suất đầu tư chi tiết. Người dùng có thể mong muốn các biểu đồ so sánh hiệu suất với chỉ số thị trường, các chỉ số đo lường rủi ro, hoặc các công cụ sàng lọc cổ phiếu dựa trên nhiều tiêu chí phức tạp hơn.

Thứ hai, chức năng trợ lý AI, nếu được tích hợp, hiện chỉ dừng ở mức độ cơ bản, chưa khai thác hết tiềm năng của các mô hình ngôn ngữ lớn trong việc cung cấp các tư vấn đầu tư cá nhân hóa, phân tích tâm lý thị trường hay dự báo xu hướng.

Thứ ba, cơ chế xác nhận giao dịch nạp tiền thông qua Google Script API, tuy giải quyết được bài toán mô phỏng, nhưng vẫn mang tính thủ công và chưa phải là một giải pháp tự động hóa hoàn toàn. Nó thiếu sự liền mạch và tin cậy nếu so sánh với việc tích hợp cổng thanh toán thực tế.

Ngoài ra, trải nghiệm người dùng tổng thể, mặc dù đã được chú trọng, vẫn có thể được cải thiện hơn nữa thông qua việc tối ưu hóa các luồng thao tác và thiết kế giao diện tinh tế hơn. Hiệu năng của hệ thống khi phải xử lý một lượng lớn dữ liệu người dùng và giao dịch đồng thời cũng là một yếu tố cần được kiểm tra và tối ưu hóa kỹ lưỡng hơn.

Cuối cùng, việc chưa có một ứng dụng di động chuyên biệt cũng là một rào cản đối với những người dùng mong muốn sự tiện lợi và truy cập thường xuyên trên các thiết bị cầm tay.

#### 7.3. Đề Xuất Các Hướng Phát Triển Và Hoàn Thiện Hệ Thống Trong Tương Lai

Từ những phân tích trên, một số hướng phát triển và hoàn thiện hệ thống có thể được ưu tiên trong tương lai. Trước hết, việc nâng cao và mở rộng các tính năng phân tích, báo cáo là rất cần thiết. Điều này bao gồm việc tích hợp các chỉ báo phân tích kỹ thuật phổ biến (như MACD, RSI, Moving Averages) trực tiếp vào biểu đồ giá, cung cấp các báo cáo hiệu suất danh mục có khả năng tùy chỉnh cao, và phát triển công cụ sàng lọc cổ phiếu mạnh mẽ hơn.

Thứ hai, phát triển trợ lý AI thông minh hơn là một hướng đi đầy tiềm năng. Hệ thống có thể được nâng cấp để cung cấp các khuyến nghị đầu tư được cá nhân hóa dựa trên hồ sơ rủi ro và mục tiêu tài chính của từng người dùng, phân tích tin tức và dữ liệu phi cấu trúc để đưa ra nhận định thị trường, hoặc thậm chí là tự động hóa một số tác vụ quản lý danh mục cơ bản.

Thứ ba, việc tích hợp với các cổng thanh toán thực tế hoặc API ngân hàng (Open Banking) sẽ là một bước tiến quan trọng, giúp tự động hóa hoàn toàn và tăng cường tính bảo mật cho quy trình nạp/rút tiền. Đồng thời, xây dựng một ứng dụng di động (Mobile App) cho cả nền tảng iOS và Android sẽ đáp ứng nhu cầu ngày càng tăng về tính di động và tiện lợi của người dùng.

Bên cạnh đó, các khía cạnh như tăng cường bảo mật (ví dụ, triển khai xác thực hai yếu tố - 2FA, thực hiện kiểm tra thâm nhập định kỳ), tối ưu hóa hiệu năng liên tục, và tiềm năng mở rộng hỗ trợ các loại tài sản đầu tư khác (như trái phiếu, chứng chỉ quỹ, vàng) cũng cần được xem xét để hệ thống ngày càng trở nên toàn diện và mạnh mẽ, thực sự trở thành một người bạn đồng hành đáng tin cậy của nhà đầu tư cá nhân.

**8. KẾT LUẬN**

Dự án **"Hệ Thống Quản Lý Danh Mục Đầu Tư"** được thực hiện với một mục tiêu rõ ràng là cung cấp một giải pháp công nghệ hỗ trợ các nhà đầu tư cá nhân tại Việt Nam trong việc quản lý tài sản, thực hiện các giao dịch tài chính và tiếp cận thông tin thị trường một cách hiệu quả. Qua một chặng đường từ phân tích, thiết kế, đến hiện thực và kiểm thử, dự án đã đạt được những thành tựu quan trọng, đồng thời cũng mở ra những nhận thức sâu sắc về cả quá trình phát triển sản phẩm lẫn những định hướng cho tương lai.

#### 8.1. Tóm Tắt Những Kết Quả Chính Của Dự Án

Kết quả nổi bật nhất của dự án là việc xây dựng thành công một ứng dụng web với đầy đủ các chức năng cốt lõi, cho phép người dùng thực hiện toàn bộ quy trình quản lý đầu tư cá nhân. Cụ thể, hệ thống đã cung cấp khả năng quản lý tài khoản người dùng một cách an toàn; quản lý ví điện tử với các nghiệp vụ nạp, rút tiền và theo dõi lịch sử giao dịch, trong đó chức năng nạp tiền được mô phỏng một cách sáng tạo thông qua Google Script API; tạo lập và theo dõi chi tiết nhiều danh mục đầu tư khác nhau; thực hiện các giao dịch mua bán cổ phiếu một cách trực quan; và cập nhật thông tin thị trường chứng khoán Việt Nam theo thời gian thực nhờ vào việc tích hợp thành công thư viện vnstock. Nền tảng công nghệ vững chắc dựa trên Django Framework và PostgreSQL, cùng với việc áp dụng các nguyên tắc thiết kế module hóa, đã tạo tiền đề cho một hệ thống ổn định, dễ bảo trì và có khả năng mở rộng.

Giao diện người dùng, được thiết kế với ngôn ngữ Tiếng Việt và tập trung vào tính trực quan, đã góp phần làm cho hệ thống trở nên gần gũi và dễ tiếp cận hơn với người dùng trong nước.

#### 8.2. Khẳng Định Giá Trị Của Hệ Thống

Giá trị mà **"Hệ Thống Quản Lý Danh Mục Đầu Tư"** mang lại không chỉ dừng lại ở việc cung cấp một công cụ tiện ích. Quan trọng hơn, nó góp phần giải quyết một nhu cầu thực tế của thị trường, nơi mà các nhà đầu tư cá nhân, đặc biệt là những người mới tham gia, thường thiếu các công cụ hỗ trợ chuyên nghiệp nhưng dễ sử dụng để quản lý tài sản của mình một cách hiệu quả. Bằng việc đơn giản hóa các quy trình phức tạp, tự động hóa việc tính toán và cung cấp thông tin thị trường một cách kịp thời, hệ thống giúp người

**TÀI LIỆU THAM KHẢO**

1. VNStock. (2025). *Hướng dẫn nhanh*. Truy cập tại: https://vnstocks.com/docs/tai-lieu/huong-dan-nhanh
2. Auth0. (2025). *Auth0 - Dịch vụ quản lý danh tính và xác thực*. Truy cập tại: https://auth0.com/
3. Google AI Studio. (2025). *Gemini API Documentation*. Truy cập tại: https://aistudio.google.com/
4. Docker. (2025). *Docker Documentation*. Truy cập tại: https://www.docker.com/
5. Django Project. (2025). *Django Official Documentation*. Truy cập tại: https://docs.djangoproject.com