**TRƯỜNG ĐẠI HỌC KIẾN TRÚC HÀ NỘI**

**KHOA CÔNG NGHỆ THÔNG TIN**

--------------------

A blue logo with a letter k

Description automatically generated

**ĐỒ ÁN TỐT NGHIỆP KỸ SƯ**

**NGÀNH: CÔNG NGHỆ THÔNG TIN**

**MÃ NGÀNH: 7480201**

**KHOÁ: 2020 - 2025**

***ĐỀ TÀI:***

**XÂY DỰNG WEB PHÂN TÍCH THỊ TRƯỜNG CHỨNG KHOÁN SỬ DỤNG REACTJS VÀ NODEJS**

|  |  |
| --- | --- |
| **GIẢNG VIÊN HƯỚNG DẪN:** | **TS. NGUYỄN ĐĂNG KHOA** |
| **SINH VIÊN THỰC HIỆN:** | **ĐỖ VĂN CHUYÊN** |
| **LỚP:** | **2020CN1** |

**HÀ NỘI 10 - 2024**

**TRƯỜNG ĐẠI HỌC KIẾN TRÚC HÀ NỘI**

**KHOA CÔNG NGHỆ THÔNG TIN**

--------------------

A blue logo with a letter k

Description automatically generated

**ĐỒ ÁN TỐT NGHIỆP KỸ SƯ**

**NGÀNH: CÔNG NGHỆ THÔNG TIN**

**MÃ NGÀNH: 7480201**

**KHOÁ: 2020 - 2025**

***ĐỀ TÀI:***

**XÂY DỰNG WEB PHÂN TÍCH THỊ TRƯỜNG CHỨNG KHOÁN SỬ DỤNG REACTJS VÀ NODEJS**

|  |  |
| --- | --- |
| **GIẢNG VIÊN HƯỚNG DẪN:** | **TS. NGUYỄN ĐĂNG KHOA** |
| **SINH VIÊN THỰC HIỆN:** | **ĐỖ VĂN CHUYÊN** |
| **LỚP:** | **2020CN1** |

**HÀ NỘI 10 - 2024**

# **LỜI CẢM ƠN**

Lời đầu tiên, em không biết nói gì hơn ngoài bày tỏ sự biết ơn sâu sắc đến các Thầy Cô khoa công nghệ thông tin trường Đại học Kiến Trúc Hà Nội. Trong suốt chặng đường học tập và làm đồ án tốt nghiệp em đã luôn nhận được sự hướng dẫn, giúp đỡ tận tình của Thầy Cô.

Đặc biệt, em xin bày tỏ sự kính trọng và lòng biết ơn sâu sắc nhất đến Thầy giáo hướng dẫn TS. Nguyễn Đăng Khoa, Thầy là người đã trực tiếp hướng dẫn, giúp đỡ cho em để em có thể hoàn thành đồ án này. Trong quá trình học tập và nghiên cứu, nếu em có những sai sót gì, kính mong Thầy Cô bỏ qua cho em.

Em xin kính chúc các Thầy Cô luôn luôn khỏe mạnh và ngày một thành công hơn trên con đường giảng dạy của mình.

Em xin trân trọng cảm ơn!

# **LỜI CAM ĐOAN**

Tôi là Đỗ Văn Chuyên, sinh viên lớp 2020CN1. Tôi cam đoan luận văn Đồ án tốt nghiệp “*Xây dựng web phân tích thị trường chứng khoán sử dụng Reactjs và Nodejs*” là công trình nghiên cứu của riêng tôi, có sự hỗ trợ từ giảng viên hướng dẫn là Thầy TS.Nguyễn Đăng Khoa. Các nội dung nghiên cứu và kết quả trong đề tài này là trung thực. Những số liệu trong các bảng biểu phục vụ cho việc phân tích, nhận xét, đánh giá được chính tác giả thu thập từ các nguồn khác nhau có ghi trong phần tài liệu tham khảo.

Nếu phát hiện có bất kỳ sự gian lận nào tôi xin hoàn toàn chịu trách nhiệm trước hội đồng cũng như kết quả luận văn của mình.

|  |  |
| --- | --- |
|  | *Hà Nội, ngày 1 tháng 1 năm 2025*  SINH VIÊN THỰC HIỆN ĐỒ ÁN  Đỗ Văn Chuyên |

# **LỜI NÓI ĐẦU**

Trong bối cảnh công nghệ thông tin hiện nay đang phát triển với tốc độ nhanh chóng trên mọi lĩnh vực, việc ứng dụng các công nghệ mới trong quản lý, kinh doanh, và phân tích thị trường trở nên vô cùng quan trọng. Tại Việt Nam, những năm gần đây chứng kiến sự đầu tư mạnh mẽ vào công nghệ và hạ tầng thông tin. Đặc biệt trong lĩnh vực tài chính, việc khai thác dữ liệu, đưa ra nhận định, và phân tích xu hướng trên thị trường chứng khoán đòi hỏi những công cụ hiện đại và linh hoạt. Chính vì thế, phát triển các hệ thống thông tin và ứng dụng web phục vụ phân tích thị trường chứng khoán không chỉ hỗ trợ các nhà đầu tư đưa ra quyết định kịp thời, mà còn nâng cao hiệu quả, tiết kiệm thời gian và công sức trong quá trình tiếp cận thông tin.

Qua quá trình tìm hiểu và nhận thấy tầm quan trọng của việc ứng dụng công nghệ web trong việc nắm bắt biến động thị trường, em đã quyết định lựa chọn đề tài “XÂY DỰNG WEB PHÂN TÍCH THỊ TRƯỜNG CHỨNG KHOÁN SỬ DỤNG REACTJS VÀ NODEJS”. Mục tiêu của đề tài là xây dựng một trang web phân tích dữ liệu chứng khoán, cung cấp thông tin, biểu đồ, cũng như các công cụ hỗ trợ phân tích một cách trực quan, tương tác và hiệu quả. Với việc kết hợp ReactJS trên giao diện người dùng và NodeJS trong xử lý dữ liệu, hệ thống sẽ tận dụng được thế mạnh của các công nghệ hiện đại, đem lại trải nghiệm tốt cho người dùng, từ các nhà đầu tư cá nhân đến chuyên gia phân tích.

Đồ án tốt nghiệp này sẽ được trình bày qua 4 chương, tập trung vào các khía cạnh từ tổng quan công nghệ, phân tích yêu cầu, thiết kế, đến quá trình triển khai và đánh giá sản phẩm. Đây sẽ là nền tảng giúp em củng cố kiến thức, kỹ năng đã học và áp dụng vào thực tiễn, đồng thời góp phần hoàn thiện năng lực phát triển ứng dụng web phục vụ lĩnh vực tài chính – chứng khoán.

**Chương 1: Tổng quan.**

**Chương 2: Cơ sở lý thuyết áp dụng trong đề tài.**

**Chương 3: Phân tích và thiết kế.**

**Chương 4: Xây dựng web**

Em xin gửi lời cảm ơn tới các Thầy, Cô giảng viên của khoa Công nghệ thông tin trường Đại học Kiến trúc Hà Nội đã cung cấp cơ sở kiến thức về ngành công nghệ thông tin, cảm ơn Thầy TS.Nguyễn Đăng Khoa - Giảng viên hướng dẫn trực tiếp đã hướng dẫn tận tâm, nhiệt tình để em có thể hoàn thành bài báo cáo tốt nghiệp này.

Vì kiến thức của bản thân còn hạn hẹp nên không thể tránh khỏi có thiếu sót và sai lầm, rất mong nhận được sự góp ý từ các thầy, cô giảng viên của khoa Công nghệ thông tin trường Đại học Kiến trúc Hà Nội.

Em xin chân thành cảm ơn!

**MỤC LỤC**

[**LỜI CẢM ƠN** 3](#_Toc190623444)

[**LỜI CAM ĐOAN** 4](#_Toc190623445)

[**LỜI NÓI ĐẦU** 5](#_Toc190623446)

[**MỤC LỤC** 7](#_Toc190623447)

[**MỤC LỤC HÌNH ẢNH** 9](#_Toc190623448)

[**MỤC LỤC BẢNG** 11](#_Toc190623449)

[**CHƯƠNG 1: TỔNG QUAN** 12](#_Toc190623450)

[**1.1. Giới thiệu tổng quan và lý do chọn đề tài** 12](#_Toc190623451)

[**1.1.1. Sơ lược về thị trường chứng khoán.** 12](#_Toc190623452)

[**1.1.2. Quá trình hình thành và phát triển:** 12](#_Toc190623453)

[**1.1.3. Cơ cấu và quy mô** 12](#_Toc190623454)

[**1.1.4. Lý do chọn đề tài** 13](#_Toc190623455)

[**1.2. Mục tiêu đề tài** 14](#_Toc190623456)

[**1.2.1. Lý thuyết** 14](#_Toc190623457)

[**1.2.2. Thực tiễn** 14](#_Toc190623458)

[**1.3. Các bước nghiên cứu** 14](#_Toc190623459)

[**1.4. Đối tượng phân tích thị trường chứng khoán.** 15](#_Toc190623460)

[**1.4.1. Cá nhân Nhà Đầu Tư** 15](#_Toc190623461)

[**1.4.2. Nhà Đầu Tư Tổ Chức** 15](#_Toc190623462)

[**1.4.3. Các Công Ty và Doanh Nghiệp** 15](#_Toc190623463)

[**1.4.4. Các Nhà Đầu Tư Quốc Tế** 15](#_Toc190623464)

[**1.5. Yêu Cầu Chung Để Niêm Yết Cổ Phiếu** 16](#_Toc190623465)

[**1.6. Thị trường chứng khoán biến động.** 16](#_Toc190623466)

[**1.6.1. Yếu Tố Kinh Tế Vĩ Mô** 16](#_Toc190623467)

[**1.6.2. Yếu Tố Chính Trị và Quy Định Pháp Luật** 16](#_Toc190623468)

[**1.6.3. Hiệu Suất và Tin Tức của Doanh Nghiệp** 17](#_Toc190623469)

[**1.6.4. Tâm Lý và Hành Vi Nhà Đầu Tư** 17](#_Toc190623470)

[**1.6.5. Yếu Tố Thị Trường Quốc Tế** 17](#_Toc190623471)

[**1.6.6. Phân Tích Cơ Bản và Kỹ Thuật** 17](#_Toc190623472)

[**1.6.7. Thanh Khoản Thị Trường** 18](#_Toc190623473)

[**1.6.8. Công Nghệ và Đổi Mới** 18](#_Toc190623474)

[**CHƯƠNG 2: CƠ SỞ LÝ THUYẾT** 19](#_Toc190623475)

[**2.1. Định nghĩa Reactjs** 19](#_Toc190623476)

[**2.2. Vòng đời Reactjs và các phần chính của Reactjs** 20](#_Toc190623477)

[**2.2.1. Các Phần Chính Của ReactJS** 20](#_Toc190623478)

[**2.2.2. Các công cụ và thư viện hỗ trợ** 23](#_Toc190623479)

[**2.2.3.Ưu nhược điểm của Reactjs** 24](#_Toc190623480)

[**2.3. Tìm hiểu về Visual Studio Code** 26](#_Toc190623481)

[**2.4. Tìm hiểu về cơ sở dữ liệu SQL** 26](#_Toc190623482)

[**2.4.1. SQL là gì?** 26](#_Toc190623483)

[**2.4.2. Lịch sử hình thành SQL như thế nào?** 27](#_Toc190623484)

[**2.4.3. Ưu và nhược điểm của SQL** 27](#_Toc190623485)

[**CHƯƠNG 3: PHÂN TÍCH VÀ THIẾT KẾ** 28](#_Toc190623486)

[**3.1. Phân tích xác định chức năng yêu cầu hệ thống** 28](#_Toc190623487)

[**3.1.1. Chức năng chính** 28](#_Toc190623488)

[**3.1.2. Mô hình hoạt động client-server** 29](#_Toc190623489)

[**3.2. Các bảng dữ liệu chính** 38](#_Toc190623490)

[**CHƯƠNG 4: XÂY DỰNG WEB** 41](#_Toc190623491)

[**4.1. Xây dựng giao diện** 41](#_Toc190623492)

[**4.1.1. Một số giao diện chính** 41](#_Toc190623493)

[**4.1.2. Danh sách các màn hình** 47](#_Toc190623494)

[**4.2. Xây dựng web phân tích chứng khoán** 48](#_Toc190623495)

[**4.2.1. Cấu trúc thư mục dự án** 48](#_Toc190623496)

[**4.2.2. Thư mục các trang của web** 49](#_Toc190623497)

[**4.3. Xây dựng backend** 50](#_Toc190623498)

[**4.4. Phân tích code các trang chính** 51](#_Toc190623499)

[**4.4.1. Front-end** 51](#_Toc190623500)

[**4.4.1. Back-end.** 61](#_Toc190623501)

[**KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN** 66](#_Toc190623502)

[**TÀI LIỆU THAM KHẢO** 68](#_Toc190623503)

# **MỤC LỤC HÌNH ẢNH**

[Hình 1.1. Component function 20](#_Toc190622058)

[Hình 1.2. Class component 21](#_Toc190622059)

[Hình 1.3. JSX 21](#_Toc190622060)

[Hình 1.4. Props 21](#_Toc190622061)

[Hình 1.5. Component function 21](#_Toc190622062)

[Hình 1.6. Hook. 22](#_Toc190622063)

[Hình 1.7. useContext. 23](#_Toc190622064)

[Hình 1.8. Routing 23](#_Toc190622065)

[Hình 1.9. Tạo project mới bằng lệnh 23](#_Toc190622066)

[Hình 1.10. Redux. 24](#_Toc190622067)

[Hình 1. 11. SQL là gì? 26](#_Toc190622068)

[Hình 2. 1. Biểu đồ phân cấp chức năng 29](#_Toc190622069)

[Hình 2. 2. Biểu đồ use-case chính 30](#_Toc190622070)

[Hình 2. 3. Biểu đồ use-case thành viên 31](#_Toc190622071)

[Hình 2. 4. Biểu đồ use-case trò chuyện 32](#_Toc190622072)

[Hình 2. 5. Biểu đồ use-case thêm vào bộ lọc của tôi 33](#_Toc190622073)

[Hình 2. 6. Biểu đồ use-case thêm vào danh mục của tôi 34](#_Toc190622074)

[Hình 2. 7. Biểu đồ use-case kết bạn 35](#_Toc190622075)

[Hình 2. 8. Biểu đồ usecase sửa trang cá nhân 36](#_Toc190622076)

[Hình 2. 9. Biểu đồ usecase tìm kiếm chức năng 36](#_Toc190622077)

[Hình 2. 10. Biểu đồ usecase quên mật khẩu 37](#_Toc190622078)

[Hình 2. 11. Biểu đồ usecase đổi 38](#_Toc190622079)

[Hình 3. 1. Giao diện đăng ký 41](#_Toc190622080)

[Hình 3. 2. Giao diện đăng nhập 42](#_Toc190622081)

[Hình 3. 3. Giao diện trang chính 43](#_Toc190622082)

[Hình 3. 4. Giao diện thông tin người dùng 43](#_Toc190622083)

[Hình 3. 5. Giao diện trang biểu đồ chi tiết 44](#_Toc190622084)

[Hình 3. 6. Giao diện bảng giá 44](#_Toc190622085)

[Hình 3. 7. Giao diện lọc cổ phiếu 45](#_Toc190622086)

[Hình 3. 8. Giao diện vĩ mô 46](#_Toc190622087)

[Hình 3. 9. Giao diện bảng giá 46](#_Toc190622088)

[Hình 3. 10. Giao diện quên mật khẩu 47](#_Toc190622089)

[Hình 3. 11. Cấu trúc thư mục dự án 49](#_Toc190622090)

[Hình 3. 12. Thư mục chứa các giao diện và chức năng, nội dung. 49](#_Toc190622091)

[Hình 3. 13. Thư mục back-end chính. 50](#_Toc190622092)

[Hình 3. 14. Header. 52](#_Toc190622093)

[Hình 3. 15. Các trang nhỏ của web 53](#_Toc190622094)

[Hình 3. 16. Redux 53](#_Toc190622095)

[Hình 3. 17. Các component của file home. 54](#_Toc190622096)

[Hình 3. 18. Truyền các prop qua các component con 54](#_Toc190622097)

[Hình 3. 19. Socket(real-time) 55](#_Toc190622098)

[Hình 3. 20. Một trong các hook dùng để call api. 56](#_Toc190622099)

[Hình 3. 21. Chức năng tìm kiếm các mã cổ phiếu. 56](#_Toc190622100)

[Hình 3. 22. Socket. 57](#_Toc190622101)

[Hình 3. 23. Call api lọc cổ phiếu. Đầu tiên call api để lấy tất cả danh sách mã cổ phiếu hiện tại, sau đó sẽ lọc theo số liệu mà người dùng yêu cầu. 58](#_Toc190622102)

[Hình 3. 24. Chia component. 58](#_Toc190622103)

[Hình 3. 25. Call các api của màn phân tích. 59](#_Toc190622104)

[Hình 3. 26. Custom và truyền các option để được biểu đồ. 60](#_Toc190622105)

[Hình 3. 27. Chia component trang vĩ mô. 60](#_Toc190622106)

[Hình 3. 28. Tạo option để thể hiện bản đồ Việt Nam. 61](#_Toc190622107)

[Hình 3. 29. Hiển thị biểu đồ. 61](#_Toc190622108)

[Hình 3. 30. Setup port. 62](#_Toc190622109)

[Hình 3. 31. Kết nối SQL. 63](#_Toc190622110)

[Hình 3. 32. Kéo data. 64](#_Toc190622111)

[Hình 3. 33. Middle ware. 65](#_Toc190622112)

[Hình 3. 34. Login api. 65](#_Toc190622113)

# **MỤC LỤC BẢNG**

[Bảng 1.1: Kịch bản usecase đăng ký 32](#_Toc190623279)

[Bảng 1.2: Kịch bản usecase đăng nhập 33](#_Toc190623280)

[Bảng 1.3: Kịch bản usecase trò chuyện 34](#_Toc190623281)

[Bảng 1.4: Kịch bản thêm vào bộ lọc của tôi 35](#_Toc190623282)

[Bảng 1.5: Kịch bản usecase thêm danh mục của tôi 35](#_Toc190623283)

[Bảng 1.6: Kịch bản usecase thêm bạn bè 36](#_Toc190623284)

[Bảng 1.7: Kịch bản usecase chỉnh sửa trang cá nhân 37](#_Toc190623285)

[Bảng 1.8: Kịch bản usecase tìm kiếm chức năng 38](#_Toc190623286)

[Bảng 1.7: Kịch bản usecase quên mật khẩu 38](#_Toc190623287)

[Bảng 1.8: Kịch bản usecase quên mật khẩu 39](#_Toc190623288)

[Bảng 1.9. Bảng user và các thuộc tính 40](#_Toc190623289)

[Bảng 1.10. Bảng bộ lọc của tôi 40](#_Toc190623290)

[Bảng 1.11. Bảng mối quan hệ 41](#_Toc190623291)

[Bảng 1.12. Bảng vốn đầu tư tỉnh thành 41](#_Toc190623292)

[Bảng 1.13. Bảng các mã chứng khoán 41](#_Toc190623293)

[Bảng 2.1. Bảng danh sách các màn chính 49](#_Toc190623294)

# **CHƯƠNG 1: TỔNG QUAN**

## **1.1. Giới thiệu tổng quan và lý do chọn đề tài**

### **1.1.1. Sơ lược về thị trường chứng khoán.**

Thị trường chứng khoán là một nền tảng tài chính nơi các công ty niêm yết cổ phiếu và trái phiếu được mua bán bởi các nhà đầu tư. Đây là nơi kết nối giữa các doanh nghiệp cần huy động vốn và các nhà đầu tư tìm kiếm cơ hội sinh lời từ việc đầu tư vào các công ty hoặc các công cụ tài chính khác.

### **1.1.2. Quá trình hình thành và phát triển:**

* **Giai đoạn khởi đầu (2000-2005):** Đây là giai đoạn thị trường mới manh nha, với số lượng cổ phiếu niêm yết còn ít và quy mô vốn hóa thị trường hạn chế. Nhà đầu tư chủ yếu là tổ chức trong nước, cá nhân có mức độ tham gia khiêm tốn.
* **Giai đoạn mở rộng (2006-2010):** Sau khi Việt Nam gia nhập Tổ chức Thương mại Thế giới (WTO) năm 2007, thị trường chứng khoán Việt Nam thu hút sự chú ý ngày càng lớn của các nhà đầu tư nước ngoài. Quy mô vốn hóa thị trường tăng mạnh, nhiều doanh nghiệp lớn thực hiện cổ phần hóa và niêm yết trên sàn giao dịch. Tuy nhiên, thị trường cũng trải qua biến động lớn do ảnh hưởng từ khủng hoảng tài chính toàn cầu năm 2008.
* **Giai đoạn nâng cao tính chuyên nghiệp (2011-2019):** Các quy định pháp lý, cơ chế quản lý, giám sát dần hoàn thiện, thị trường phát triển thêm các sản phẩm mới như trái phiếu doanh nghiệp, chứng quyền có bảo đảm, quỹ ETF,... Việc mở rộng room ngoại, áp dụng các công nghệ giao dịch hiện đại giúp thị trường ngày càng minh bạch, tăng thanh khoản và tạo môi trường đầu tư thuận lợi hơn.
* **Giai đoạn chuyển đổi và hội nhập sâu rộng (2020-đến nay):** Việt Nam được nhiều tổ chức xếp hạng xem xét đưa vào nhóm thị trường mới nổi, mở ra triển vọng thu hút nguồn vốn quốc tế. Công cuộc số hóa, áp dụng các công nghệ mới trong giao dịch và quản lý, cùng với chính sách vĩ mô ổn định, tiếp tục tạo động lực để thị trường chứng khoán trở thành kênh dẫn vốn quan trọng cho nền kinh tế.

### 

### **1.1.3. Cơ cấu và quy mô**

**1.1.3.1. Thị trường chứng khoán Việt Nam bao gồm:**

* **Sở Giao dịch Chứng khoán TP. Hồ Chí Minh (HOSE):** Tập trung các doanh nghiệp có quy mô lớn, chất lượng cao.
* **Sở Giao dịch Chứng khoán Hà Nội (HNX)**: Bao gồm các công ty với quy mô vừa và nhỏ, đồng thời tổ chức giao dịch trái phiếu Chính phủ, thị trường UPCoM (thị trường giao dịch cổ phiếu của các công ty đại chúng chưa niêm yết).
* **Trung tâm Lưu ký Chứng khoán Việt Nam (VSD)**: Quản lý hoạt động đăng ký, lưu ký, thanh toán bù trừ chứng khoán, đảm bảo tính an toàn và tin cậy trong giao dịch.

**1.1.3.2. Đặc điểm và xu hướng**

Tốc độ tăng trưởng nhanh, được hỗ trợ bởi nền kinh tế năng động, lực lượng lao động trẻ và sự quan tâm ngày càng lớn của cộng đồng nhà đầu tư trong và ngoài nước.

* Chính sách nhà nước luôn hướng đến mục tiêu minh bạch, hiện đại hóa, tăng tính thanh khoản, cũng như hội nhập quốc tế sâu rộng.
* Sự ra đời của nhiều sản phẩm chứng khoán phái sinh, hợp đồng tương lai, ETF và các loại chứng quyền mang đến nhiều lựa chọn đầu tư, giúp thị trường từng bước tiệm cận với chuẩn mực quốc tế.

### **1.1.4. Lý do chọn đề tài**

* **Nhu cầu phân tích thị trường chứng khoán ngày càng tăng**: Trong bối cảnh kinh tế ngày một phát triển, việc đầu tư chứng khoán đã trở thành xu hướng được nhiều cá nhân và tổ chức quan tâm. Sự xuất hiện liên tục của các doanh nghiệp niêm yết, cùng với việc thay đổi giá trị cổ phiếu theo thời gian thực, đòi hỏi những công cụ phân tích thị trường hiệu quả. Một hệ thống phân tích chứng khoán chuyên sâu, cập nhật dữ liệu liên tục và trực quan sẽ giúp nhà đầu tư có cái nhìn tổng thể, dễ dàng đưa ra quyết định kịp thời và chính xác hơn.
* **Hạn chế trong tiếp cận thông tin và phân tích truyền thống**: Trước đây, nhà đầu tư thường tiếp cận thông tin thông qua báo chí, tạp chí, kênh truyền hình tài chính hoặc sử dụng những phần mềm phân tích chuyên biệt phức tạp. Tuy nhiên, những kênh thông tin này có hạn chế về tính tương tác, khả năng tùy biến, hoặc chi phí cao. Việc xây dựng một ứng dụng web tập trung, dễ sử dụng, trực quan sẽ giúp giải quyết những khó khăn này, cho phép nhà đầu tư nhanh chóng tìm kiếm, phân tích, so sánh dữ liệu và xu hướng cổ phiếu một cách linh hoạt hơn.
* **Ứng dụng công nghệ web hiện đại trong phát triển hệ thống phân tích**: Sự phổ biến của các công nghệ như ReactJS ở phía giao diện người dùng (Front-end) và NodeJS ở phía máy chủ (Back-end) giúp việc xây dựng các ứng dụng web hiệu suất cao, dễ mở rộng và bảo trì trở nên khả thi. Đây là những công nghệ đang được ưa chuộng và khuyến khích học tập trong ngành công nghệ thông tin, bởi khả năng phản ứng thời gian thực, giao diện tương tác mượt mà, hỗ trợ nhiều luồng dữ liệu lớn và tích hợp linh hoạt với các dịch vụ bên thứ ba.
* **Cơ hội học tập và nâng cao kỹ năng**: Thực hiện đề tài này không chỉ dừng lại ở khía cạnh đáp ứng nhu cầu thực tiễn, mà còn mang lại cơ hội cho người thực hiện nâng cao kiến thức và kỹ năng. Việc tìm hiểu, phân tích nghiệp vụ trong lĩnh vực chứng khoán, kết hợp với triển khai những công nghệ Web hiện đại, giúp củng cố nền tảng kỹ thuật, kỹ năng tư duy hệ thống, khả năng phát triển giải pháp công nghệ thông tin hoàn chỉnh.
* **Đóng góp vào thị trường tài chính trong nước**: Xây dựng một ứng dụng phân tích thị trường chứng khoán tiện ích, thân thiện với người dùng, có khả năng cập nhật và xử lý dữ liệu linh hoạt, sẽ góp phần hỗ trợ nhà đầu tư Việt Nam trong việc tiếp cận thông tin hiệu quả hơn. Qua đó, đề tài cũng góp phần thúc đẩy tính minh bạch, hiện đại hóa và hội nhập của thị trường chứng khoán Việt Nam trong bối cảnh quốc tế hóa.

## **1.2. Mục tiêu đề tài**

### **1.2.1.** **Lý thuyết**

* Nghiên cứu công cụ thiết kế như Canva, Figma.
* Nghiên cứu ngôn ngữ lập trình Reactjs, Nodejs.
* Nghiên cứu xây dựng giao diện với Reactjs.
* Nghiên cứu hệ quản trị cơ sở dữ liệu SQL.

### **1.2.2. Thực tiễn**

* Xây dựng ứng dụng gồm client, server có khả năng giao tiếp với nhau.
* Ứng dụng hỗ trợ các tính năng tìm kiếm mã chứng khoán, tradingview, biểu đồ hiển thị dữ liệu,… .

## **1.3. Các bước nghiên cứu**

* Phân tích yêu cầu.
* Lựa chọn công nghệ.
* Nghiên cứu cơ sở lý thuyết của công nghệ đã chọn.
* Áp dụng lý thuyết vào xây dựng ứng dụng thực tiễn.
* Kiểm tra, tham khảo các ứng dụng khác để tối ưu hóa ứng dụng.

## **1.4. Đối tượng phân tích thị trường chứng khoán.**

### **1.4.1. Cá nhân Nhà Đầu Tư**

* **Nhà Đầu Tư Cá Nhân:** Những người đầu tư cho mục tiêu cá nhân như tiết kiệm hưu trí, tăng trưởng tài sản, hoặc mục tiêu tài chính cá nhân khác.
* **Nhà Đầu Tư Tự Do (Retail Investors):** Những người tham gia thị trường chứng khoán thông qua các sàn giao dịch trực tuyến, thường với số vốn nhỏ đến trung bình.

### **1.4.2. Nhà Đầu Tư Tổ Chức**

* **Quỹ Đầu Tư (Mutual Funds):** Các quỹ tập trung vào việc đầu tư vào danh mục chứng khoán đa dạng để đạt được lợi nhuận cho nhà đầu tư của mình.
* **Quỹ Hưu Trí (Pension Funds):** Các quỹ này quản lý vốn hưu trí của người lao động và đầu tư vào thị trường chứng khoán để đảm bảo lợi nhuận lâu dài.
* **Ngân Hàng Đầu Tư (Investment Banks):** Các tổ chức tài chính tham gia vào việc tư vấn, môi giới và đầu tư vào các giao dịch chứng khoán lớn.
* **Các Công Ty Bảo Hiểm:** Đầu tư vào chứng khoán để quản lý rủi ro và tối ưu hóa lợi nhuận cho các chính sách bảo hiểm.

### **1.4.3. Các Công Ty và Doanh Nghiệp**

* **Công Ty Chứng Khoán:** Cung cấp dịch vụ môi giới, tư vấn và quản lý tài sản cho nhà đầu tư.
* **Các Doanh Nghiệp Niêm Yết:** Phân tích hiệu quả kinh doanh và triển vọng phát triển để thu hút nhà đầu tư và tăng giá trị cổ phiếu.

### **1.4.4. Các Nhà Đầu Tư Quốc Tế**

* **Nhà Đầu Tư Nước Ngoài:** Đầu tư vào các thị trường chứng khoán nước ngoài để đa dạng hóa danh mục đầu tư và tận dụng các cơ hội tăng trưởng toàn cầu.
* **Các Quỹ Đầu Tư Toàn Cầu:** Quản lý danh mục đầu tư đa quốc gia, đầu tư vào nhiều thị trường chứng khoán khác nhau.

**1.4.5. Các Tổ Chức Phi Lợi Nhuận và Các Nhà Nghiên Cứu**

* **Các Trung Tâm Nghiên Cứu Tài Chính:** Nghiên cứu và cung cấp báo cáo, phân tích về thị trường chứng khoán và các xu hướng kinh tế.
* **Các Tổ Chức Phi Lợi Nhuận:** Đôi khi tham gia vào việc nghiên cứu và giám sát thị trường để đảm bảo tính minh bạch và trách nhiệm xã hội.

## **1.5. Yêu Cầu Chung Để Niêm Yết Cổ Phiếu**

* **Vốn Điều Lệ:** Đạt mức vốn tối thiểu yêu cầu của sàn giao dịch.
* **Lịch Sử Hoạt Động và Tài Chính:** Có lịch sử hoạt động kinh doanh và tài chính ổn định trong một khoảng thời gian nhất định.
* **Quản Trị Công Ty:** Có hệ thống quản trị công ty minh bạch, hiệu quả và tuân thủ các quy định pháp luật.
* **Công Khai Thông Tin:** Tuân thủ các quy định về công khai thông tin, đảm bảo tính minh bạch và chính xác.

## **1.6. Thị trường chứng khoán biến động.**

### **1.6.1. Yếu Tố Kinh Tế Vĩ Mô**

* **Tăng Trưởng Kinh Tế (GDP):** Khi nền kinh tế tăng trưởng mạnh, doanh nghiệp thường có lợi nhuận cao hơn, dẫn đến giá cổ phiếu tăng. Ngược lại, khi kinh tế suy thoái, giá cổ phiếu có thể giảm.
* **Lạm Phát và Lãi Suất:** Lạm phát cao có thể làm giảm sức mua của người tiêu dùng và lợi nhuận của doanh nghiệp, ảnh hưởng tiêu cực đến giá cổ phiếu. Lãi suất tăng thường khiến chi phí vay mượn tăng, ảnh hưởng đến khả năng mở rộng kinh doanh của các công ty và làm giảm hấp dẫn của cổ phiếu so với các công cụ đầu tư khác như trái phiếu.
* **Tỷ Giá Ngoại Tệ:** Sự biến động của tỷ giá ngoại tệ có thể ảnh hưởng đến các công ty xuất khẩu và nhập khẩu, từ đó ảnh hưởng đến giá cổ phiếu của các doanh nghiệp này.

### **1.6.2. Yếu Tố Chính Trị và Quy Định Pháp Luật**

* **Ổn Định Chính Trị:** Môi trường chính trị ổn định tạo điều kiện thuận lợi cho các doanh nghiệp phát triển, làm tăng niềm tin của nhà đầu tư và giá cổ phiếu. Ngược lại, bất ổn chính trị có thể gây ra sự không chắc chắn, dẫn đến giảm giá cổ phiếu.
* **Chính Sách Thuế và Quy Định:** Các thay đổi trong chính sách thuế, quy định về môi trường, lao động và các quy định khác có thể ảnh hưởng đến hoạt động kinh doanh của các công ty, từ đó ảnh hưởng đến giá cổ phiếu.

### **1.6.3. Hiệu Suất và Tin Tức của Doanh Nghiệp**

* **Báo Cáo Tài Chính:** Các báo cáo tài chính tốt hơn dự kiến có thể làm tăng giá cổ phiếu, trong khi báo cáo tài chính yếu hơn dự kiến có thể làm giảm giá cổ phiếu.
* **Tin Tức và Sự Kiện:** Tin tức về sáp nhập, mua lại, thay đổi ban lãnh đạo, hoặc các sự kiện quan trọng khác có thể tạo ra biến động lớn trong giá cổ phiếu.

### **1.6.4. Tâm Lý và Hành Vi Nhà Đầu Tư**

* **Tâm Lý Thị Trường:** Sự lạc quan hoặc bi quan của nhà đầu tư có thể dẫn đến các đợt tăng hoặc giảm mạnh trên thị trường. Thị trường chứng khoán không chỉ phản ánh các yếu tố cơ bản mà còn phản ánh cảm xúc và kỳ vọng của nhà đầu tư.
* **Đám Đông và Hiệu Ứng Xô Động:** Khi nhiều nhà đầu tư tham gia mua hoặc bán cùng một lúc, điều này có thể làm tăng tính biến động của thị trường. Hiệu ứng đám đông thường dẫn đến các đợt tăng hoặc giảm không dự đoán trước.

### **1.6.5. Yếu Tố Thị Trường Quốc Tế**

**Thị Trường Chứng Khoán Quốc Tế:** Sự biến động trên các thị trường chứng khoán lớn như Mỹ, Trung Quốc, hay châu Âu có thể ảnh hưởng đến các thị trường khác. Ví dụ, một cuộc khủng hoảng tài chính ở Mỹ có thể lan tỏa và gây ảnh hưởng đến các thị trường ở Việt Nam.

* **Sự Kiện Toàn Cầu:** Các sự kiện như chiến tranh, thiên tai, đại dịch (như COVID-19) có thể gây ra sự bất ổn và ảnh hưởng đến thị trường chứng khoán toàn cầu.

### **1.6.6. Phân Tích Cơ Bản và Kỹ Thuật**

* **Phân Tích Cơ Bản:** Nhà đầu tư sử dụng phân tích cơ bản để đánh giá giá trị thực của cổ phiếu dựa trên các yếu tố tài chính, kinh tế và quản trị công ty. Những thay đổi trong giá trị thực này có thể dẫn đến sự điều chỉnh giá cổ phiếu trên thị trường.
* **Phân Tích Kỹ Thuật:** Phân tích kỹ thuật dựa trên các biểu đồ giá và các chỉ số kỹ thuật để dự đoán xu hướng tương lai của giá cổ phiếu. Sự thay đổi trong các xu hướng này có thể dẫn đến các đợt mua hoặc bán theo kế hoạch của nhà đầu tư kỹ thuật.

### **1.6.7. Thanh Khoản Thị Trường**

* **Khối Lượng Giao Dịch:** Thị trường có thanh khoản cao thường có sự biến động giá thấp hơn vì có nhiều nhà đầu tư tham gia và dễ dàng mua bán cổ phiếu. Ngược lại, thị trường thanh khoản thấp có thể dẫn đến biến động giá lớn hơn.

### **1.6.8. Công Nghệ và Đổi Mới**

* **Công Nghệ Mới:** Sự xuất hiện của công nghệ mới hoặc các đổi mới trong ngành có thể tạo ra cơ hội tăng trưởng cho các công ty, từ đó làm tăng giá cổ phiếu. Ngược lại, sự lỗi thời của công nghệ có thể làm giảm giá cổ phiếu của các công ty không thích ứng kịp.

# **CHƯƠNG 2: CƠ SỞ LÝ THUYẾT**

## **2.1. Định nghĩa Reactjs**

**ReactJS** là một thư viện JavaScript mã nguồn mở, được phát triển bởi Facebook (nay là Meta) và cộng đồng lập trình viên, nhằm xây dựng giao diện người dùng (UI) cho các ứng dụng web. Điểm nổi bật của ReactJS là khái niệm **“Component”** – các thành phần giao diện được tách biệt và tái sử dụng, giúp việc phát triển trở nên linh hoạt, dễ bảo trì và mở rộng. ReactJS sử dụng **Virtual DOM** (DOM ảo) để cập nhật và phản hồi với dữ liệu nhanh hơn so với thao tác trực tiếp trên DOM thật, từ đó cải thiện hiệu năng và trải nghiệm người dùng. Bằng cách chỉ cập nhật những phần giao diện bị thay đổi, ReactJS giúp giảm thiểu số lần thao tác trên trình duyệt, giúp ứng dụng web phản hồi nhanh và mượt mà.

* **Đặc điểm**
* **Component-Based:**

ReactJS tổ chức giao diện dưới dạng các thành phần nhỏ (components). Mỗi thành phần là một khối giao diện độc lập, có thể được tái sử dụng và kết hợp với nhau để tạo nên ứng dụng hoàn chỉnh. Điều này giúp cho việc phát triển và bảo trì mã nguồn trở nên dễ dàng, linh hoạt.

* **Tính ngang hàng của các ứng dụng**

ReactJS sử dụng Virtual DOM để tối ưu việc cập nhật giao diện. Khi dữ liệu thay đổi, ReactJS so sánh trạng thái giao diện mới với trạng thái trước đó trên DOM ảo, sau đó chỉ cập nhật những phần tử thực sự thay đổi trên DOM thật. Cách làm này giúp tăng hiệu suất, giảm số lần thao tác DOM và cải thiện tốc độ phản hồi của ứng dụng.

* **One-way Data Flow (Luồng dữ liệu một chiều):**

Dữ liệu trong ReactJS thường đi theo một chiều từ cha xuống con, giúp việc quản lý dữ liệu, theo dõi và kiểm soát luồng thông tin trở nên đơn giản hơn. Tính năng này làm tăng tính dự đoán được của ứng dụng, giảm khả năng mắc lỗi, từ đó giúp việc kiểm thử và gỡ lỗi dễ dàng hơn.

* **JSX (JavaScript XML):**

ReactJS đề xuất sử dụng JSX - một cú pháp mở rộng cho JavaScript cho phép bạn viết cấu trúc giao diện trong JavaScript. JSX trông giống như HTML, nhưng thực chất được chuyển đổi về JavaScript. Điều này giúp lập trình viên mô tả giao diện một cách trực quan, dễ đọc, đồng thời vẫn tận dụng được sức mạnh của JavaScript.

* **Hệ sinh thái phong phú:**

ReactJS có một hệ sinh thái phát triển mạnh mẽ, bao gồm nhiều thư viện, công cụ, gói tiện ích, cũng như sự hỗ trợ từ cộng đồng lớn. Điều này giúp bạn dễ dàng tìm kiếm giải pháp cho các yêu cầu phức tạp, mở rộng tính năng và tích hợp ReactJS với các công nghệ khác.

* **Khả năng kết hợp với các công nghệ khác:**

ReactJS chỉ tập trung vào phần hiển thị (view layer) nên có thể dễ dàng tích hợp với các framework, thư viện khác trong việc quản lý trạng thái (Redux, MobX), định tuyến (React Router), hoặc kết nối với back-end (NodeJS, GraphQL).

## **2.2. Vòng đời Reactjs và các phần chính của Reactjs**

**Vòng đời (Lifecycle) trong ReactJS** thường được đề cập đến khi làm việc với **Class Component**, mô tả quá trình khởi tạo, cập nhật và hủy bỏ một thành phần giao diện (component). Trong các phiên bản React hiện đại (từ React 16.8 trở đi), **Hooks** cũng ra đời như một giải pháp thay thế linh hoạt, giúp quản lý trạng thái và vòng đời trong **Function Component**. Tuy nhiên, ở đây sẽ tóm tắt vòng đời truyền thống của Class Component, bao gồm ba giai đoạn chính:

### **2.2.1. Các Phần Chính Của ReactJS**

**a. Components (Thành Phần)**

* **Component Function (Hàm Thành Phần):** Là các hàm JavaScript nhận các đối số (props) và trả về các phần tử React (React elements). Đây là cách phổ biến để định nghĩa các component hiện đại.

A computer screen shot of text

Description automatically generated

Hình 1.1. Component function

* **Class Components (Thành Phần Lớp):** Trước đây, React hỗ trợ các component dựa trên lớp để sử dụng trạng thái (state) và các phương thức vòng đời. Tuy nhiên, với sự xuất hiện của Hooks, việc sử dụng các class components đã giảm đi.

**A screen shot of a computer

Description automatically generated**

Hình 1.2. Class component

**b.** **JSX (JavaScript XML)**

* **JSX** là cú pháp mở rộng của JavaScript, cho phép viết các phần tử HTML trong JavaScript. Nó giúp mã nguồn trở nên dễ đọc và viết hơn.

A black and red text

Description automatically generated

Hình 1.3. JSX

**c. Props (Thuộc Tính)**

* **Props** là các đối số được truyền vào các component, giúp tái sử dụng và cấu hình các component với dữ liệu khác nhau.

A screenshot of a computer code

Description automatically generated

Hình 1.4. Props

**d. State (Trạng Thái)**

* **State** là dữ liệu nội bộ của một component, có thể thay đổi theo thời gian và ảnh hưởng đến cách component hiển thị.

A screen shot of a computer code

Description automatically generated

Hình 1.5. Component function

**e. Hooks**

* **Hooks** là các hàm đặc biệt cho phép sử dụng state và các tính năng khác của React trong function components mà không cần viết class.
* **useState:** Quản lý state trong function components.
* **useEffect:** Thực hiện các side effects như gọi API, thao tác DOM.
* **useContext, useReducer, etc.:** Các hooks khác để quản lý context, reducer, và nhiều tính năng nâng cao.

****

Hình 1.6. Hook.

**f. Context**

* **Context** cho phép chia sẻ dữ liệu giữa các component mà không cần phải truyền props qua từng cấp component.

A screen shot of a computer program

Description automatically generated

Hình 1.7. useContext.

**g. Routing**

* **React Router** là thư viện phổ biến để quản lý điều hướng và routing trong ứng dụng React.



Hình 1.8. Routing

### **2.2.2. Các công cụ và thư viện hỗ trợ**

**a. Create React App**

* **Create React App** là công cụ khởi tạo dự án React nhanh chóng, cấu hình sẵn cho webpack, Babel, và các công cụ khác.

A screenshot of a computer

Description automatically generated

Hình 1.9. Tạo project mới bằng lệnh

**b. Redux**

* **Redux** là thư viện quản lý state dựa trên mô hình flux, giúp quản lý state ứng dụng một cách có cấu trúc và dễ theo dõi.

A close-up of a computer screen

Description automatically generated

Hình 1.10. Redux.

### **2.2.3.Ưu nhược điểm của Reactjs**

ReactJS là một thư viện JavaScript phổ biến được sử dụng rộng rãi để xây dựng giao diện người dùng (UI) cho các ứng dụng web hiện đại. Dưới đây là tổng quan về những ưu điểm và nhược điểm của ReactJS:

**a.Ưu Điểm của ReactJS**

1. **Kiến Trúc Dựa Trên Thành Phần (Component-Based Architecture)**
   * **Tái Sử Dụng Thành Phần:** React cho phép phát triển các thành phần UI độc lập và có thể tái sử dụng trong nhiều phần của ứng dụng, giúp giảm thiểu mã lặp lại và tăng tính nhất quán.
   * **Dễ Bảo Trì:** Việc chia nhỏ ứng dụng thành các thành phần nhỏ giúp dễ dàng quản lý, bảo trì và phát triển ứng dụng.
2. **Virtual DOM**
   * **Hiệu Suất Cao:** React sử dụng Virtual DOM để tối ưu hóa việc cập nhật giao diện. Thay vì cập nhật toàn bộ DOM mỗi khi có thay đổi, React chỉ cập nhật các phần cần thiết, giảm thiểu thao tác DOM và cải thiện hiệu suất.
3. **JSX (JavaScript XML)**
   * **Cú Pháp Dễ Hiểu:** JSX cho phép viết mã HTML trong JavaScript, làm cho việc phát triển giao diện trở nên trực quan và dễ hiểu hơn.
   * **Hỗ Trợ Công Cụ Phát Triển:** JSX tích hợp tốt với các công cụ phát triển như Babel, giúp chuyển đổi mã JSX thành JavaScript thuần.
4. **Dòng Dữ Liệu Một Chiều (Unidirectional Data Flow)**
   * **Quản Lý Dữ Liệu Dễ Dàng:** Dòng dữ liệu một chiều giúp dễ dàng theo dõi và quản lý dữ liệu trong ứng dụng, giảm thiểu các lỗi liên quan đến trạng thái phức tạp.
5. **Cộng Đồng Mạnh Mẽ và Hệ Sinh Thái Phong Phú**
   * **Nhiều Thư Viện và Công Cụ:** Sự hỗ trợ từ cộng đồng lớn cung cấp nhiều thư viện, công cụ và tài nguyên giúp phát triển ứng dụng nhanh chóng và hiệu quả.
   * **Hỗ Trợ Tốt:** Cộng đồng React rộng lớn luôn sẵn sàng hỗ trợ, cung cấp giải pháp cho các vấn đề phát sinh trong quá trình phát triển.
6. **Khả Năng Mở Rộng và Tích Hợp**
   * **Linh Hoạt:** React có thể tích hợp dễ dàng với các thư viện và framework khác như Redux, React Router, giúp mở rộng chức năng của ứng dụng.
   * **Phát Triển Di Động với React Native:** React Native cho phép phát triển ứng dụng di động cho cả iOS và Android bằng cách sử dụng cùng một kiến thức về React.
7. **Hỗ Trợ Server-Side Rendering (SSR)**
   * **Cải Thiện SEO và Tốc Độ Tải Trang:** Sử dụng các framework như Next.js, React hỗ trợ render phía máy chủ giúp cải thiện SEO và tốc độ tải trang ban đầu.

**b. Nhược Điểm của ReactJS**

1. **Đường Cong Học Cao (Steep Learning Curve)**
   * **JSX và Các Khái Niệm Mới:** Việc làm quen với JSX và các khái niệm như hooks, state management có thể gây khó khăn cho người mới bắt đầu.
   * **Sự Phức Tạp của Ecosystem:** Hệ sinh thái rộng lớn với nhiều thư viện và công cụ có thể khiến người học cảm thấy quá tải.
2. **Hệ Sinh Thái Thay Đổi Nhanh Chóng**
   * **Cập Nhật Liên Tục:** React và các thư viện liên quan thường xuyên được cập nhật, đôi khi làm cho việc duy trì và nâng cấp dự án trở nên khó khăn.
   * **Không Thống Nhất:** Sự đa dạng của các thư viện và phương pháp có thể dẫn đến thiếu sự nhất quán trong việc phát triển ứng dụng.
3. **Yêu Cầu Boilerplate Code Với Các Công Cụ Quản Lý Trạng Thái**
   * **Redux và Các Thư Viện Quản Lý Trạng Thái Khác:** Việc thiết lập Redux hoặc các thư viện quản lý trạng thái khác thường yêu cầu viết nhiều mã lặp lại, làm tăng độ phức tạp của mã nguồn.
4. **Chỉ Là Thư Viện Cho Lớp View**
   * **Cần Thêm Thư Viện Khác:** Để xây dựng một ứng dụng hoàn chỉnh, bạn cần tích hợp React với các thư viện khác cho routing, quản lý trạng thái, và các chức năng khác, điều này có thể làm tăng độ phức tạp của dự án.
5. **Thách Thức Về SEO Mặc Dù Có SSR**
   * **Cần Thiết Cấu Bổ Sung:** Mặc dù hỗ trợ SSR cải thiện SEO, việc cấu hình và triển khai SSR có thể phức tạp và đòi hỏi thêm công cụ như Next.js hoặc Gatsby.
6. **Không Phù Hợp Với Các Ứng Dụng Nhỏ**
   * **Overkill Cho Các Dự Án Nhỏ:** Đối với các ứng dụng nhỏ hoặc tĩnh, React có thể là quá phức tạp và không cần thiết, làm tăng thời gian phát triển và tài nguyên cần thiết.
7. **Vấn Đề về Tối Ưu Hiệu Suất Trong Một Số Trường Hợp**
   * **Cần Tối Ưu Thủ Công:** Trong một số trường hợp phức tạp, việc tối ưu hóa hiệu suất yêu cầu sự can thiệp thủ công như shouldComponentUpdate hoặc React.memo, điều này có thể tăng độ phức tạp của mã nguồn.

## **2.3. Tìm hiểu về Visual Studio Code**

Visual Studio Code (VS Code) là một trình soạn thảo mã nguồn mở, miễn phí và mạnh mẽ được phát triển bởi Microsoft. Được phát hành lần đầu vào năm 2015, VS Code nhanh chóng trở thành một trong những công cụ phổ biến nhất đối với lập trình viên trên toàn thế giới nhờ vào tính linh hoạt, tính năng phong phú và khả năng tùy biến cao. Dưới đây là tổng quan chi tiết về VS Code, bao gồm các tính năng chính, ưu điểm, cách sử dụng và các mẹo hữu ích.

## **2.4. Tìm hiểu về cơ sở dữ liệu SQL**

### **2.4.1. SQL là gì?**

A logo of a database

Description automatically generated

Hình 1. 1. SQL là gì?

SQL (viết tắt của **Structured Query Language**) là một ngôn ngữ lập trình đặc biệt được thiết kế để quản lý và thao tác dữ liệu trong các hệ quản trị cơ sở dữ liệu quan hệ (**RDBMS** - Relational Database Management Systems). SQL là tiêu chuẩn quốc tế để tương tác với các cơ sở dữ liệu, được sử dụng rộng rãi trong phát triển phần mềm, phân tích dữ liệu, và quản trị hệ thống.

### **2.4.2. Lịch sử hình thành SQL như thế nào?**

**a. Ban Đầu Phát Triển SQL**

* **Donald D. Chamberlin và Raymond F. Boyce:** Hai nhà nghiên cứu của IBM đã phát triển SQL (ban đầu gọi là SEQUEL – Structured English Query Language) vào giữa những năm 1970. SQL được thiết kế để cho phép người dùng dễ dàng truy vấn và quản lý dữ liệu bằng cách sử dụng cú pháp gần giống tiếng Anh.

**b. Phiên Bản Đầu Tiên**

* **1979:** IBM phát hành phiên bản đầu tiên của SQL như một phần của dự án System R. Ngôn ngữ này nhanh chóng thu hút sự chú ý vì tính đơn giản và mạnh mẽ của nó trong việc quản lý dữ liệu.

### **2.4.3. Ưu và nhược điểm của SQL**

* **Ưu điểm của SQL**
* Tiêu Chuẩn và Phổ Biến
* Khả Năng Truy Vấn Mạnh Mẽ
* Đảm Bảo Tính Toàn Vẹn Dữ Liệu (Data Integrity)
* Hiệu Suất Cao
* Bảo Mật và Quản Lý Quyền Hạn
* Khả Năng Mở Rộng (Scalability)
* Hỗ Trợ Công Cụ và Tích Hợp
* **Nhược điểm của SQL**
* Hạn Chế Với Dữ Liệu Phi Quan Hệ (Non-Relational Data)
* Hiệu Suất Với Các Ứng Dụng Lớn
* Đòi Hỏi Kiến Thức Chuyên Sâu
* Khả Năng Mở Rộng Hạn Chế So Với NoSQL
* Khó Khăn Trong Việc Xử Lý Các Giao Dịch Phức Tạp
* Phụ Thuộc Vào Kiến Trúc Cơ Sở Dữ Liệu Quan Hệ
* Chi Phí Bảo Trì và Phát Triển

# **CHƯƠNG 3: PHÂN TÍCH VÀ THIẾT KẾ**

## **3.1. Phân tích xác định chức năng yêu cầu hệ thống**

### **3.1.1. Chức năng chính**

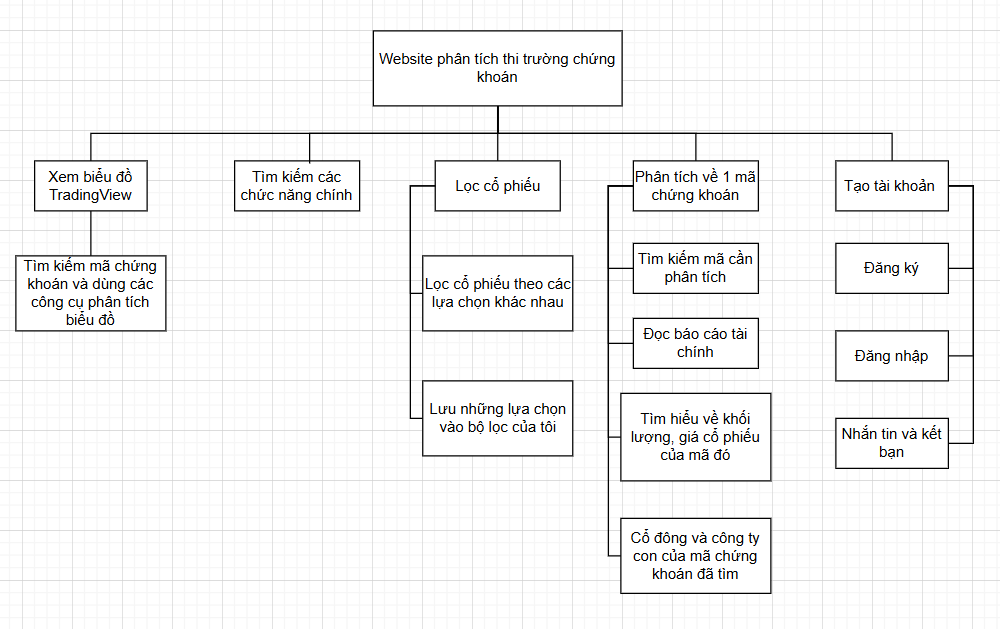
**3.1.1.1. Chức năng của Server**

* Quản lý user:
* Thêm, sửa, xoá người dùng.
* Quản lý cơ sở dữ liệu:
* Quản lý các dữ liệu kéo về từ báo cáo tài chính.
* Quản lý dữ liệu mã cổ phiếu.

**3.1.1.2. Chức năng của Client**

* Cho phép người dùng đăng ký tài khoản, đăng nhập hệ thống, lấy lại mật khẩu.
* Đăng ký: Xử lý, kiểm tra input của người dùng, input sẽ được gửi lên server để xử lý tiếp. Nhận response từ server và thông báo cho người dùng.
* Đăng nhập: gửi yêu cầu đăng nhập của người dùng lên server và nhận response từ server.
* Cho phép người dùng thay đổi thông tin cá nhân, thay đổi mật khẩu, thay đổi hình đại diện.
* Người dùng có thể xem biểu đồ, phân tích, tải báo cáo.

### **3.1.2. Mô hình hoạt động client-server**



Hình 2. 1. Biểu đồ phân cấp chức năng

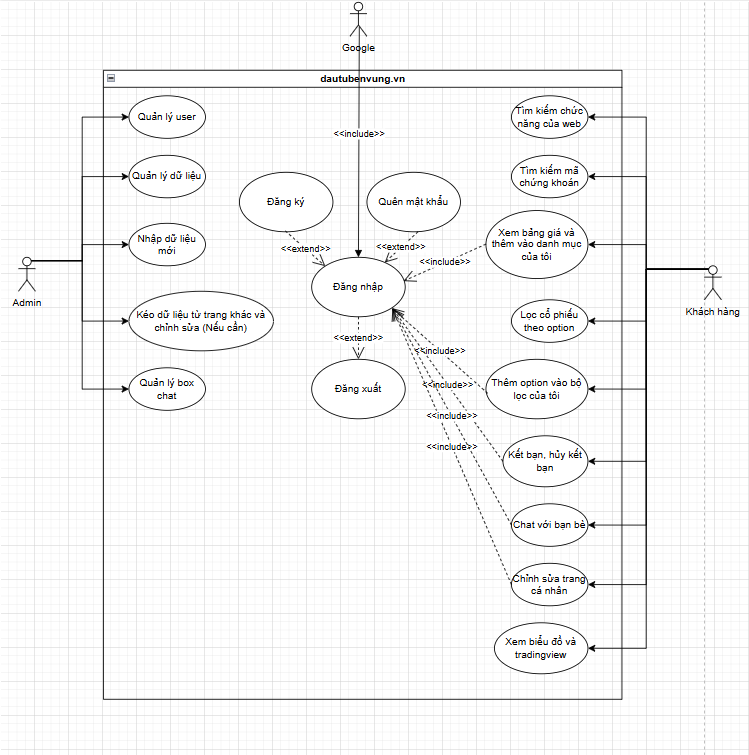
**3.1.2.1. Biểu đồ usercase**

Biểu đồ usercase thể hiện số lượng chức năng mà người dùng có thể tương tác trực tiếp.

* Đối với khách:

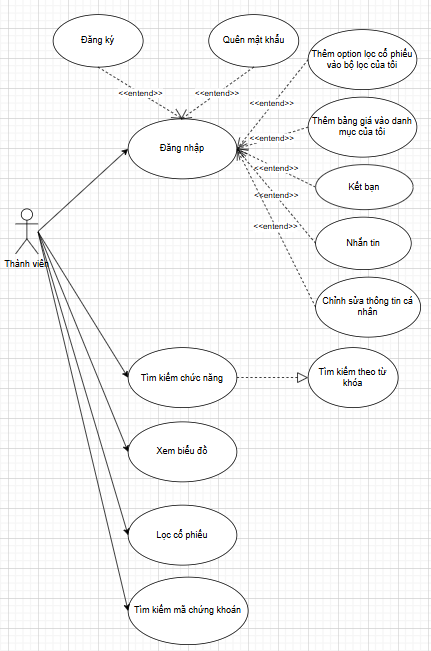
Khách có thể đăng ký tài khoản để đăng nhập vào chương trình và bắt đầu sử dụng.

a) Biểu đồ usecase chính



Hình 2. 2. Biểu đồ use-case chính

b) Biểu đồ usecase thành viên(khách hàng):



Hình 2. 3. Biểu đồ use-case thành viên

Usecase đăng ký:

|  |  |
| --- | --- |
| **Tên Usecase** | Đăng ký |
| **Tác nhân** | Thành viên |
| **Mô tả** | Cho phép tác nhân đăng ký vào hệ thống |
| **Tiền điều kiện** | Hệ thống đã được triển khai, cho phép người dùng truy cập và sử dụng tính năng đăng ký |
| **Luồng tin tức chính** | 1. Người dùng chọn chức năng đăng ký.  2. Form đăng ký xuất hiện yêu cầu nhập “họ tên”, “email”, “số điện thoại”, “password”.  3. Người dùng nhập thông tin theo yêu cầu.  3. Nhấn đăng ký:  - Nếu thông tin tài khoản đúng, chuyển đến trang chủ của web và đăng nhập bằng tài khoản đã đăng ký  - Nếu thông tin sai yêu cầu nhập lại.  4. UC kết thúc. |
| **Hậu điều kiện** | Người dùng có tài khoản để truy cập được vào hệ thống |

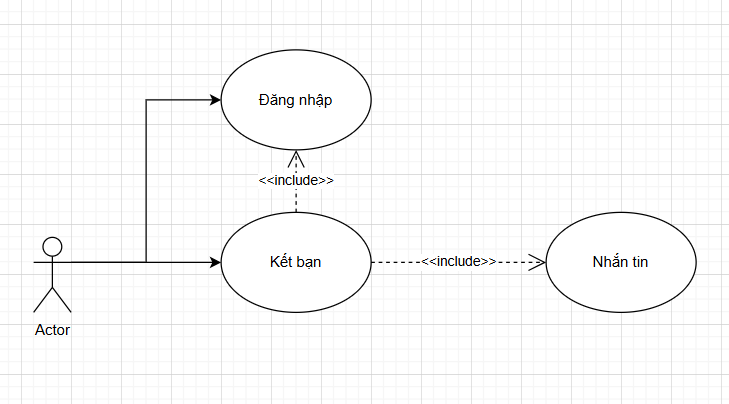
Bảng 1.1: Kịch bản usecase đăng ký

Usecase đăng nhập:

|  |  |
| --- | --- |
| **Tên Usecase** | Đăng nhập |
| **Tác nhân** | Quản trị viên, thành viên |
| **Mô tả** | Cho phép tác nhân đăng nhập vào hệ thống |
| **Tiền điều kiện** | Tác nhân đăng ký tài khoản thành công |
| **Luồng tin tức chính** | 1. Người dùng chọn chức năng đăng nhập.  2. Form đăng nhập xuất hiện yêu cầu nhập “số điện thoại”, “password”.  3. Nhấn đăng nhập:- Nếu nhập đúng, chuyển đến trang chủ của web.  - Nếu thông tin sai yêu cầu nhập lại.  4. UC kết thúc. |
| **Hậu điều kiện** | Người dùng truy cập được vào hệ thống |

Bảng 1.2: Kịch bản usecase đăng nhập

Usecase trò chuyện:

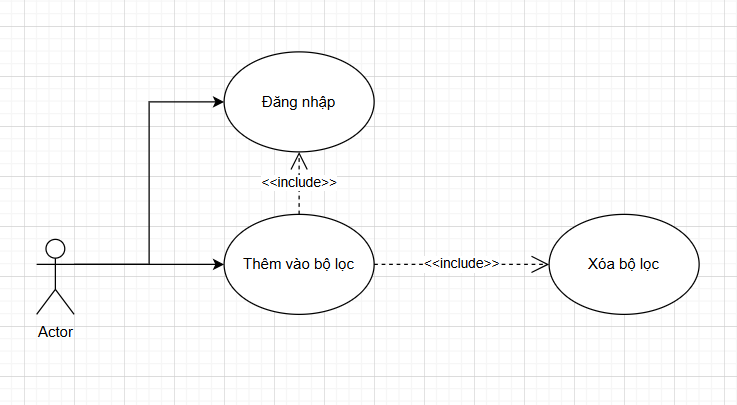


Hình 2. 4. Biểu đồ use-case trò chuyện

|  |  |
| --- | --- |
| **Tên Usecase** | Đăng nhập |
| **Tác nhân** | Thành viên |
| **Mô tả** | Cho phép thành viên trò chuyện trực tuyến trên hệ thống cùng các thành viên khác |
| **Tiền điều kiện** | - Các thành viên đã đăng nhập thành công và đã kết bạn với nhau.  - Kết nối mạng ổn định. |
| **Luồng tin tức chính** | 1. Thành viên chọn nhắn tin trong trang cá nhân của thành viên khác.  2. Hệ thống mở cửa sổ chat, hiển thị lịch sử trò chuyện (nếu có).  3. Thành viên nhập nội dung tin nhắn và nhấn nút gửi.  4. Tin nhắn được hiển thị ngay trong cửa sổ chat và gửi đến thành viên được nhắn trong thời gian thực.  5. Thành viên được nhắn tin nhận được tin nhắn, nhập nội dung phản hồi và gửi lại.  6. Cả hai tác nhân tiếp tục trao đổi tin nhắn đến khi kết thúc cuộc trò chuyện. |
| **Hậu điều kiện** | Người dùng truy cập được vào hệ thống |

Bảng 1.3: Kịch bản usecase trò chuyện

Usecase thêm vào bộ lọc của tôi:

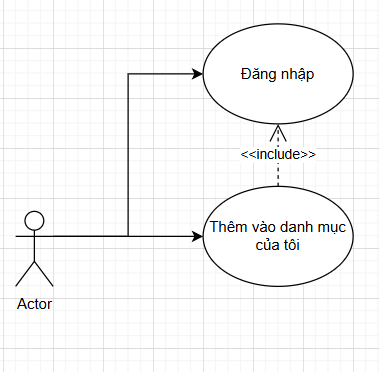


Hình 2. 5. Biểu đồ use-case thêm vào bộ lọc của tôi

|  |  |
| --- | --- |
| **Tên Usecase** | Đăng nhập |
| **Tác nhân** | Thành viên |
| **Mô tả** | Cho phép thành viên thêm các option lọc cổ phiếu vào bộ lọc của mình |
| **Tiền điều kiện** | - Các thành viên đã đăng nhập thành công.  - Kết nối mạng ổn định. |
| **Luồng tin tức chính** | 1. Thành viên chọn option bộ lọc.  2. Nhấn vào ngôi sao của option đã chọn. |
| **Hậu điều kiện** | Người dùng lưu được option vào bộ lọc của mình |

Bảng 1.4: Kịch bản thêm vào bộ lọc của tôi

Usecase thêm danh mục bảng giá vào danh mục của tôi:

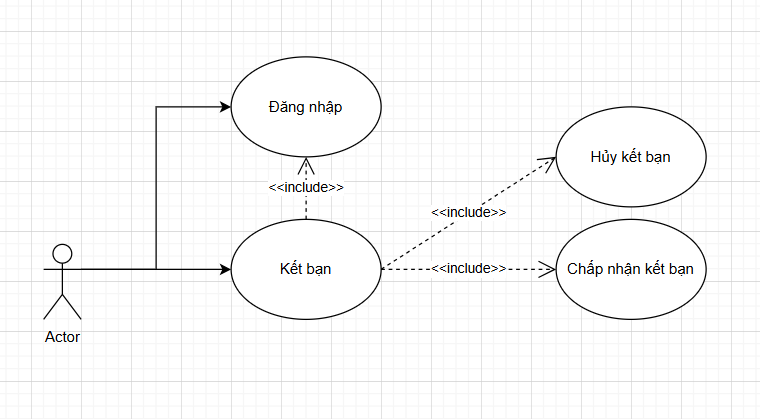


Hình 2. 6. Biểu đồ use-case thêm vào danh mục của tôi

|  |  |
| --- | --- |
| **Tên Usecase** | Đăng nhập |
| **Tác nhân** | Thành viên |
| **Mô tả** | Cho phép thành viên thêm bảng giá vào danh mục của tôi |
| **Tiền điều kiện** | - Các thành viên đã đăng nhập thành công.  - Kết nối mạng ổn định. |
| **Luồng tin tức chính** | 1. Thành viên điền tên danh mục và nhấn dấu + bên cạnh ô input.  2. Hệ thống sẽ thêm cho thành viên 1 danh mục.  3. Thành viên chỉ việc tìm kiếm mã trên ô tìm kiếm hệ thống sẽ tự thêm vào danh mục đó. |
| **Hậu điều kiện** | Người dùng thêm được vào danh mục bảng giá để dễ theo dõi. |

Bảng 1.5: Kịch bản usecase thêm danh mục của tôi

Usecase kết bạn:

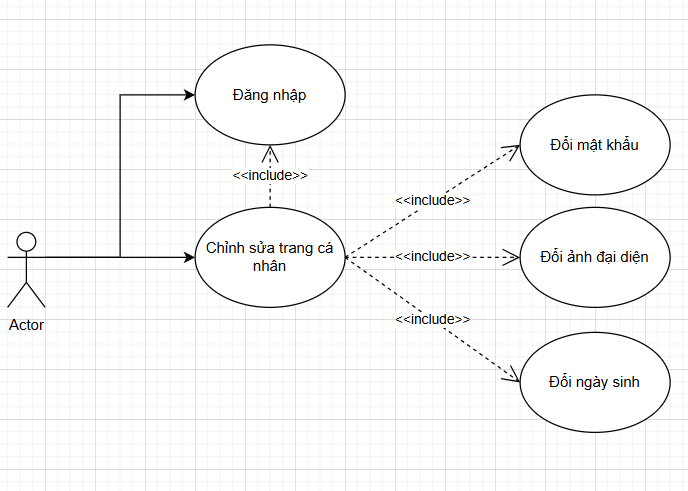


Hình 2. 7. Biểu đồ use-case kết bạn

|  |  |
| --- | --- |
| **Tên Usecase** | Đăng nhập |
| **Tác nhân** | Thành viên |
| **Mô tả** | Cho phép thành viên kết bạn với thành viên khác |
| **Tiền điều kiện** | - Các thành viên đã đăng nhập thành công.  - Kết nối mạng ổn định. |
| **Luồng tin tức chính** | 1. Thành viên 1 lấy link từ trang cá nhân và gửi qua nền tảng khác cho thành viên 2.  2. Thành viên 2 lấy link và gử kết bạn cho thành viên 1.  3. Thành viên 1 chấp nhận hoặc từ chối kết bạn của thành viên 2. |
| **Hậu điều kiện** | Người dùng thêm được bạn bè. |

Bảng 1.6: Kịch bản usecase thêm bạn bè

Usecase chỉnh sửa trang cá nhân:

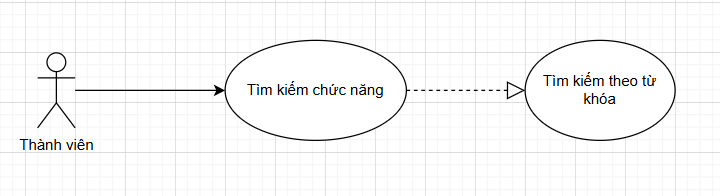


Hình 2. 8. Biểu đồ usecase sửa trang cá nhân

|  |  |
| --- | --- |
| **Tên Usecase** | Đăng nhập |
| **Tác nhân** | Thành viên |
| **Mô tả** | Cho phép thành viên chỉnh sửa trang cá nhân |
| **Tiền điều kiện** | - Các thành viên đã đăng nhập thành công.  - Kết nối mạng ổn định. |
| **Luồng tin tức chính** | 1. Thành viên vào trang cá nhân bấm chỉnh sửa.  2. Nhập ngày sinh, ảnh đại diện hoặc thay đổi mật khẩu. |
| **Hậu điều kiện** | Người dùng thay đổi được trang cá nhân của mình. |

Bảng 1.7: Kịch bản usecase chỉnh sửa trang cá nhân

Biểu đồ use-case thành viên tìm kiếm chức năng

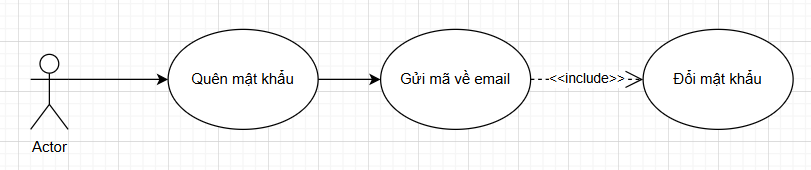


Hình 2. 9. Biểu đồ usecase tìm kiếm chức năng

|  |  |
| --- | --- |
| **Tên Usecase** | Thành viên tìm kiếm chức năng |
| **Tác nhân** | Thành viên |
| **Mô tả** | Thành viên tìm kiếm chức năng có trong hệ thống |
| **Tiền điều kiện** | Thành viên truy cập vào hệ thống |
| **Luồng sự kiện chính** | 1. Form tìm kiếm xuất hiện  2. Click vào form tìm kiếm sẽ hiển thị các option chức năng người dùng có thể chọn hoặc tìm kiếm  3. Trả về kết quả  a. Nếu chức năng tồn tại, sẽ chuyển sang chức năng đó  b. Nếu chức năng không tồn tại, sẽ trả về không có gì |
| **Hậu điều kiện** | Khách hàng tìm thấy chức năng mong muốn thành công |

Bảng 1.8: Kịch bản usecase tìm kiếm chức năng

Biểu đồ use-case quên mật khẩu

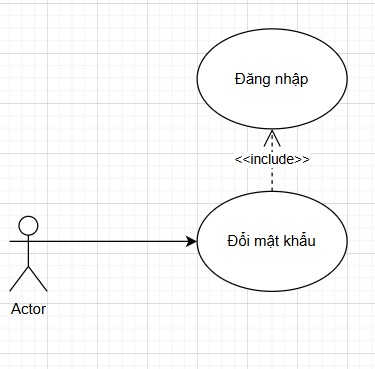


Hình 2. 10. Biểu đồ usecase quên mật khẩu

|  |  |
| --- | --- |
| **Tên Usecase** | Thành viên quên mật khẩu |
| **Tác nhân** | Thành viên |
| **Mô tả** | Thành viên quên mật khẩu và muốn lấy lại mật khẩu mới |
| **Tiền điều kiện** | Thành viên truy cập vào hệ thống |
| **Luồng sự kiện chính** | 1. Nhấn vào quên mật khẩu ở trang đăng nhập  2. Nhập email để nhận mã đổi mật khẩu rồi nhấn gửi mã  3. Nhập mã để chuyển sang trang đổi mật khẩu  a. Nếu sai thì không cho click chuyển tiếp  b. Nếu đúng thì cho chuyển tiếp và đổi mật khẩu |
| **Hậu điều kiện** | Thành viên đổi lại được mật khẩu đã quên |

Bảng 1.7: Kịch bản usecase quên mật khẩu

Biểu đồ use-case đổi mật khẩu



Hình 2. 11. Biểu đồ usecase đổi

|  |  |
| --- | --- |
| **Tên Usecase** | Đổi mật khẩu |
| **Tác nhân** | Thành viên |
| **Mô tả** | Thành viên muốn đổi mật khẩu mới |
| **Tiền điều kiện** | Thành viên truy cập vào hệ thống |
| **Luồng sự kiện chính** | 1. Truy cập trang cá nhân và nhấn vào chỉnh sửa  2. Sau đó nhấn tiếp vào đổi mật khẩu  3. Nhập mật khẩu cũ và nhập 2 lần mật khẩu muốn đổi  a. Nếu sai mật khẩu cũ thì hiện thông báo và nhập lại  b. Nếu nhập 2 lần mật khẩu muốn đổi không trùng khớp thì thông báo và nhập lại  c. Nếu đủ điều kiện thì được phép đổi mật khẩu |
| **Hậu điều kiện** | Thành viên đổi mật khẩu thành công |

Bảng 1.8: Kịch bản usecase quên mật khẩu

## **3.2. Các bảng dữ liệu chính**

* Bảng USERS:

Ý nghĩa: đặc trưng cho từng user, ghi lại thông tin của user, thông tin thiết bị mà user dùng đăng kí. Khóa chính là ID dùng phân biệt user với nhau.

|  |  |  |  |
| --- | --- | --- | --- |
| **Cột** | **Kiểu dữ liệu** | **Khóa chính** | **Ý nghĩa** |
| userID | int | X | Đặc trưng cho từng user, phân biệt user với nhau |
| email | varchar |  | Email đăng nhập |
| phone\_number | varchar |  | Số điện thoại |
| name | varchar |  | Tên của user |
| password | varchar |  | Mật khẩu của user |
| createdOn | datetime |  | Ngày tạo |
| isOnline | tinyint |  | Hoạt động hoặc không hoạt động |
| image | varchar |  | Ảnh đại diện |
| birthdate | varchar |  | Ngày sinh |

Bảng 1.9. Bảng user và các thuộc tính

* Bảng bộ lọc của tôi:

|  |  |  |  |
| --- | --- | --- | --- |
| **Cột** | **Kiểu dữ liệu** | **Khóa chính** | **Ý nghĩa** |
| id | int | X | Đặc trưng cho từng bộ lọc |
| userId | int |  | Đặc trưng cho từng user |
| label | varchar |  | Tên của option |
| number | tinyint |  | Có thể chọn nhiều lần hoặc không |

Bảng 1.10. Bảng bộ lọc của tôi

* Bảng mối quan hệ:

|  |  |  |  |
| --- | --- | --- | --- |
| **Cột** | **Kiểu dữ liệu** | **Khóa chính** | **Ý nghĩa** |
| id | int | X | Đặc trưng cho từng mỗi quan hệ |
| user1\_id | int |  | Đặc trưng cho từng user |
| user2\_id | int |  | Đặc trưng cho từng user |
| status | enum |  | Trạng thái bạn bè |
| action\_user\_id | int |  | Ai là người yêu cầu |
| created\_at | timestamp |  | Ngày tạo lời mời |
| updated\_at | timestamp |  | Ngày cập nhật |

Bảng 1.11. Bảng mối quan hệ

* Bảng danh mục của tôi trang bảng giá:

|  |  |  |  |
| --- | --- | --- | --- |
| **Cột** | **Kiểu dữ liệu** | **Khóa chính** | **Ý nghĩa** |
| id | int | X | Đặc trưng cho từng danh mục |
| iboardID | int |  | Đặc trưng của danh mục đã tạo |
| list\_symbol | varchar |  | Danh sách các mã của danh mục của tôi đã thêm |

Bảng 1.12. Bảng vốn đầu tư tỉnh thành

* Bảng các mã chứng khoán:

|  |  |  |  |
| --- | --- | --- | --- |
| **Cột** | **Kiểu dữ liệu** | **Khóa chính** | **Ý nghĩa** |
| comGroupCode | varchar |  | Nhóm mã |
| comTypeCode | varchar |  | Tên mã |
| icbCode | varchar |  | Loại sàn |
| organCode | varchar |  | Mã cơ quan |
| organName | varchar |  | Tên cơ quan |
| organShortName | varchar |  | Tên viết tắt |
| organTypeCode | varchar |  | Tên đầy đủ của cơ quan |
| ticker | varchar |  | Tên cơ quan |

Bảng 1.13. Bảng các mã chứng khoán

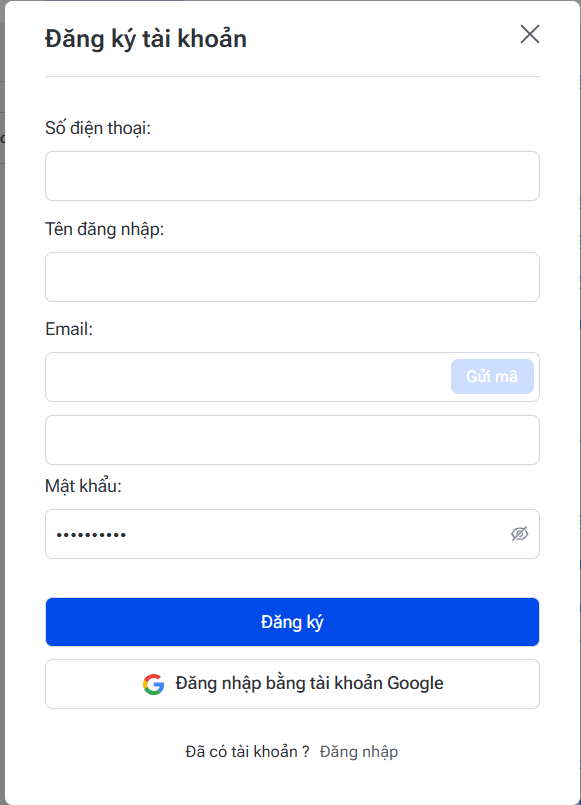
# **CHƯƠNG 4: XÂY DỰNG WEB**

## **4.1. Xây dựng giao diện**

### **4.1.1. Một số giao diện chính**

**4.1.1.1. Giao diện đăng ký**

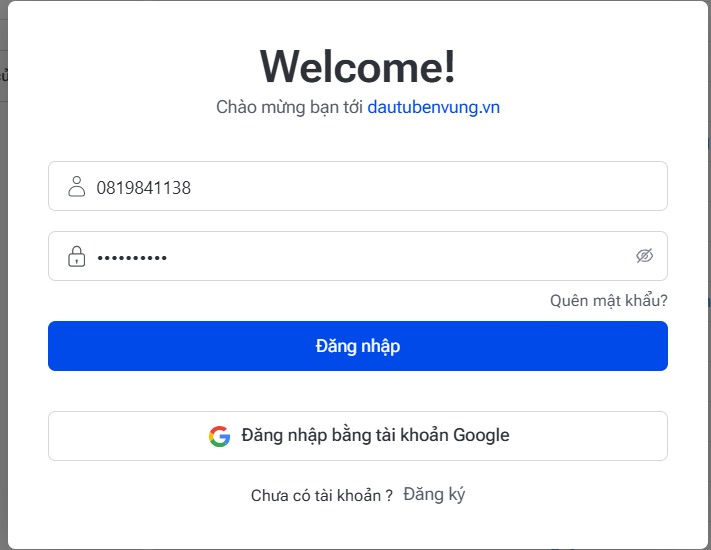
Chức năng đăng ký, đăng nhập cho phép người dùng đăng ký làm thành viên của trang web. Nếu người dùng chưa có tài khoản đăng nhập thì chọn “Đăng ký” nhập các trường thông tin như số điện thoại (thông tin bắt buộc), mật khẩu (thông tin bắt buộc), tên đăng nhập (thông tin bắt buộc), email (thông tin bắt buộc), mã gửi về email (thông tin bắt buộc). Khi đã điền đầy đủ thông tin người dùng chọn đăng ký để hoàn tất.

****

Hình 3. . Giao diện đăng ký

**4.1.1.2. Giao diện đăng nhập**

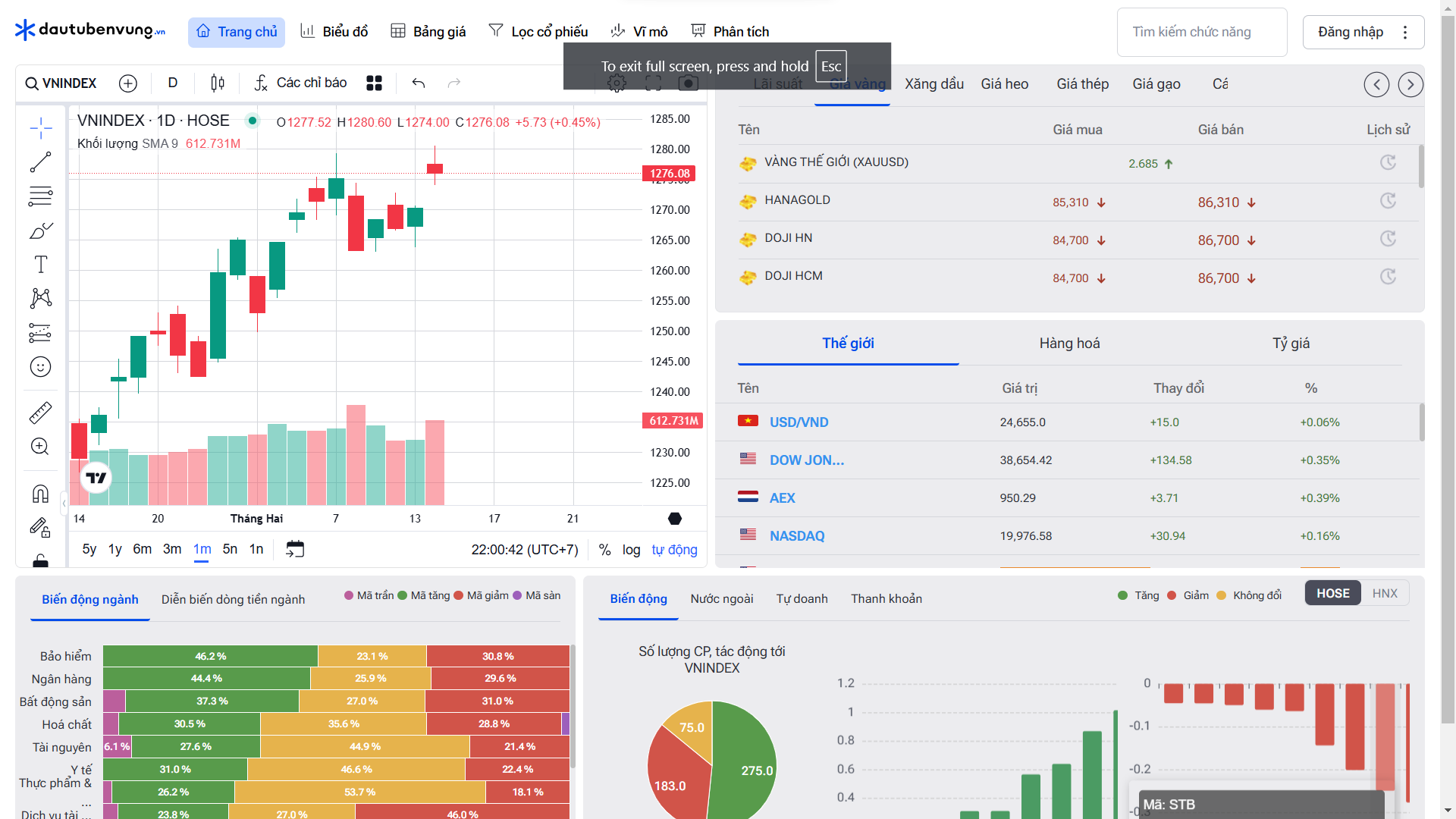
Người dùng cần đăng nhập thông tin tài khoản, mật khẩu (khi đã có tài khoản đăng nhập) để truy cập vào web, nếu chưa có tài khoản người dùng chọn đăng ký tài khoản (chi tiết ở bước màn hình đăng ký).



Hình 3. . Giao diện đăng nhập

**4.1.1.3. Giao diện trang chính (Home)**

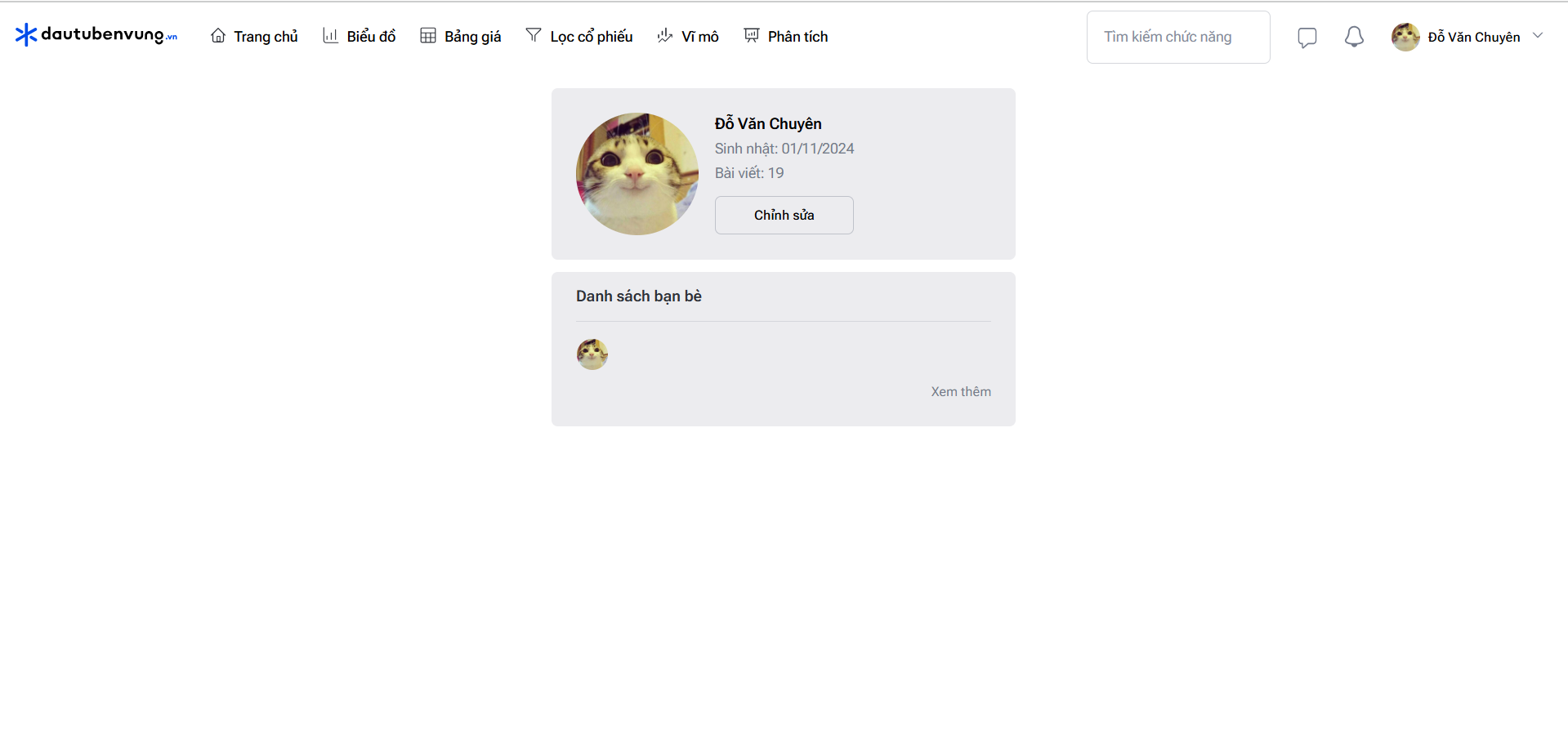
Giao diện màn hình chính trong web trong đó gồm các trang : trang chủ, biểu đồ, bảng giá, lọc cổ phiếu, phân tích.

****

Hình 3. . Giao diện trang chính

**4.1.1.4. Giao diện thông tin người dùng**

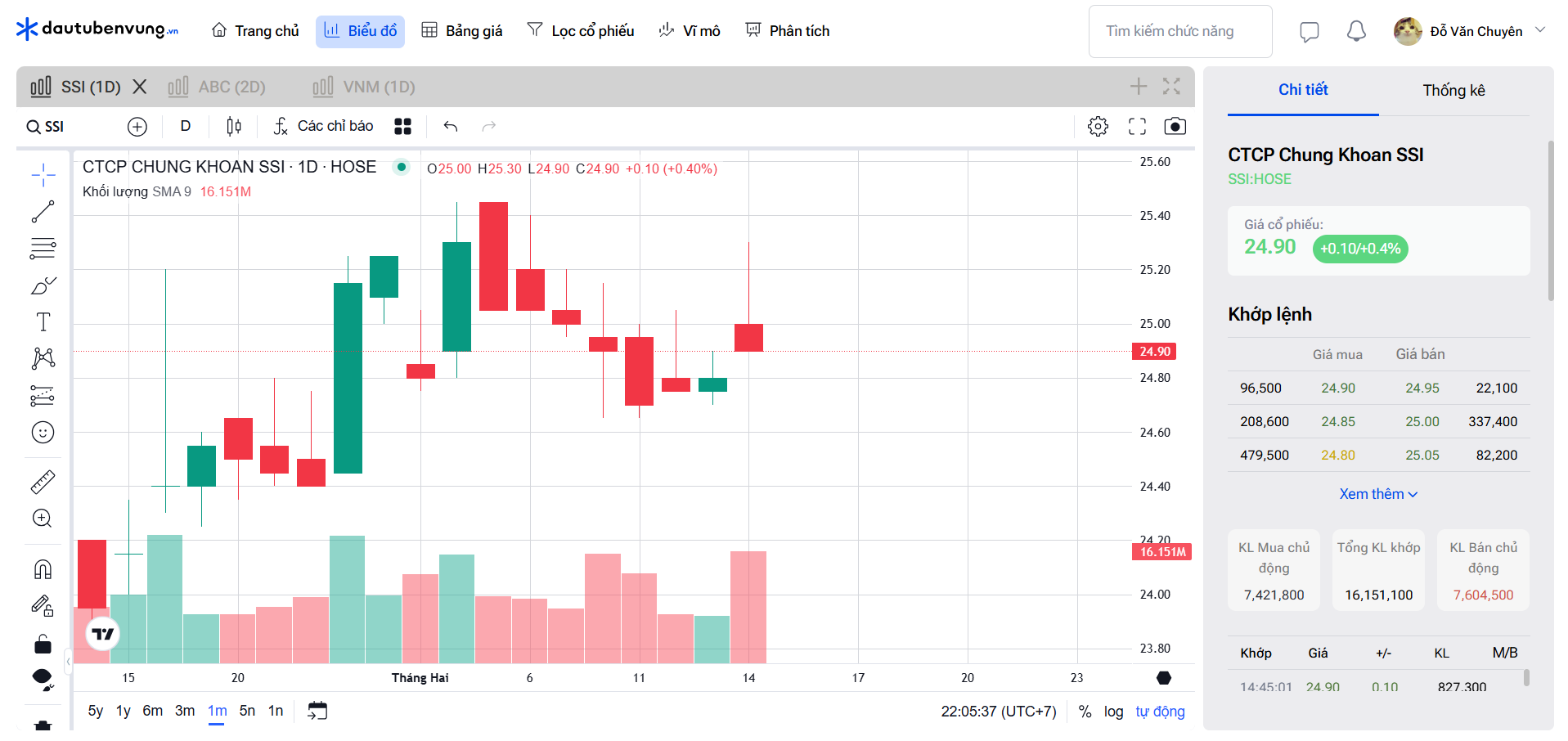
Màn hình thông tin người dùng quản lý các thông tin của người dùng như tên, sinh nhật, mật khẩu,… Người dùng có thể thay đổi các thông tin về ảnh, mật khẩu, sinh nhật bên trong màn hình quản lý thông tin.

****

Hình 3. . Giao diện thông tin người dùng

**4.1.1.5. Giao diện trang biểu đồ**

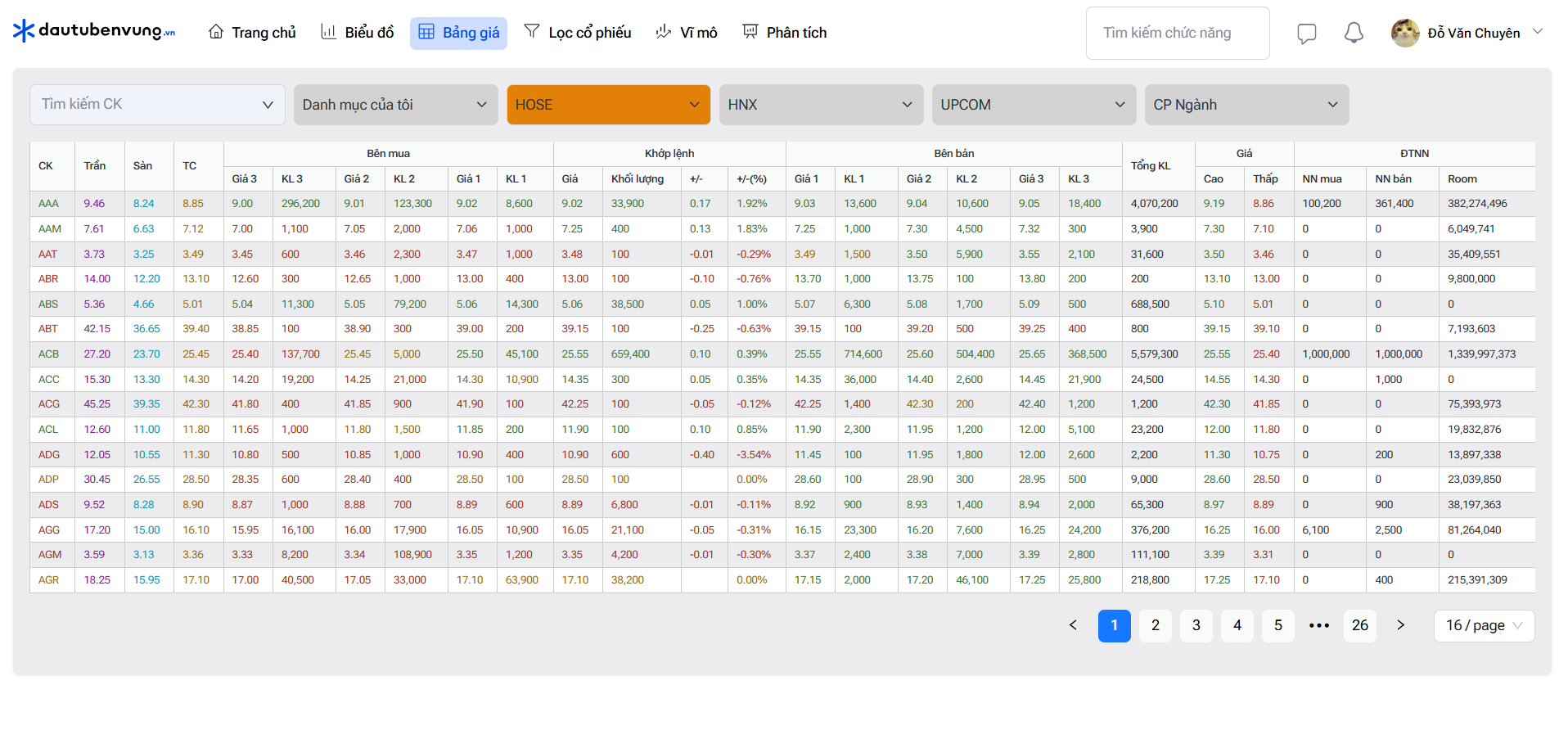
Trang biểu đồ gồm trading view của tất cả các mã chứng khoán và bên phải là chỉ số giá tăng giảm của cố phiếu cùng với đó là thống kê tham chiếu…

****

Hình 3. . Giao diện trang biểu đồ chi tiết

**4.1.1.6. Giao diện bảng giá**

Danh sách giá các cổ phiếu trong phiên cũng như là kết thúc phiên. Gồm 3 sàn : HOSE, HNX, UPCOM

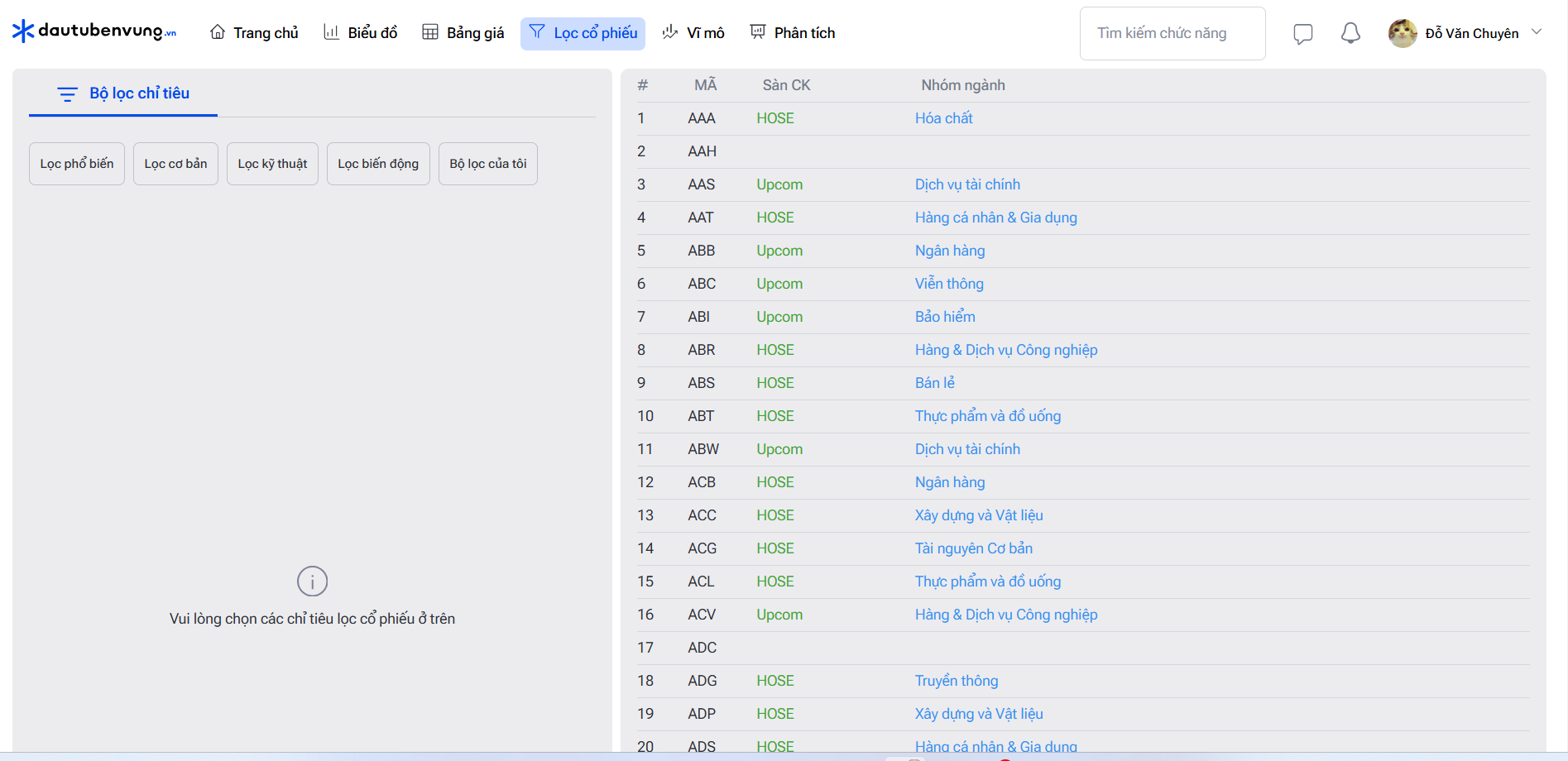
****

Hình 3. 6. Giao diện bảng giá

**4.1.1.7. Giao diện lọc cổ phiếu**

Lọc cổ phiếu là quá trình sử dụng các tiêu chí cụ thể để chọn ra những cổ phiếu phù hợp từ hàng ngàn cổ phiếu trên thị trường. Mục đích của việc lọc cổ phiếu bao gồm:

* **Tiết kiệm thời gian và công sức:** Thay vì phải phân tích từng cổ phiếu một, việc lọc giúp nhà đầu tư nhanh chóng thu hẹp danh sách các cổ phiếu tiềm năng dựa trên các tiêu chí đã định.
* **Xác định cơ hội đầu tư tốt:** Bằng cách áp dụng các chỉ số tài chính như tỷ lệ P/E, lợi nhuận trên vốn (ROE), tăng trưởng doanh thu, nhà đầu tư có thể tìm ra những cổ phiếu có tiềm năng tăng giá hoặc cổ tức hấp dẫn.
* **Quản lý rủi ro:** Lọc cổ phiếu theo các tiêu chí về độ ổn định tài chính, nợ thấp, hoặc các yếu tố khác giúp giảm thiểu rủi ro đầu tư.
* **Định hướng chiến lược đầu tư:** Nhà đầu tư có thể sử dụng các bộ lọc khác nhau tùy thuộc vào chiến lược đầu tư của mình, chẳng hạn như đầu tư theo giá trị, tăng trưởng, hay theo ngành cụ thể.
* **Phân tích thị trường nhanh chóng:** Việc lọc cổ phiếu giúp nhà đầu tư nắm bắt xu hướng thị trường và tìm ra những cổ phiếu phù hợp với điều kiện thị trường hiện tại.
* **Tùy chỉnh theo mục tiêu cá nhân:** Nhà đầu tư có thể thiết lập các tiêu chí riêng phù hợp với mục tiêu tài chính và khẩu vị rủi ro của mình, từ đó lựa chọn được các cổ phiếu phù hợp nhất.

****

Hình 3. 7. Giao diện lọc cổ phiếu

**4.1.1.8. Giao diện vĩ mô**

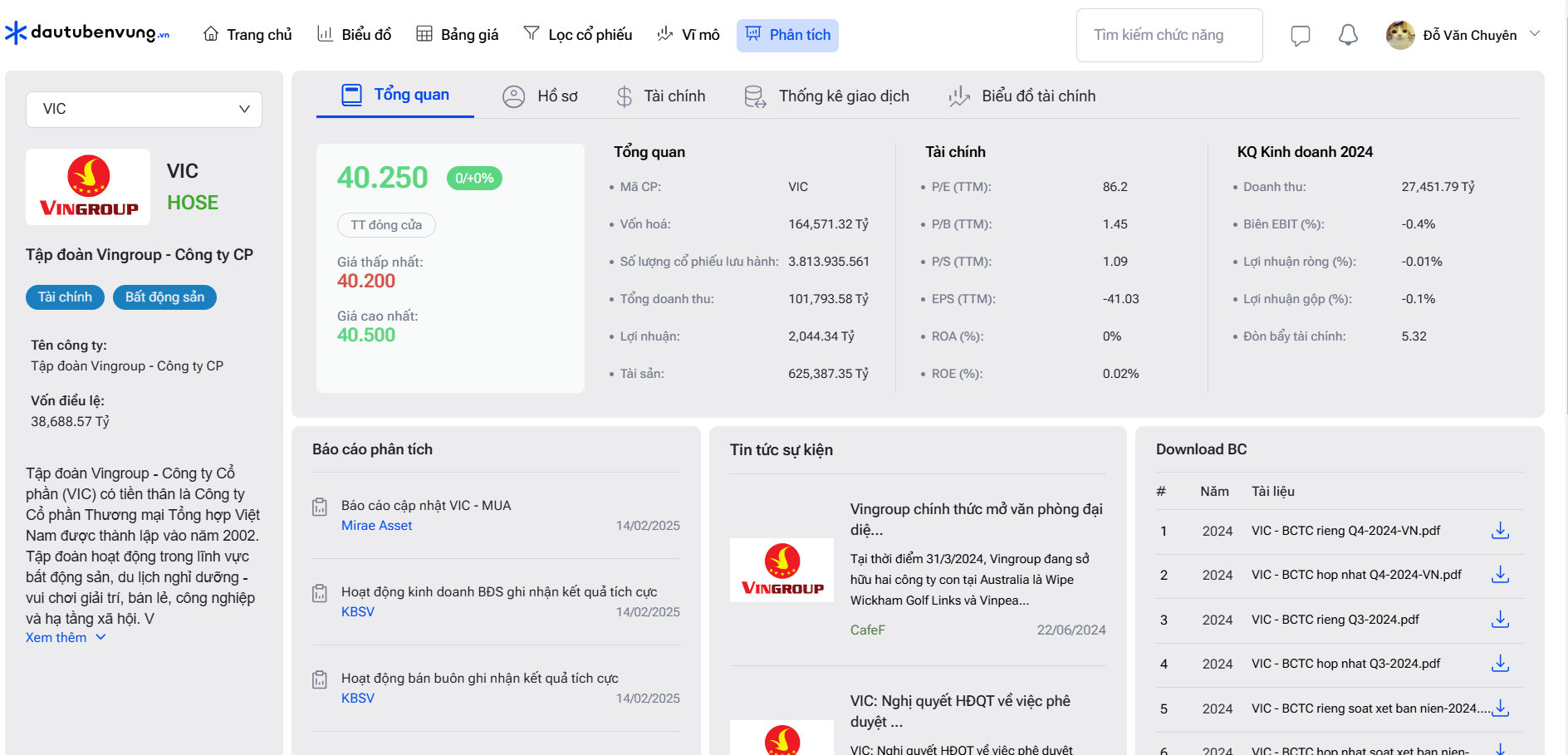
Phân tích vĩ mô là việc xem xét các yếu tố kinh tế tổng thể ảnh hưởng đến thị trường chứng khoán và các ngành cụ thể. Nhà đầu tư sử dụng phân tích vĩ mô để đánh giá môi trường kinh tế trước khi chọn ngành hoặc cổ phiếu để đầu tư.



Hình 3. 8. Giao diện vĩ mô

**4.1.1.8. Giao diện phân tích**

Phân tích dùng để tìm kiếm các mã cổ phiếu trên sàn chứng khoán và cùng với đó là xem cổ đông, báo cáo tài chính, các công ty con và khái quát công ty.



Hình 3. 9. Giao diện bảng giá

**4.1.1.9. Giao diện quên mật khẩu**

Khi thành viên/khách hàng quên mật khẩu thì sẽ lấy lại bằng cách gửi mã về email xác nhận mã và đổi mật khẩu.

****

Hình 3. 10. Giao diện quên mật khẩu

### **4.1.2. Danh sách các màn hình**

* Client

Phần tương tác với người dùng. Người dùng phải đăng ký và đăng nhập vào để có thể sử dụng được ứng dụng.

* Tuỳ chỉnh trading view và các mã để phân tích thị trường.
* Kết nối internet luôn được kiểm tra để đảm bảo hoạt động ổn định.
* Server

Server có các chức năng chính như quản lý người dùng, các dữ liệu của các mã…

* Danh sách các màn hình

|  |  |  |
| --- | --- | --- |
| STT | Tên màn hình | Ý nghĩa/Ghi chú |
| 1 | Trang chủ | Tổng quan các mã, ngành nổi bật |
| 2 | Trang đăng nhập | Gồm các chức năng về đăng nhập, đăng ký. |
| 3 | Trang đăng ký | Tạo hồ sơ thông tin người dùng |
| 4 | Trang quản lý thông tin người dùng | Hiển thị thông tin cá nhân của người dùng cùng với các tùy chỉnh. |
| 5 | Trang biểu đồ | Gồm biểu đồ trading view và chi tiết, tổng quan của mã tương ứng. |
| 6 | Trang lọc cổ phiếu | Sử dụng các tiêu chí cụ thể để chọn ra những cổ phiếu phù hợp từ hàng ngàn cổ phiếu trên thị trường. |
| 7 | Trang vĩ mô | Để xem xét các yếu tố kinh tế tổng thể ảnh hưởng đến thị trường chứng khoán và các ngành cụ thể. Nhà đầu tư sử dụng phân tích vĩ mô để đánh giá môi trường kinh tế trước khi chọn ngành hoặc cổ phiếu để đầu tư. |
| 8 | Trang phân tích | Hiển thị chi tiết các thông tin của từng mã chứng khoán |

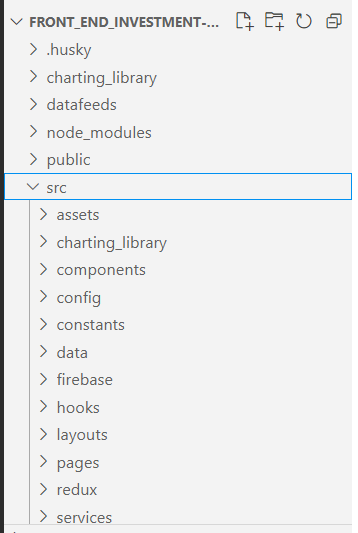
Bảng 2.1. Bảng danh sách các màn chính

## **4.2. Xây dựng web phân tích chứng khoán**

### **4.2.1. Cấu trúc thư mục dự án**

Cấu trúc thư mục Reactjs được chia thành các thư mục, cụ thể:

* **node\_modules/**: Chứa các gói phụ thuộc.
* **public/**: Chứa tệp HTML gốc và tài sản tĩnh.
* **src/**: Chứa mã nguồn của ứng dụng.
  + **components/**: Chứa các thành phần React tái sử dụng.
  + **App.js**: Thành phần gốc của ứng dụng.
  + **index.js**: Điểm vào của ứng dụng.
* **package.json**: Quản lý các phụ thuộc và script.
* **README.md**: Tài liệu mô tả dự án.



Hình 3. 11. Cấu trúc thư mục dự án

### **4.2.2. Thư mục các trang của web**

Thư mục này để tổng hợp các trang chính của web chứng khoán có đuôi là tsx và ts.

* **.tsx**: Đây là tệp giao diện chính cho từng trang của web gồm các chức năng và khung giao diện.
* **.ts**: Được sử dụng trong css làm cho giao diện thêm sinh động hơn

A screenshot of a computer

Description automatically generated

Hình 3. 12. Thư mục chứa các giao diện và chức năng, nội dung.

## **4.3. Xây dựng backend**

**4.3.1. Thư mục backend**

Cấu trúc thư mục backend có file cụ thể như sau:

A screenshot of a computer

Description automatically generated

Hình 3. 13. Thư mục back-end chính.

**4.3.1. Cấu trúc thư mục backend**

* **controllers/**: Chứa các file logic điều khiển xử lý yêu cầu từ client.
* **models/**: Chứa các file định nghĩa mô hình cơ sở dữ liệu.
* **routes/**: Chứa các định nghĩa API endpoint.
* **middlewares/**: Xử lý các chức năng trung gian như xác thực, logging.
* **config/**: Cấu hình ứng dụng như kết nối cơ sở dữ liệu.
* **utils/**: Chứa các tiện ích chung như xử lý lỗi, gửi email.
* **app.js**: File khởi tạo ứng dụng.
* **server.js**: Điểm bắt đầu của server.

**4.3.2. Công nghệ sử dụng**

* **Ngôn ngữ**: JavaScript/TypeScript.
* **Framework**: Express.js.
* **Cơ sở dữ liệu**: MySQL.
* **Thư viện hỗ trợ**:
  + JWT (JSON Web Token) để xác thực.
  + bcrypt để mã hóa mật khẩu.
  + dotenv để quản lý biến môi trường.

**4.3.3. Các API chính**

1. **Quản lý người dùng**:
   * Đăng ký (POST /register).
   * Đăng nhập (POST / login).
   * Cập nhật thông tin người dùng (PUT /update-user).
2. **Quản lý mã chứng khoán**:
   * Lấy danh sách mã chứng khoán (GET /info-company).
3. **Báo cáo tài chính**:
   * Tải báo cáo tài chính (GET /info-company/symbol=AAA).

**4.3.4. Kiểm thử**

* **Công cụ kiểm thử**: Postman, Swagger.
* **Loại kiểm thử**:
  + Kiểm thử chức năng API.

**4.3.5. Triển khai**

* **Môi trường**:
  + Dev: Chạy trên localhost.

## **4.4. Phân tích code các trang chính**

### **4.4.1. Front-end**

a. Header.

****

Hình 3. 14. Header.

Đây là phần tìm kiếm của header:

* Dùng useEffect để lấy list danh sách chức năng ban đầu của tìm kiếm khi người dùng mount component.
* Sử dụng filter để lọc theo tên chức năng tìm kiếm và in hoa nó lên.

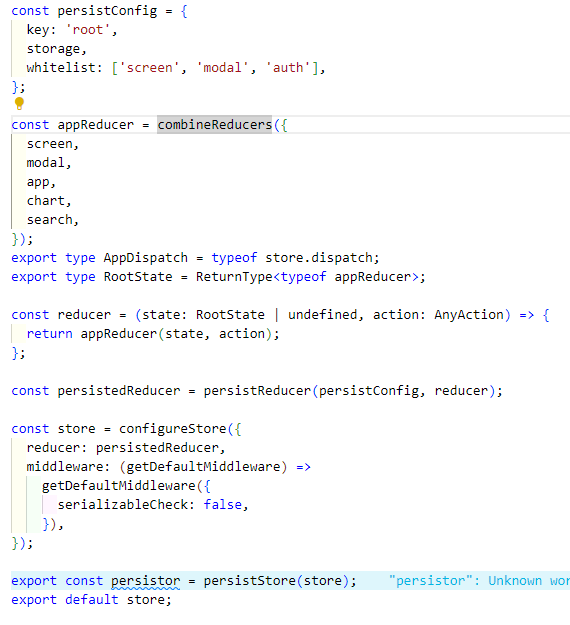
b. File App.tsx.



Hình 3. 15. Các trang nhỏ của web

Dùng react-router-dom để link các trang với nhau để người dung có thể chuyển trang trên web.

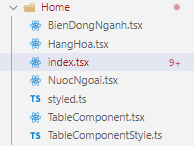
c. Redux.



Hình 3. 16. Redux

Sử dụng redux để lưu các giá trị trong web. Những giá trị này có thể lấy ra và dung trong bất khì component nào kể cả component không có liên quan đến component khởi tạo giá trị

d. Trang Home.



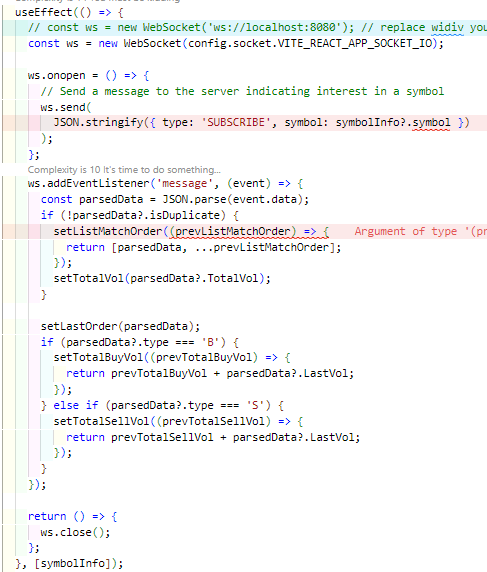
Hình 3. 17. Các component của file home.

Chia nhỏ các component ra để dễ quản lý và sẽ tạo cho web hoạt động mượt mà hơn.

  
Hình 3. 18. Truyền các prop qua các component con

Truyền prop vào các component con để nhận dữ liệu từ cha xuống.

e. Trang biểu đồ.



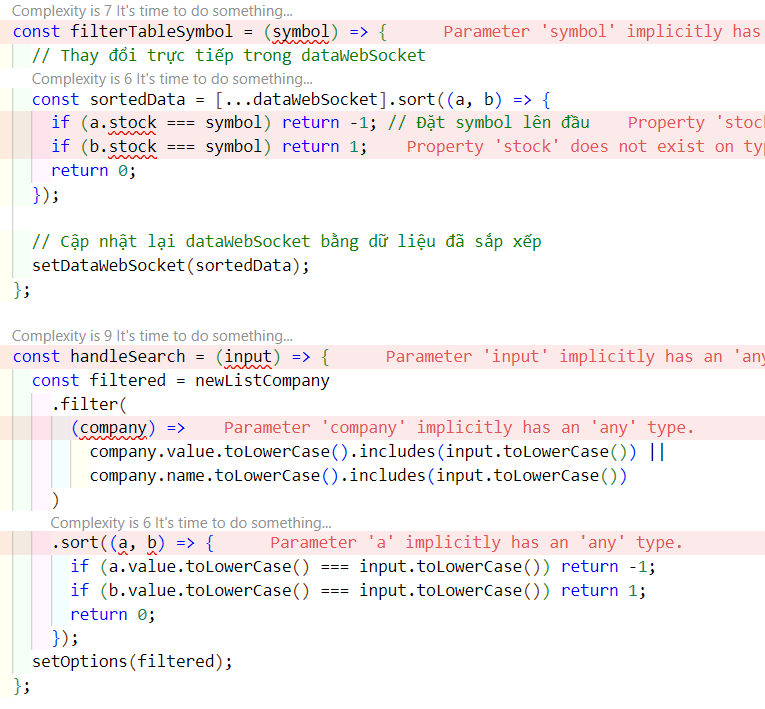
Hình 3. 19. Socket(real-time)

Dùng useEffect để call api web socket lấy data realtime để hiển thị khi giá cổ phiếu nhảy.

d. Trang bảng giá.

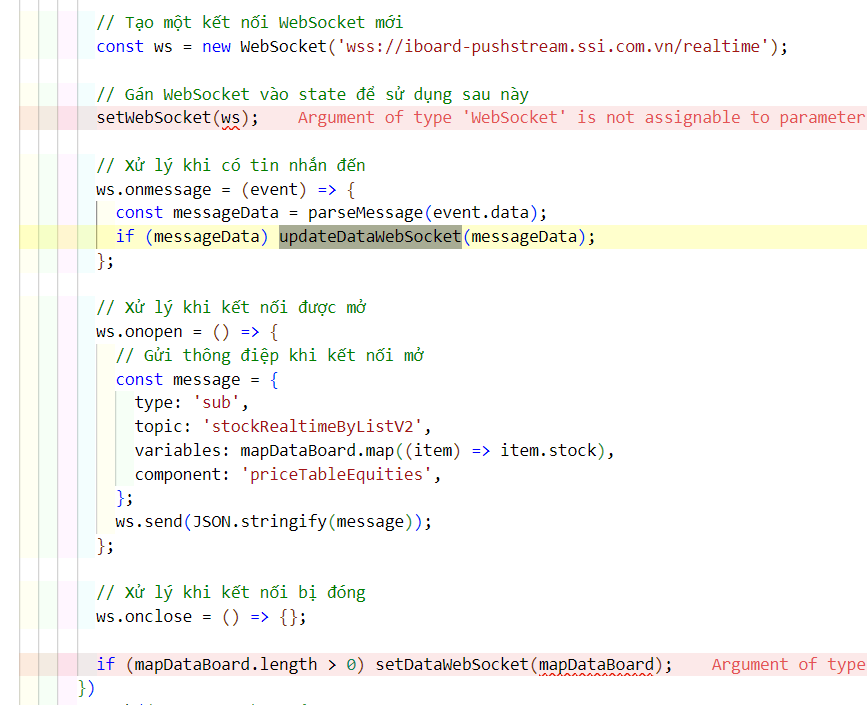
  
Hình 3. 20. Một trong các hook dùng để call api.

Đầu tiên cần get hết các dữ liệu của bảng sau đó dung filter để tìm kiếm các mã.



Hình 3. 21. Chức năng tìm kiếm các mã cổ phiếu.

Sau đó lắng nghe web socket



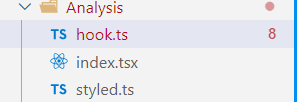
Hình 3. 22. Socket.

e. Trang lọc cổ phiếu.



Hình 3. 23. Call api lọc cổ phiếu.  
Đầu tiên call api để lấy tất cả danh sách mã cổ phiếu hiện tại, sau đó sẽ lọc theo số liệu mà người dùng yêu cầu.

f. Trang phân tích.



Hình 3. 24. Chia component.

Đầu tiên tạo 1 file hook.ts để custom hook rồi gọi những biến cần thiết ở index.tsx



Hình 3. 25. Call các api của màn phân tích.

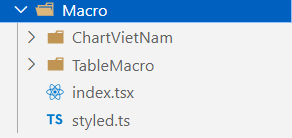
khi người dùng mount vào component thì gọi tất cả các api cần thiết rồi trả dữ liệu ra giao diện web.



Hình 3. 26. Custom và truyền các option để được biểu đồ.

Đây là 1 trong những biểu đồ chính của phần biểu đồ tài chính.

g. Trang vĩ mô.

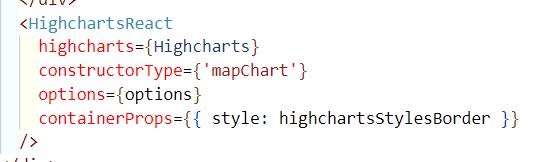


Hình 3. 27. Chia component trang vĩ mô.

Chia file để dễ quản lý các component và có thể dùng lại các component đó.



Hình 3. 28. Tạo option để thể hiện bản đồ Việt Nam.



Hình 3. 29. Hiển thị biểu đồ.

Sử dụng biểu đồ Hightchart để hiển thị bản đồ Việt Nam.

### **4.4.1. Back-end.**

Đầu tiên chúng ta setup port api cho backend



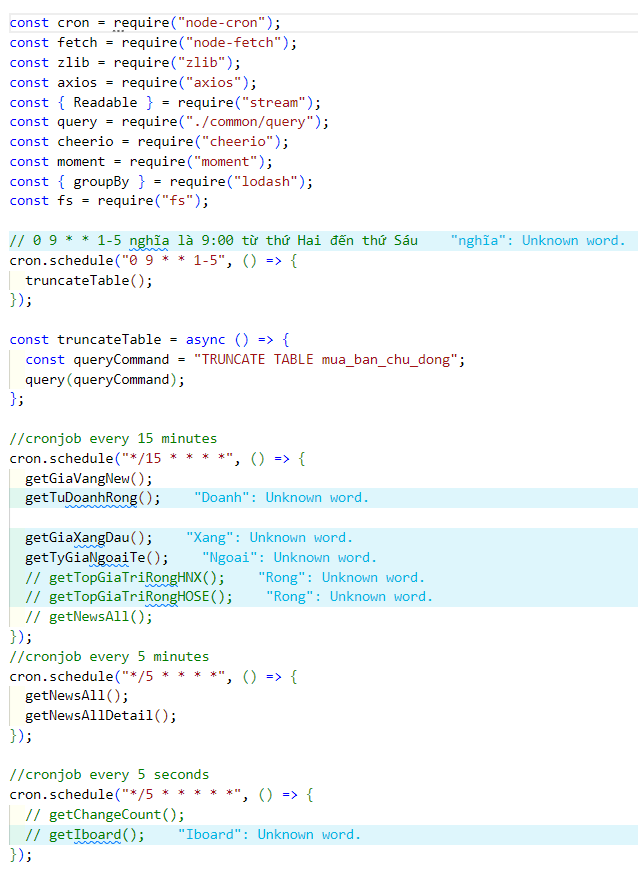
Hình 3. 30. Setup port.



Hình 3. 31. Kết nối SQL.

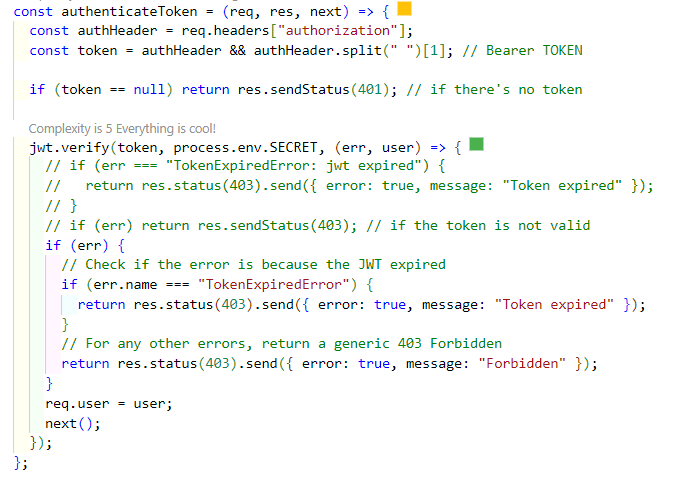
và kết nối với database để lấy dữ liệu cần thiết

tiếp theo là file cronJob.js



Hình 3. 32. Kéo data.

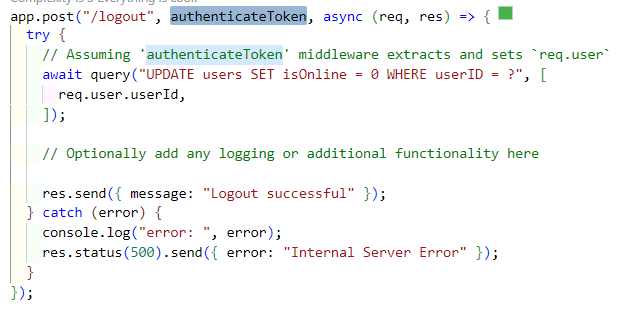
File này sẽ cho ta thấy được tất cả các dữ liệu từ các nguồn khác nhau lấy về database.



Hình 3. 33. Middle ware.

authenticateToken là 1 middleware để khi mà những api cần người dung đăng nhập tài khoản thì chúng ta sẽ thêm nó vào.

Ví dụ như logout:



Hình 3. 34. Login api.

# **KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN**

**1. Kết luận**

Đồ án "Xây dựng web phân tích thị trường chứng khoán sử dụng ReactJS và NodeJS" đã hoàn thành với các mục tiêu đề ra, gồm:

* Xây dựng một hệ thống phân tích thị trường chứng khoán hiện đại, hỗ trợ các nhà đầu tư cá nhân và tổ chức trong việc tiếp cận thông tin và ra quyết định.
* Ứng dụng các công nghệ mới như ReactJS ở frontend và NodeJS ở backend, mang lại hiệu suất cao, giao diện trực quan, thân thiện với người dùng.
* Tích hợp các chức năng như hiển thị biểu đồ giao dịch, quản lý mã chứng khoán, và phân tích dữ liệu, đáp ứng nhu cầu thực tiễn trong lĩnh vực tài chính.

Qua quá trình thực hiện, em đã học hỏi và củng cố các kiến thức chuyên môn về lập trình web, quản lý cơ sở dữ liệu, cũng như các kỹ năng triển khai và kiểm thử hệ thống. Mặc dù còn một số hạn chế cần khắc phục, đồ án đã hoàn thành những chức năng cơ bản và sẵn sàng để mở rộng trong tương lai.

**2. Hướng phát triển**

Để nâng cao chất lượng và mở rộng tính năng, hệ thống có thể phát triển thêm các nội dung sau:

* **Cải thiện giao diện và trải nghiệm người dùng (UI/UX):**
  + Sử dụng các thư viện thiết kế hiện đại hơn như TailwindCSS, Material UI.
  + Tối ưu giao diện trên các thiết bị di động để đảm bảo tính tương thích (responsive design).
* **Mở rộng chức năng hệ thống:**
  + Thêm tính năng giao dịch chứng khoán trực tiếp (mua/bán cổ phiếu ảo).
  + Cung cấp dự đoán thị trường dựa trên dữ liệu lịch sử và trí tuệ nhân tạo (AI).
  + Phát triển chức năng thông báo real-time về biến động giá cổ phiếu.
* **Tăng cường hiệu suất và bảo mật:**
  + Áp dụng caching (Redis) để tăng tốc độ xử lý dữ liệu.
  + Tích hợp cơ chế xác thực đa yếu tố (MFA) để bảo vệ tài khoản người dùng.
  + Tối ưu hóa cơ sở dữ liệu, giảm thời gian truy vấn.
* **Hỗ trợ ngôn ngữ và thị trường quốc tế:**
  + Đa ngôn ngữ (multilingual) để mở rộng đối tượng sử dụng.
  + Kết nối và tích hợp dữ liệu từ các thị trường chứng khoán lớn trên thế giới.
* **Triển khai trên môi trường thực tế:**
  + Phát hành ứng dụng trên các nền tảng cloud như AWS, Azure.
  + Cung cấp phiên bản ứng dụng dành cho thiết bị di động trên Android và iOS.
* **Tương tác cộng đồng:**
  + Phát triển diễn đàn thảo luận dành cho nhà đầu tư.
  + Tích hợp hệ thống đánh giá và xếp hạng mã chứng khoán.

Với những định hướng trên, hệ thống có tiềm năng trở thành một nền tảng phân tích thị trường tài chính mạnh mẽ, hỗ trợ đắc lực cho nhà đầu tư trong việc tiếp cận thông tin và tối ưu hóa quyết định đầu tư.

# **TÀI LIỆU THAM KHẢO**

[1] MDN Web Docs. *React Documentation*. <https://developer.mozilla.org/>

[2] Node.js Official Documentation. *Node.js API Reference*. <https://nodejs.org/>

[3] W3Schools. *SQL Tutorial*. <https://www.w3schools.com/sql/>

[4] Facebook Open Source. *ReactJS*. <https://reactjs.org/>

[5] Sequelize Documentation. *Node.js ORM for SQL Databases*. <https://sequelize.org/>

[6] TradingView. *JavaScript Charting Library*. <https://www.tradingview.com/>

[7] Sở Giao dịch Chứng khoán TP. Hồ Chí Minh (HOSE). <https://www.hsx.vn/>

[8] Sở Giao dịch Chứng khoán Hà Nội (HNX). <https://www.hnx.vn/>

[9] Dữ liệu cào của: <https://fireant.vn/>, <https://fialda.com/home>, <https://www.ssi.com.vn/>