TRƯỜNG ĐẠI HỌC TRÀ VINH
TRƯỜNG KỸ THUẬT VÀ CÔNG NGHỆ
KHOA CÔNG NGHỆ THÔNG TIN

ĐỒ ÁN KẾT THÚC HỌC PHẦN

CÔNG NGHỆ PHẦN MỀM

(MHP:220055)

XÂY DỰNG WEBSITE QUẢN LÝ TÀI CHÍNH

Giáo viên hướng dẫn :
Ts.Nguyễn Bảo Ân

Sinh viên thực hiện:
Huỳnh Quốc Kiệt-110122100-DA22TTD
Mai Hồng Lợi-110122106-DA22TTD
Đặng Minh Hiếu-110122075-DA22TTD

Vĩnh long,ngày….tháng 7 năm 2025

TRƯỜNG ĐẠI HỌC TRÀ VINH

TRƯỜNG KỸ THUẬT VÀ CÔNG NGHỆ

KHOA CÔNG NGHỆ THÔNG TIN

ĐỒ ÁN KẾT THÚC HỌC PHẦN

CÔNG NGHỆ PHẦN MỀM

(MHP:220055)

XÂY DỰNG WEBSITE QUẢN LÝ TÀI CHÍNH

Giáo viên hướng dẫn :

Ts.Nguyễn Bảo Ân

Sinh viên thực hiện:

Huỳnh Quốc Kiệt-110122100-DA22TTD

Mai Hồng Lợi-110122106-DA22TTD

Đặng Minh Hiếu-110122075-DA22TTD

Vinh long,ngày….tháng 7 năm 2025

NHẬN XÉT CỦA GIẢNG VIÊN HƯỚNG DẪN

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

NHẬN XÉT CỦA THÀNH VIÊN HỘI ĐỒNG

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

LỜI MỞ ĐẦU

Trong thời đại số hóa hiện nay, nhu cầu quản lý tài chính cá nhân và doanh nghiệp một cách hiệu quả, minh bạch và khoa học trở nên vô cùng cần thiết. Việc ghi chép thủ công bằng sổ sách truyền thống không còn phù hợp khi khối lượng giao dịch lớn, đa dạng và cần được phân loại, thống kê, báo cáo nhanh chóng. Điều này đặt ra yêu cầu cấp thiết về một hệ thống quản lý tài chính có thể hỗ trợ người dùng kiểm soát thu chi, quản lý khoản vay nợ, lập kế hoạch chi tiêu và theo dõi tình hình tài chính một cách trực quan.

Xuất phát từ thực tế đó, nhóm chúng em đã lựa chọn đề tài **"Xây dựng website Quản lý tài chính"**, nhằm xây dựng một hệ thống phần mềm hỗ trợ người dùng ghi nhận, quản lý, thống kê và báo cáo các hoạt động tài chính hàng ngày. Ứng dụng giúp tiết kiệm thời gian, giảm thiểu sai sót và mang lại cái nhìn tổng quan nhất về tình hình tài chính của bản thân hoặc tổ chức.

Dự án được phát triển theo kiến trúc client/server với sự tách biệt rõ ràng giữa frontend và backend, đảm bảo tính mở rộng, dễ bảo trì, cũng như phù hợp với định hướng phát triển phần mềm hiện đại. Các thành phần trong hệ thống được thiết kế để giao tiếp qua API, đồng thời nhóm cũng áp dụng đầy đủ các quy trình và công cụ cần thiết trong suốt vòng đời phát triển, từ lên kế hoạch, thiết kế giao diện trên Figma, quản lý tiến độ công việc trên Jira, lập trình trên Visual Studio Code, kiểm thử đến triển khai website lên GitHub.

Thông qua dự án này, nhóm không chỉ củng cố kiến thức đã học về công nghệ phần mềm, lập trình web và quản lý cơ sở dữ liệu, mà còn có cơ hội tiếp cận thực tiễn phát triển một hệ thống phần mềm hoàn chỉnh theo mô hình công nghiệp. Đây cũng là dịp để rèn luyện kỹ năng làm việc nhóm, giải quyết vấn đề, lập kế hoạch tài chính và sử dụng thành thạo các công cụ hỗ trợ chuyên nghiệp trong quá trình phát triển phần mềm.

LỜI CẢM ƠN

Để hoàn thành bài báo cáo môn học Công Nghệ Phần Mềm, nhóm chúng em xin gửi lời cảm ơn chân thành và sâu sắc nhất đến thầy Nguyễn Bảo Ân. Trong suốt quá trình học tập và thực hiện đề tài nhóm chúng em đã nhận được sự hướng dẫn và chỉ dẫn tận tình của thầy. Qua từng buổi học thầy đã giúp chúng em nắm vững các kiến thức và các cốt lõi trong nền tảng của môn học.

Nhóm chúng em cũng xin bày tỏ lời cảm ơn đến thầy Nguyễn Bảo Ân và các Thầy và Cô trong Khoa kỹ Thuật Công Nghệ , Trường Đại Học Trà vinh, đã giúp cho chúng em có những nền tảng về các kiến thức của các môn học trong suốt thời gian học tập.

Chúng em rất mong nhận được những ý kiến đóng góp quý báu của Thầy/Cô và các bạn để hoàn thiện hơn với kiến thức của mình và phát triển hơn trong tương lai.

Nhóm chúng em xin chân thành cảm ơn!

MỤC LỤC

Contents

DANH MỤC HÌNH ẢNH

DANH MỤC BẢNG BIỂU

CHƯƠNG 1 : TỔNG QUAN

1.1 Lý do chọn đề tài

Trong bối cảnh xã hội hiện đại với nhịp sống nhanh và chi phí sinh hoạt ngày càng phức tạp, việc quản lý tài chính cá nhân một cách khoa học đã trở thành một kỹ năng thiết yếu đối với mọi đối tượng, từ sinh viên, người đi làm cho đến các hộ gia đình. Một kế hoạch tài chính vững vàng không chỉ giúp đảm bảo sự ổn định trong cuộc sống hàng ngày mà còn là nền tảng để hiện thực hóa các mục tiêu dài hạn như mua sắm tài sản, đầu tư hay chuẩn bị cho tương lai.

Tuy nhiên, trên thực tế, nhiều người vẫn đang gặp khó khăn trong việc theo dõi dòng tiền của mình. Các phương pháp quản lý thủ công như ghi chép bằng sổ tay hay sử dụng bảng tính Excel tuy quen thuộc nhưng lại bộc lộ nhiều hạn chế lớn:

- **Dễ sai sót:** Việc nhập liệu thủ công có nguy cơ cao dẫn đến nhầm lẫn, thiếu chính xác.
- **Tốn thời gian:** Quá trình ghi chép và tổng hợp dữ liệu mất nhiều thời gian và công sức.
- **Khó phân tích:** Dữ liệu rời rạc khiến việc tổng hợp, tạo báo cáo trực quan và phân tích xu hướng chi tiêu trở nên vô cùng khó khăn.
- **Thiếu tính năng tự động:** Hoàn toàn không có khả năng cảnh báo khi chi tiêu vượt ngân sách, nhắc nhở các khoản thanh toán định kỳ hay đưa ra các gợi ý tài chính thông minh.

Xuất phát từ những nhu cầu cấp thiết và các bất cập nêu trên, đề tài "Xây dựng website quản lý tài chính cá nhân" được lựa chọn. Mục tiêu của dự án là phát triển một giải pháp công nghệ tập trung, giúp người dùng khắc phục hoàn toàn những nhược điểm của phương pháp truyền thống. Bằng cách cung cấp một công cụ theo dõi, phân tích và lập kế hoạch tài chính một cách trực quan, an toàn và hiệu quả, dự án không chỉ giải quyết một bài toán thực tiễn mà còn mang lại giá trị thiết thực, giúp người dùng kiểm soát tốt hơn sức khỏe tài chính của bản thân.

1.2 Mục tiêu của đề tài

Việc quản lý tài chính cá nhân một cách hiệu quả là vô cùng cần thiết trong cuộc sống hiện đại. Nó không chỉ giúp đảm bảo sự ổn định tài chính mà còn là nền tảng để đạt được các mục tiêu lớn hơn trong cuộc sống như mua nhà, đầu tư hay nghỉ hưu an

toàn. Mục tiêu chính của dự án này là xây dựng một website giúp người dùng:

- **Theo dõi thu chi:** Ghi chép lại tất cả các khoản thu nhập và chi tiêu hàng ngày một cách dễ dàng.
- **Phân loại giao dịch:** Tự động hoặc thủ công phân loại các khoản chi tiêu (ví dụ: ăn uống, đi lại, giải trí) để có cái nhìn rõ ràng về thói quen chi tiêu.
- **Lập ngân sách:** Thiết lập các hạn mức chi tiêu cho từng hạng mục để kiểm soát và điều chỉnh hành vi tiêu dùng.
- **Báo cáo và phân tích:** Cung cấp các biểu đồ, báo cáo trực quan về tình hình tài chính theo tuần, tháng, năm, giúp người dùng dễ dàng nắm bắt và đưa ra quyết định.
- **Đặt mục tiêu tiết kiệm:** Hỗ trợ người dùng đặt ra các mục tiêu tài chính cụ thể (ví dụ: tiết kiệm cho một chuyến du lịch, mua một món đồ công nghệ) và theo dõi tiến độ thực hiện.

1.3 Đối tượng và phạm vi nghiên cứu

- Hệ thống website quản lý tài chính cá nhân.
- Các nghiệp vụ chính bao gồm: ghi chép thu nhập, chi tiêu, nợ, tiết kiệm và báo cáo thống kê tài chính.
- Công nghệ xây dựng website: frontend, backend, database, API và triển khai trên cloud hoặc server.

Phạm vi nghiên cứu:

- Đối tượng sử dụng: sinh viên, nhân viên văn phòng, cá nhân có nhu cầu quản lý tài chính cá nhân.
- **Phạm vi chức năng:**
  - Đăng ký, đăng nhập, quản lý thông tin người dùng
  - Quản lý thu nhập và chi tiêu
  - Quản lý nợ và tiết kiệm
  - Hiển thị báo cáo, thống kê bằng biểu đồ

Phạm vi công nghệ:

- Frontend: HTML, CSS, JavaScript (hoặc framework như ReactJS)
- Backend: NodeJS, ExpressJS (hoặc PHP/Laravel, Python/Flask tùy chọn)
- Database: MySQL hoặc MongoDB
- Triển khai: hosting hoặc cloud server

1.4 Phương pháp thực hiện

1. **Khảo sát và thu thập yêu cầu:**

- Khảo sát nhu cầu người dùng mục tiêu (sinh viên, nhân viên văn phòng, hộ gia đình).
- Xác định các chức năng cần thiết: quản lý thu nhập, chi tiêu, nợ, tiết kiệm, báo cáo thống kê.

2. **Phân tích và đặc tả hệ thống:**

- Vẽ **Use Case Diagram** để xác định chức năng. Đặc tả chi tiết yêu cầu chức năng và phi chức năng.

3. **Thiết kế hệ thống:**

- Thiết kế cơ sở dữ liệu: xác định bảng, thuộc tính, mối quan hệ (ERD). Thiết kế kiến trúc hệ thống (frontend, backend, database).
- Thiết kế giao diện người dùng (UI/UX) trên Figma hoặc phác thảo tay.

4. **Xây dựng và phát triển:**

- Frontend: xây dựng giao diện bằng HTML, CSS, JavaScript (hoặc ReactJS).
- Backend: xây dựng API với NodeJS + ExpressJS để xử lý nghiệp vụ.
- Database: thiết kế và cài đặt MySQL hoặc MongoDB.

5. **Kiểm thử (Testing):**

- Kiểm thử đơn vị (unit test) cho từng chức năng.
- Kiểm thử tích hợp (integration test) giữa frontend, backend và database.
- Kiểm thử giao diện người dùng (UI testing).

6. **Triển khai (Deployment):**

- Triển khai website lên hosting hoặc cloud server (AWS, Azure, hoặc VPS).
- Cấu hình domain (nếu có) và đảm bảo website hoạt động ổn định.

7. **Đánh giá và hoàn thiện:**

- Thu thập ý kiến phản hồi từ người dùng thử nghiệm. Sửa lỗi, cải thiện giao diện, bổ sung tính năng cần thiết.

8. **Viết báo cáo và chuẩn bị thuyết trình:**

- Tổng hợp kết quả, viết báo cáo chi tiết các bước thực hiện.Chuẩn bị slide thuyết trình và demo website trước Thầy/Cô.

1.5 Ý nghĩa của đề tài

1. Ý nghĩa thực tiễn

- Giúp người dùng quản lý thu chi, nợ và tiết kiệm một cách khoa học, thuận tiện, mọi lúc mọi nơi.
- Giảm thiểu sai sót khi ghi chép thủ công, hỗ trợ đưa ra quyết định chi tiêu hợp lý.
- Góp phần nâng cao ý thức quản lý tài chính cá nhân, từ đó cải thiện chất lượng cuộc sống.

2. Ý nghĩa khoa học

- Vận dụng kiến thức công nghệ phần mềm, cơ sở dữ liệu, lập trình frontend – backend trong thực tiễn.

Tích hợp các công nghệ mới như RESTful API, cloud, microservices (nếu mở rộng) để phát triển ứng dụng web hiện đại.

- Làm nền tảng cho các nghiên cứu và phát triển ứng dụng tài chính cá nhân nâng cao trong tương lai (tích hợp AI gợi ý chi tiêu, kết nối ngân hàng, ví điện tử).

3. Ý nghĩa đào tạo

- Giúp nhóm thực hành quy trình phát triển phần mềm từ phân tích, thiết kế, xây dựng, kiểm thử đến triển khai.
- Rèn luyện kỹ năng làm việc nhóm, quản lý dự án, sử dụng GitHub, triển khai trên server thực tế.
- Là hành trang vững chắc cho công việc lập trình viên web, backend developer hoặc product manager trong lĩnh vực công nghệ tài chính.

2. Bảng phân công công việc

STT	Họ tên thành viên	Công việc phụ trách
1	Huỳnh Quốc Kiệt	- Thiết kế giao diện trên Figma (login, dashboard, thu chi, báo cáo) - Quản lý dự án trên Jira: tạo backlog, task, sprint - Viết phần thiết kế giao diện trong báo cáo

2	Đặng Minh Hiếu	Lập trình front-end trên Visual Studio Code (HTML, CSS, JS/React) - Quản lý GitHub: tạo repository, commit, push/pull code - Viết phần cài đặt website và mã nguồn trong báo cáo
3	Mai Hồng Lợi	-Làm slide thuyết trình PowerPoint -Quản lý dự án trên Jira: tạo backlog, task, sprint -Viết phần kết quả thực nghiệm và đánh giá trong báo cáo

Bảng 2 . Phân công công việc của nhóm

CHƯƠNG 2 : CƠ SỞ LÝ THUYẾT

2.1. Khái niệm về quản lý tài chính

Quản lý tài chính cá nhân là quá trình lập kế hoạch, tổ chức, kiểm soát và giám sát các hoạt động tài chính của bản thân hoặc gia đình, bao gồm:

- Quản lý thu nhập (tiền lương, thưởng, nguồn thu khác)
- Quản lý chi tiêu (ăn uống, đi lại, giải trí, học tập,…)
- Quản lý nợ (khoản vay, trả góp)
- Quản lý tiết kiệm và đầu tư

Việc quản lý tốt giúp cân đối chi tiêu, tích lũy tài sản và đảm bảo an toàn tài chính trong tương lai.

2.2. Khái niệm website và công nghệ phát triển web

2.2.1. Website

Website là tập hợp các trang web liên kết với nhau, được lưu trữ trên server, truy cập qua internet bằng trình duyệt web.

2.2.2. Công nghệ phát triển web

- **Frontend**: xây dựng giao diện người dùng bằng HTML, CSS, JavaScript hoặc framework (ReactJS, Angular, VueJS).
- **Backend**: xử lý logic nghiệp vụ, kết nối database, xây dựng RESTful API bằng NodeJS, PHP/Laravel hoặc Python/Flask.
- **Database**: lưu trữ dữ liệu (MySQL, PostgreSQL, MongoDB).
- **Triển khai**: hosting hoặc cloud server (AWS, GCP, Azure).

2.3. Các mô hình phát triển phần mềm

2.3.1. Mô hình thác nước

Phát triển tuần tự qua các giai đoạn: yêu cầu ➔thiết kế ➔cài đặt ➔kiểm thử ➔ triển khai ➔bảo trì. Phù hợp dự án nhỏ, yêu cầu rõ ràng ngay từ đầu.

2.3.2. Mô hình Agile/Scrum

Phát triển theo từng sprint (2-4 tuần), có tính lặp, phản hồi sớm, dễ thích ứng thay đổi yêu cầu. Phù hợp với dự án web và mobile hiện nay.

2.4. Kiến trúc phần mềm website quản lý tài chính

Hệ thống website quản lý tài chính thường áp dụng kiến trúc client-server hoặc microservices nếu mở rộng:

- **Frontend**: hiển thị giao diện người dùng, gọi API.

- **Backend**: xử lý dữ liệu, xác thực người dùng, cung cấp API cho frontend.
- **Database**: lưu trữ thông tin người dùng, thu chi, nợ, tiết kiệm.
- **RESTful API**: chuẩn giao tiếp giữa frontend và backend.
- **Triển khai cloud**: đảm bảo khả năng mở rộng và truy cập mọi lúc mọi nơi.

2.5. Các công nghệ sử dụng trong đề tài

2.5.1. HTML, CSS, JavaScript

- **HTML (HyperText Markup Language)**: ngôn ngữ đánh dấu dùng để xây dựng cấu trúc nội dung của website, định nghĩa các thành phần như tiêu đề, đoạn văn, hình ảnh, liên kết.
- **CSS (Cascading Style Sheets)**: ngôn ngữ tạo kiểu giúp định dạng và thiết kế giao diện website như màu sắc, font chữ, bố cục, khoảng cách, tạo hiệu ứng responsive phù hợp với nhiều thiết bị (mobile, tablet, laptop).
- **JavaScript**: ngôn ngữ lập trình phía client giúp website trở nên động và tương tác hơn, xử lý các sự kiện người dùng như click, nhập dữ liệu, validation form, gọi API để lấy dữ liệu và hiển thị lên giao diện mà không cần reload toàn bộ trang.

2.5.2. ReactJS

- **ReactJS** là thư viện JavaScript do Facebook phát triển, hỗ trợ xây dựng giao diện người dùng một cách nhanh chóng, quản lý DOM ảo (Virtual DOM) giúp tăng tốc độ render.
- Cho phép xây dựng **component-based architecture** – chia giao diện thành các thành phần nhỏ (component) có thể tái sử dụng, dễ bảo trì và phát triển tính năng mới.
- Hỗ trợ **Single Page Application (SPA)**, giúp website tải nhanh, chỉ render phần giao diện thay đổi mà không phải tải lại toàn bộ trang.

2.5.3. NodeJS và ExpressJS

- **NodeJS**: nền tảng chạy JavaScript phía server, sử dụng mô hình non-blocking I/O và event-driven, giúp xây dựng server xử lý song song nhiều request với tốc độ cao.
- **ExpressJS**: framework của NodeJS, cung cấp bộ công cụ mạnh mẽ để xây dựng RESTful API dễ dàng, định nghĩa route, middleware và quản lý request-response hiệu quả.
- Hỗ trợ tích hợp với các database như MySQL, MongoDB, và triển khai trên

cloud nhanh chóng.

2.5.4. MySQL/MongoDB

- **MySQL**: hệ quản trị cơ sở dữ liệu quan hệ (RDBMS), sử dụng SQL để truy vấn, dữ liệu được lưu trong các bảng có mối quan hệ, đảm bảo tính toàn vẹn và dễ quản lý khi dữ liệu có cấu trúc rõ ràng (ví dụ: user, income, expense).
- **MongoDB**: cơ sở dữ liệu NoSQL, lưu dữ liệu dưới dạng document JSON, dễ mở rộng, phù hợp lưu trữ dữ liệu không có cấu trúc cố định hoặc dữ liệu cần thay đổi linh hoạt theo từng user.

2.5.5. Git/GitHub

GitHub là một nền tảng dựa trên web cung cấp dịch vụ lưu trữ mã nguồn cho các dự án phần mềm sử dụng hệ thống kiểm soát phiên bản Git.Đây được xem là một mạng xã hội cho các lập trình viên, nơi họ có thể lưu trữ, quản lý và chia sẻ mã nguồn một cách hiệu quả.

- **Git**: hệ thống quản lý phiên bản phân tán, giúp lưu lại lịch sử thay đổi của source code, hỗ trợ làm việc nhóm hiệu quả, tránh mất mát dữ liệu khi code bị lỗi.
- **GitHub**: nền tảng lưu trữ code online sử dụng Git, cung cấp tính năng pull request, code review, issue tracker, và CI/CD pipeline để triển khai tự động.

2.5.6. Docker

- **Docker**: nền tảng ảo hóa cấp hệ điều hành (containerization), giúp đóng gói ứng dụng và môi trường chạy vào container, đảm bảo chạy được trên mọi máy chủ mà không phụ thuộc vào cấu hình của hệ điều hành.Giúp triển khai nhanh, dễ scale hệ thống, quản lý nhiều microservices hiệu quả

2.5.7 Quản lý dự án phần mềm với Jira

- **Jira:** là một công cụ quản lý dự án và theo dõi lỗi do Atlassian phát triển, được sử dụng rộng rãi trong các nhóm phát triển phần mềm.Ban đầu, Jira được xây dựng chuyên biệt cho các lập trình viên và dựa trên nguyên lý Agile, nhưng sau đó đã đượcปรับ chỉnh để phù hợp với nhiều nhu cầu đa dạng hơn.

2.5.8 Thiết kế giao diện và trải nghiệm người dùng (UI/UX) với Figma

- **Figma:** là một ứng dụng thiết kế đồ họa và tạo mẫu dựa trên nền tảng web, chuyên dùng cho việc thiết kế giao diện người dùng (UI) và trải nghiệm người dùng (UX).Công cụ này cho phép các nhà thiết kế, nhà phát triển, và các bên liên quan khác cùng nhau hợp tác một cách hiệu quả.

2.5.9 Áp dụng nguyên tắc RESTful API

REST (Representational State Transfer) là một kiểu kiến trúc để thiết kế các ứng dụng mạng. Một API tuân thủ các nguyên tắc của REST được gọi là RESTful API.

- Kiến trúc Client-Server: Phân tách rõ ràng giữa giao diện người dùng (client) và lưu trữ dữ liệu (server), cho phép chúng phát triển độc lập.
- Phi trạng thái (Stateless): Mỗi yêu cầu từ client đến server phải chứa tất cả thông tin cần thiết để server hiểu và xử lý nó. Server không lưu trữ bất kỳ trạng thái nào của client giữa các yêu cầu.
- Giao diện đồng nhất: Sử dụng các phương thức HTTP tiêu chuẩn (GET, POST, PUT, DELETE) để thực hiện các hành động trên tài nguyên.

2.6 Ngôn ngữ và công nghệ sử dụng

1. Ngôn ngữ lập trình

- **HTML (HyperText Markup Language)**: ngôn ngữ đánh dấu dùng để xây dựng cấu trúc trang web, tạo khung cho giao diện người dùng.
- **CSS (Cascading Style Sheets)**: ngôn ngữ tạo kiểu giúp định dạng, thiết kế giao diện đẹp mắt, responsive trên nhiều thiết bị.
- **JavaScript**: ngôn ngữ lập trình phía client, giúp website có các chức năng động và tương tác tốt hơn với người dùng.
- **NodeJS (JavaScript runtime)**: cho phép sử dụng JavaScript phía server, xây dựng backend nhanh chóng và dễ dàng mở rộng.
- **SQL** (Structured Query Language): ngôn ngữ truy vấn dữ liệu trong MySQL.

2. Công nghệ frontend

- **ReactJS (nếu sử dụng)**: thư viện JavaScript hỗ trợ xây dựng Single Page Application, quản lý component, tăng tốc độ render và nâng cao trải nghiệm người dùng.
- **Bootstrap/Tailwind CSS (nếu sử dụng)**: framework CSS giúp thiết kế giao diện nhanh, responsive, tiết kiệm thời gian viết CSS thuần.

3. Công nghệ backend

- **NodeJS + ExpressJS**: xây dựng server backend, định nghĩa route, xử lý logic nghiệp vụ và cung cấp RESTful API cho frontend.

4. Cơ sở dữ liệu

- **MySQL:** hệ quản trị cơ sở dữ liệu quan hệ, lưu trữ dữ liệu có cấu trúc rõ ràng,

đảm bảo tính toàn vẹn dữ liệu.

- **MongoDB (nếu sử dụng):** cơ sở dữ liệu NoSQL, lưu trữ dữ liệu dạng document JSON, linh hoạt cho việc mở rộng.

5. Công cụ quản lý mã nguồn

- **Git:** hệ thống quản lý phiên bản giúp lưu lại lịch sử thay đổi, hỗ trợ làm việc nhóm.
- **GitHub:** nền tảng lưu trữ code online, quản lý project, pull request, issue tracker, tích hợp CI/CD.

6. Công cụ triển khai

- Docker (nếu sử dụng): đóng gói ứng dụng thành container, đảm bảo chạy đồng nhất trên nhiều môi trường, dễ triển khai.

7. Công cụ thiết kế và hỗ trợ phát triển

- **Figma:** thiết kế giao diện UI/UX trước khi code.
- **Visual Studio Code:** IDE chính để lập trình frontend và backend, tích hợp nhiều extension hỗ trợ JavaScript, ReactJS, NodeJS.
- **Postman:** công cụ test API, đảm bảo backend hoạt động đúng trước khi kết nối frontend.

CHƯƠNG 3: PHÂN TÍCH VÀ THIẾT KẾ HỆ THỐNG

3.1. Khảo sát hiện trạng

Hiện nay, nhiều cá nhân quản lý tài chính bằng sổ tay hoặc file Excel. Phương pháp này có nhược điểm:

- Dễ thất lạc, mất dữ liệu nếu không sao lưu.
- Không thể thống kê, phân tích nhanh thu chi, tiết kiệm.
- Không có tính năng nhắc nhở hoặc biểu đồ trực quan.

Vì vậy, việc xây dựng website quản lý tài chính giúp giải quyết các vấn đề trên, nâng cao hiệu quả quản lý chi tiêu.

3.2. Phân tích yêu cầu hệ thống

3.2.1. Yêu cầu chức năng

Các yêu cầu chức năng mô tả những gì hệ thống phải làm. Đối với website quản lý tài chính, các chức năng chính bao gồm:

Quản lý người dùng:

- Đăng ký, đăng nhập, đăng xuất tài khoản.
- Quản lý thông tin cá nhân.
- Quên mật khẩu/Đặt lại mật khẩu.

Quản lý giao dịch:

- Thêm, sửa, xóa các khoản thu nhập và chi tiêu.
- Gán giao dịch vào các danh mục cụ thể (ăn uống, mua sắm, hóa đơn,...).
- Thêm ghi chú, hình ảnh hóa đơn cho mỗi giao dịch.

Quản lý ngân sách:

- Tạo ngân sách chi tiêu cho các danh mục theo tháng.
- Hiển thị cảnh báo khi chi tiêu gần đạt hoặc vượt ngân sách.

Báo cáo và thống kê:

- Hiển thị tổng quan tài chính (tổng thu, tổng chi, số dư).
- Xem báo cáo thu chi dưới dạng biểu đồ (tròn, cột) theo thời gian và danh mục.
- Xuất dữ liệu báo cáo ra các định dạng phổ biến (CSV, Excel).

Quản lý mục tiêu tiết kiệm:

- Tạo các mục tiêu tiết kiệm với số tiền và thời hạn cụ thể.

- Theo dõi tiến độ hoàn thành mục tiêu.

Đối với admin (nếu có)

- Quản lý thông tin người dùng
- Xem báo cáo tổng quan hệ thống

3.2.2. Yêu cầu phi chức năng

Các yêu cầu phi chức năng mô tả hệ thống nên hoạt động như thế nào, tập trung vào các thuộc tính chất lượng.

Bảo mật:

- Mã hóa mật khẩu người dùng.
- Sử dụng giao thức HTTPS để bảo vệ dữ liệu truyền tải.
- Dữ liệu tài chính của người dùng phải được bảo mật tuyệt đối, chống truy cập trái phép.

Hiệu năng:

- Thời gian tải trang không quá 3 giây.
- Hệ thống phải phản hồi các yêu cầu của người dùng trong vòng 2 giây.

Tính khả dụng (Usability):

- Giao diện thân thiện, dễ sử dụng, phù hợp với cả người dùng không rành về công nghệ.
- Website phải tương thích và hiển thị tốt trên các trình duyệt phổ biến (Chrome, Firefox, Safari) và các thiết bị khác nhau (máy tính, máy tính bảng, điện thoại).

Độ tin cậy (Reliability):

- Hệ thống phải hoạt động ổn định 24/7, với thời gian hoạt động (uptime) đạt 99.5%.
- Dữ liệu phải được sao lưu định kỳ để tránh mất mát.

Khả năng bảo trì (Maintainability):

- Mã nguồn được tổ chức rõ ràng, có tài liệu và bình luận đầy đủ để dễ dàng sửa lỗi và phát triển các tính năng mới.

3.2.3. Thiết kế hệ thống

3.2.3.1. Kiến trúc tổng thể

Hệ thống được thiết kế theo kiến trúc Client-Server.

- **Client (Phía người dùng):** Là một ứng dụng web đơn trang (Single Page Application - SPA) được xây dựng bằng framework React.js. Người dùng sẽ tương tác

với giao diện này thông qua trình duyệt web.

- **Server (Phía máy chủ):** Bao gồm:
- **API Server:** Được xây dựng bằng Node.js và Express.js, tuân thủ theo nguyên tắc RESTful API. Đây là nơi xử lý các logic nghiệp vụ, xác thực người dùng và giao tiếp với cơ sở dữ liệu.
- **Cơ sở dữ liệu (Database):** Sử dụng hệ quản trị cơ sở dữ liệu quan hệ (ví dụ: PostgreSQL hoặc MySQL) để lưu trữ toàn bộ dữ liệu của ứng dụng như thông tin người dùng, giao dịch, ngân sách...

3.2.4. Thiết kế cơ sở dữ liệu

Mô hình quan hệ thực thể (ERD) được sử dụng để mô tả cấu trúc dữ liệu.

- **Bảng Users:** Lưu thông tin người dùng (id, username, password_hash, email, created_at).
- **Bảng Categories:** Lưu các danh mục thu/chi (id, user_id, name, type - 'income' hoặc 'expense').
- **Bảng Transactions:** Lưu thông tin giao dịch (id, user_id, category_id, amount, date, description).
- **Bảng Budgets:** Lưu ngân sách (id, user_id, category_id, amount, month, year).
- **Bảng SavingsGoals:** Lưu mục tiêu tiết kiệm (id, user_id, goal_name, target_amount, current_amount, due_date).

3.2.5 Thiết kế API

API được thiết kế theo chuẩn RESTful, sử dụng JSON làm định dạng trao đổi dữ liệu. Tài liệu API chi tiết được tạo bằng Swagger/OpenAPI, mô tả rõ các endpoint, phương thức HTTP, tham số yêu cầu và cấu trúc phản hồi.

Một số endpoint chính:

- POST /api/users/register: Đăng ký người dùng mới.
- POST /api/users/login: Đăng nhập.

Method	Endpoint	Mô tả
POST	/api/auth/register	Đăng ký tài khoản mới
POST	/api/auth/login	Đăng nhập

Bảng 3.2.5 Các API liên quan Authentication

3.2.6. Thiết kế giao diện

- **Công cụ:** Figma được sử dụng để thiết kế toàn bộ giao diện và tạo các prototype tương tác.

Nguyên tắc thiết kế:

- **Tối giản và sạch sẽ:** Giao diện được thiết kế gọn gàng, tập trung vào nội dung chính để người dùng không bị phân tâm.
- **Nhất quán:** Sử dụng bộ màu sắc, font chữ và các thành phần giao diện (nút, biểu mẫu) thống nhất trên toàn bộ website.
- **Trực quan hóa dữ liệu:** Sử dụng biểu đồ để trình bày dữ liệu tài chính một cách sinh động, dễ hiểu.

4. Triển khai và công nghệ sử dụng

4.1. Danh sách các công nghệ đã sử dụng

- **Frontend:** React.js, Redux, Axios.
- **Backend:** Node.js, Express.js.
- **Cơ sở dữ liệu:** PostgreSQL.
- **Quản lý dự án:** Jira.
- **Thiết kế UI/UX:** Figma.
- **Quản lý mã nguồn:** Git & GitHub.
- **CI/CD:** GitHub Actions.
- **Kiểm thử API:** Postman.
- **Đóng gói & Triển khai:** Docker.
- **Tài liệu API:** Swagger.

4.2. Quy trình CI/CD với GitHub Actions

Một quy trình làm việc tự động được thiết lập bằng GitHub Actions:

- Khi lập trình viên đẩy mã nguồn (push) lên một nhánh (branch): GitHub Actions sẽ tự động chạy các bài kiểm thử đơn vị (unit test) và kiểm tra chất lượng mã nguồn.
- Khi một Pull Request được tạo: Quy trình tương tự được kích hoạt để đảm bảo mã mới không gây ra lỗi khi hợp nhất vào nhánh chính.
- Khi mã được hợp nhất (merge) vào nhánh chính (main): GitHub Actions sẽ tự động xây dựng (build) image cho cả frontend và backend, sau đó đẩy lên Docker

Hub.

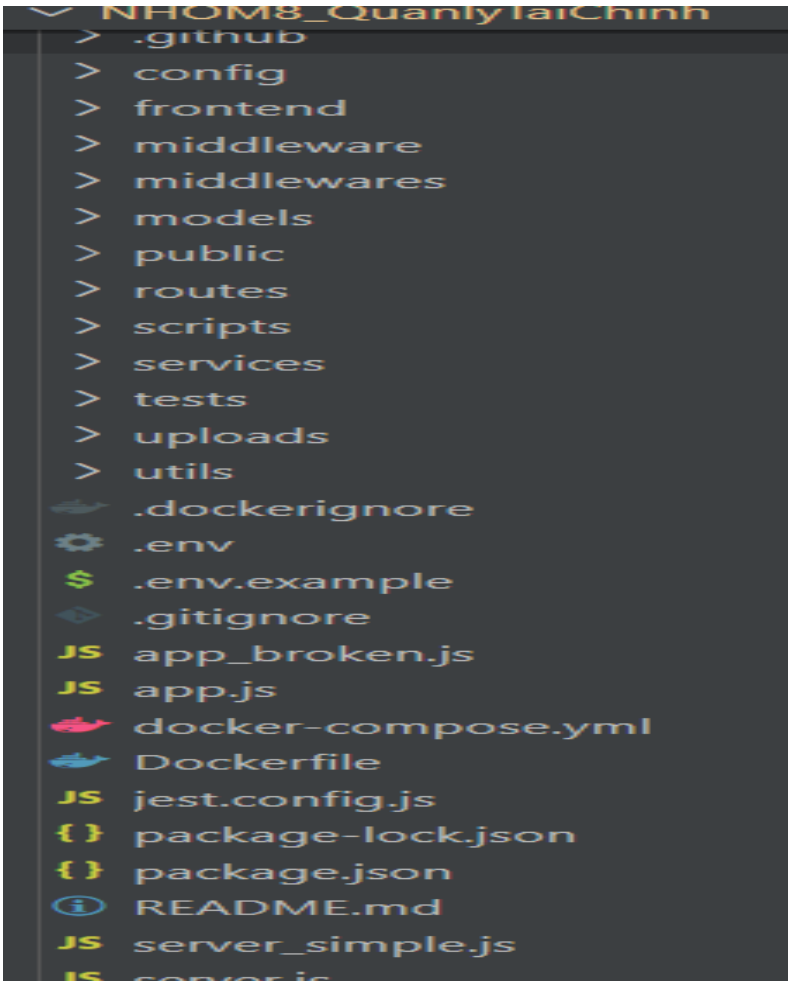
- Triển khai: Kích hoạt một workflow khác để kéo image mới nhất từ Docker Hub về máy chủ và khởi chạy ứng dụng.

4.3 Cấu trúc thư mục

Cấu trúc thư mục được xây dựng như sau:

Thư mục backend/ chứa toàn bộ mã nguồn phía server, nơi xử lý logic nghiệp vụ, kết nối cơ sở dữ liệu, xác thực người dùng và cung cấp các API cho frontend gọi đến. Thư mục backend/src/ gồm các thư mục như config/ để kết nối với cơ sở dữ liệu thông qua file môi trường .env, models/ để định nghĩa dữ liệu (MongoDB), controllers/ để xử lý request, routes/ để khai báo các endpoint, và services/ để xử lý logic nghiệp vụ. Backend đóng vai trò là trung tâm xử lý và giao tiếp với frontend thông qua RESTful API.

Thư mục solution/ là nơi xây dựng giao diện người dùng bằng ngôn ngữ HTML, CSS và Javascript. Đây là ứng dụng client-side, nơi người dùng tương tác trực tiếp thông qua trình duyệt, thực hiện các hành động như đăng ký, đăng nhập, thêm danh mục công việc, thêm công việc,… Solution gửi request tới backend qua HTTP (dùng axios), nhận dữ liệu JSON và hiển thị ra màn hình. Toàn bộ điều hướng, xử lý trạng thái và logic hiển thị đều nằm trong phần này, giúp tách biệt hoàn toàn khỏi backend.



Hình 4.3 Cấu trúc thư mục

CHƯƠNG 4. KẾT QUẢ THỰC NGHIỆM

4.1. Mục tiêu thực nghiệm

Thực nghiệm nhằm đánh giá mức độ hoàn thiện của hệ thống, tính đúng đắn của các chức năng chính và mức độ phù hợp với yêu cầu thực tế quản lý. Đồng thời, qua thực nghiệm có thể phát hiện các lỗi phát sinh và kịp thời sửa chữa trước khi triển khai.

- Kiểm tra tính đúng đắn của các chức năng website quản lý tài chính.
- Đánh giá giao diện, hiệu năng và mức độ phù hợp với yêu cầu thực tế.
- Xác minh khả năng triển khai trên môi trường cục bộ và quản lý dự án qua GitHub, Jira.

Quy trình phát triển phần mềm bằng công cụ:

- **Figma**: Thiết kế giao diện
- **Visual Studio Code**: Phát triển source code
- **GitHub**: Quản lý mã nguồn
- **Jira**: Quản lý dự án

4.2. Quy trình thực nghiệm Thiết kế giao diện trên Figma

- Phân tích yêu cầu.
- Vẽ wireframe, prototype.
- Xây dựng UI/UX chi tiết.

Lập trình website trên Visual Studio Code

- Code front-end (HTML, CSS, JS, React hoặc PHP).
- Code back-end (Node.js/Express hoặc PHP) kết nối database.

Quản lý source code trên GitHub

- Tạo repository.
- Push code, commit theo từng chức năng.
- Pull code về máy khác test.

Quản lý dự án trên Jira

- Tạo project Agile/Scrum.
- Tạo Product Backlog, Sprint Backlog.
- Quản lý task theo To do – In progress – Done.
- Triển khai hệ thống trên máy cục bộ Nhập dữ liệu mẫu:

- Danh mục thu nhập, chi tiêu.
- Các khoản thu, chi thực tế.

Thực hiện kiểm thử các chức năng:

- Thêm, sửa, xóa khoản thu chi.
- Quản lý danh mục.
- Xem báo cáo.
- Đăng nhập, đăng xuất.

Đánh giá kết quả thu được so với yêu cầu đề ra.

4.3. Kết quả thực nghiệm FigmFigma

4.3.1. Thiết kế giao diện trên Figma

Các màn hình được thiết kế gồm:

- Home
- Chi tiêu
- Theo dõi
- Biểu đồ
- Thống kê
- Tài khoản

Hình 4.3.1. Giao diện thiết kế trên Figma

4.3.2. Giao diện quản lý chi tiêu

Quản lý chi tiêu:

- Tên tài khoản
- Số tiền(VND)
- Chọn danh mục
- Ngày,thánh,năm(mm/dd/yyyy)

Hình 4.3.2 Giao diện chi tiêu

4.3.3 Giao diện biểu đồ chi tiêu Mô tả biểu đồ:

- **Trục hoành (X):** Tháng (Tháng 1, Tháng 2, Tháng 3, Tháng 4, Tháng 5, Tháng 6).
- **Trục tung (Y):** Số Tiền (VND), thang đo từ 0 đến 8,000,000.
- **Chú thích:** Chi Tiêu (VND).

Hình 4.3.3 Giao diện biểu đồ chi tiêu

4.3.4. Giao diện theo dõi giao dịch **Lịch sử giao dịch:**

- Ngày
- Mô tả
- Số tiền (VND)
- Loại

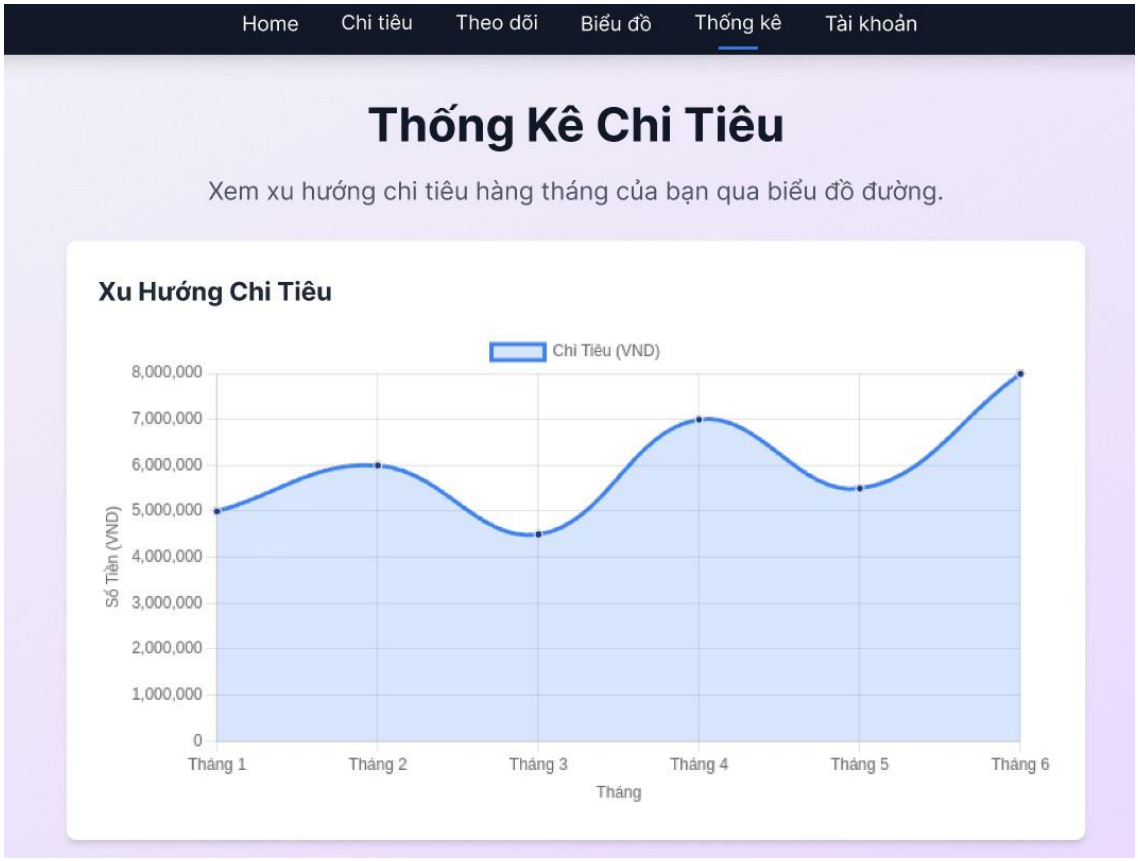
Hình 4.3.4 Giao diện theo dõi giao dịch

4.3.5. Quản lý thống kê chi tiêu

Mô tả biểu đồ thông kê chi tiêu:

Biểu đồ đường trên thể hiện xu hướng chi tiêu hàng tháng của người dùng từ Tháng 1 đến Tháng 6. Trục hoành hiển thị các tháng, trục tung thể hiện số tiền (VND) với thang đo từ 0 đến 8,000,000 VND.

- Chú thích: Chi Tiêu (VND).



Hình 4.3.5 Giao diện quản lý thống kê chi tiêu

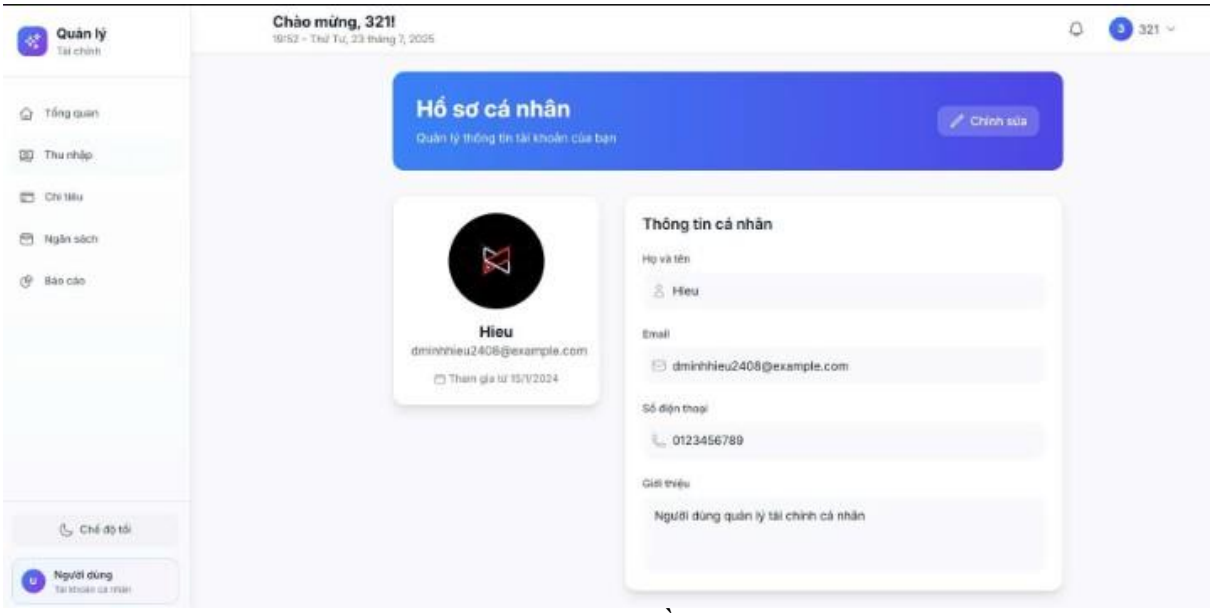
4.4 Kết quả thực nghiệm của website

4.4.1 Giao diện hồ sơ cá nhân

Nội dung hiển thị:

- Thông tin cá nhân: Họ và tên, email, số điện thoại, giới thiệu.
- Ảnh đại diện của người dùng.

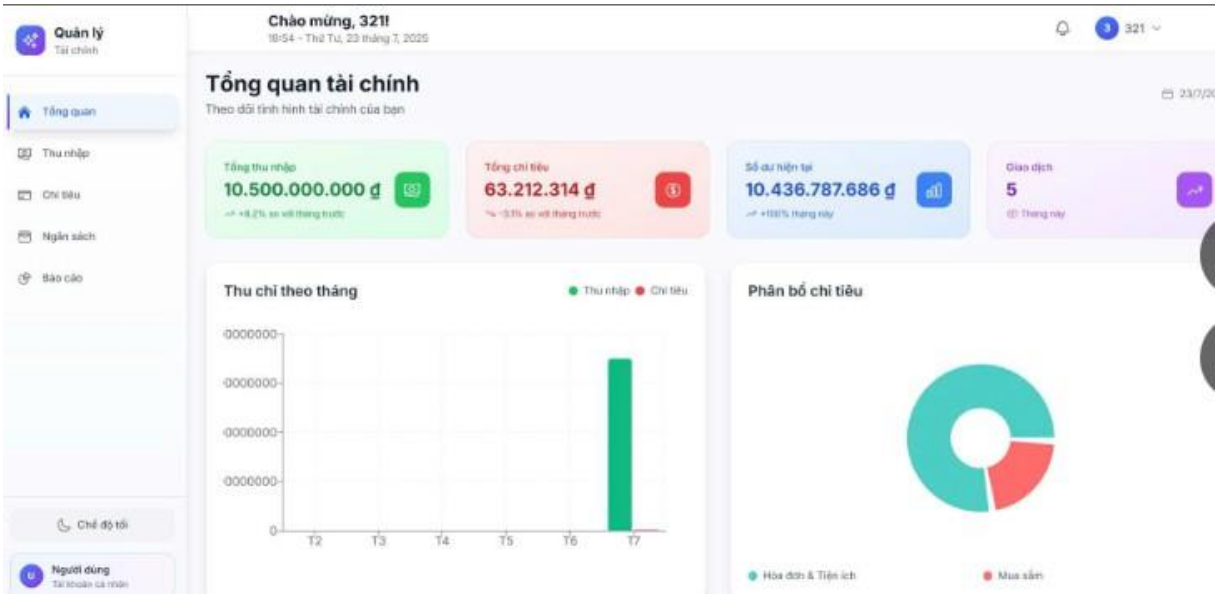
- Nút “Chỉnh sửa” để cập nhật thông tin cá nhân.
- Menu bên trái gồm: Tổng quan, Thu nhập, Chi tiêu, Ngân sách, Báo cáo, và Tài khoản cá nhân.



Hình 4.4.1 Giao diện hồ sơ cá nhân của website

4.4.2 Giao diện tổng quan tài chính **Nội dung hiển thị:**

- Tổng thu nhập: 10.500.000.000 ₫ (+42.3% so với tháng trước)
- Tổng chi tiêu: 63.212.314 ₫ (-31% so với tháng trước)
- Số dư hiện tại: 10.436.787.686 ₫ (+160% tháng này)
- Giao dịch: 5 (tháng này)
- Biểu đồ cột thu chi theo tháng (T3 – T7).
- Biểu đồ donut phân bổ chi tiêu theo danh mục (Hóa đơn & Tiện ích, Mua sắm).

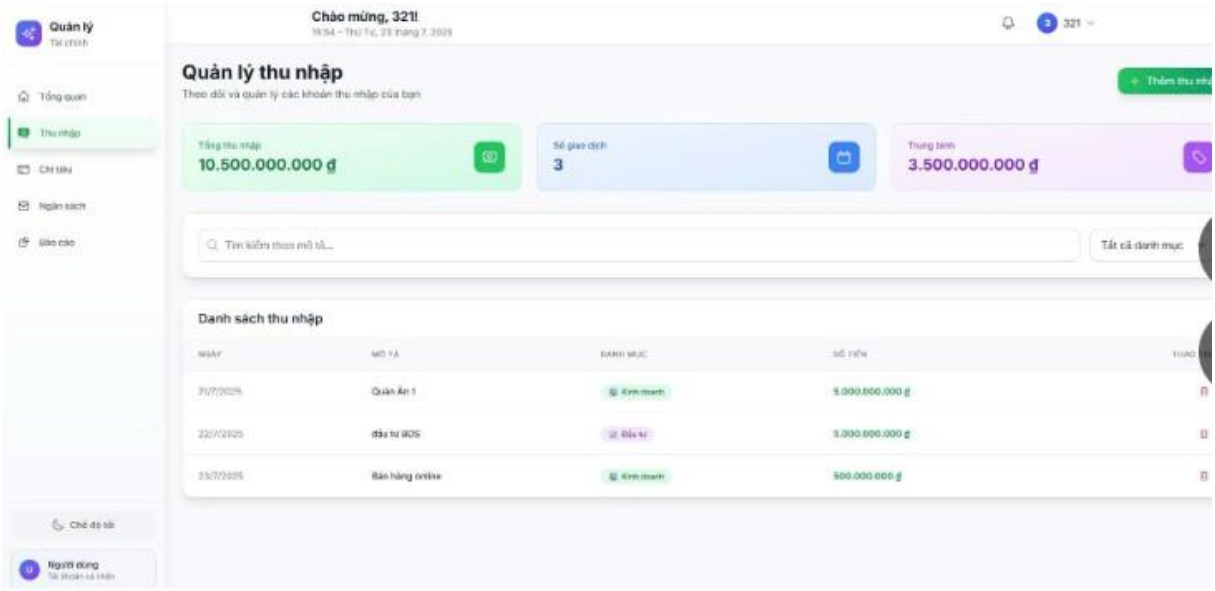


Hình 4.4.2 Giao diện trang tổng quan tài chính, hiển thị số liệu thu nhập, chi tiêu, số dư và biểu đồ phân tích dữ liệu trực quan

4.4.3 Giao diện quản lý thu nhập

Nội dung hiển thị:

- **Tổng thu nhập:** 10.500.000.000 ₫
- **Số giao dịch:** 3
- **Thu nhập trung bình:** 3.500.000.000 ₫
- Danh sách các khoản thu nhập gồm:
  - **Ngày:** 23/7/2025, 22/7/2025
  - **Mô tả:** Quán Ăn 1, Đầu tư BĐS, Bán hàng online
  - **Danh mục:** Kinh doanh, Bất động sản
  - **Số tiền:** từ 500.000.000 ₫ đến 5.000.000.000 ₫
  - **Thao tác:** Nút xóa từng mục thu nhập

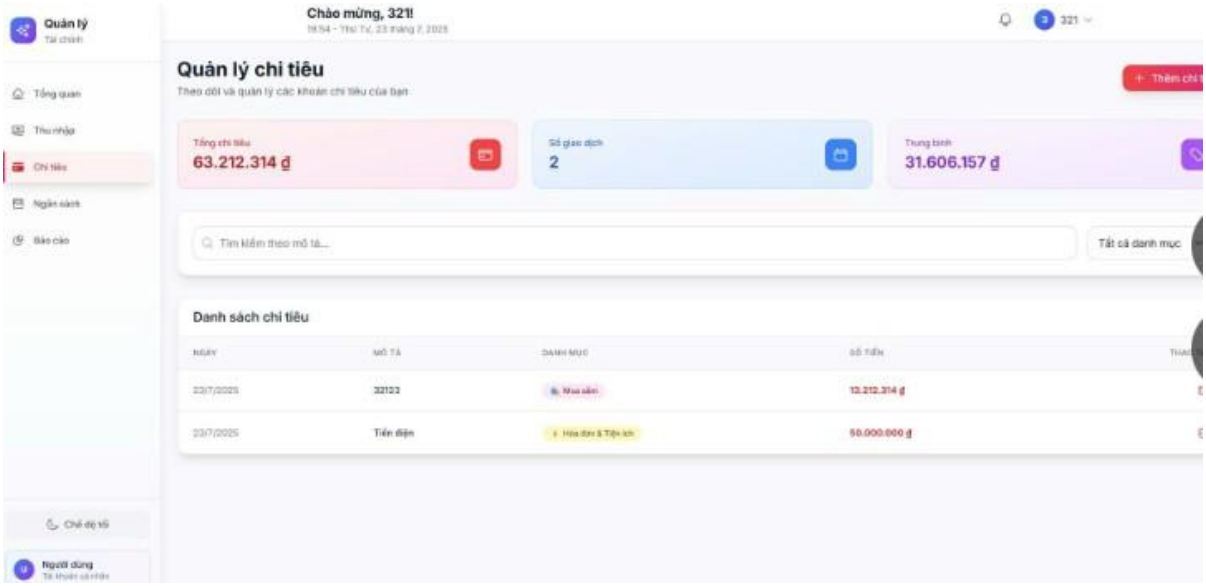


Hình 4.4.3 Giao diện trang quản lý thu nhập, hiển thị danh sách các khoản thu nhập của người dùng kèm mô tả, danh mục và số tiền.

4.4.4. Giao diện quản lý chi tiêu

Nội dung hiển thị:

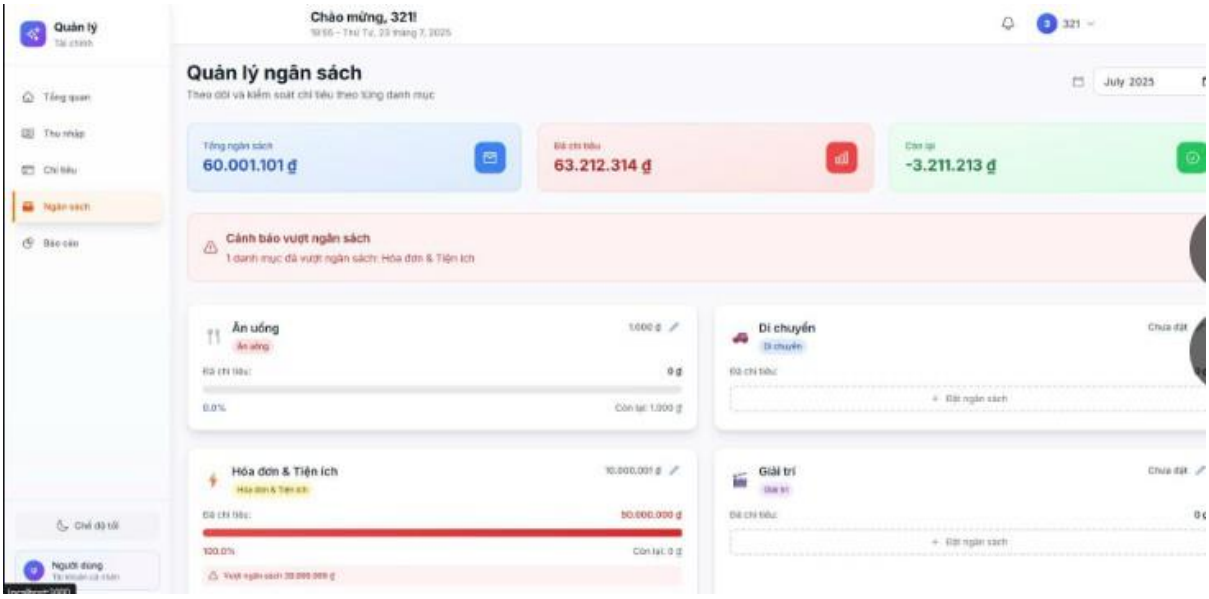
- **Tổng chi tiêu:** 63.212.314 ₫
- **Số giao dịch:** 2
- **Chi tiêu trung bình:** 31.606.157 ₫
- Danh sách các khoản chi tiêu gồm:
  - **Ngày:** 23/7/2025
  - **Mô tả:** 32123, Tiền điện
  - **Danh mục:** Mua sắm, Hóa đơn & Tiện ích
  - **Số tiền:** 63.212.314 ₫ và 50.000.000 ₫
  - **Thao tác:** Nút xóa từng khoản chi



Hình 4.4.4 Giao diện trang quản lý chi tiêu, hiển thị danh sách các khoản chi tiêu với mô tả, danh mục và số tiền chi tiết.

4.4.5 Giao diện quản lý ngân sách **Nội dung hiển thị:**

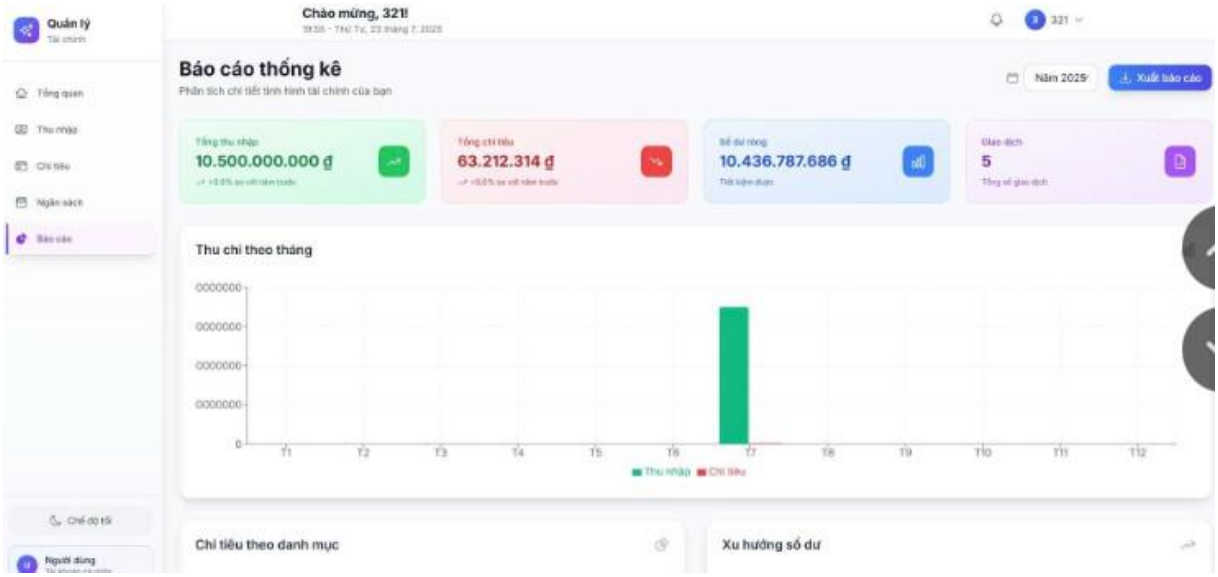
- **Tổng ngân sách:** 60.001.101 ₫
- **Đã chi tiêu:** 63.212.314 ₫
- **Còn lại:** -3.211.213 ₫ (âm, vượt ngân sách)
- Thông báo cảnh báo vượt ngân sách cho danh mục **Hóa đơn & Tiện ích**.
- Danh sách các danh mục ngân sách gồm:
  - **Ăn uống:** 1.000 ₫, còn lại 1.000 ₫
  - **Hóa đơn & Tiện ích:** đã chi hết 50.000.000 ₫, còn lại 0 ₫
  - **Đi chuyển:** chưa đặt ngân sách
  - **Giải trí:** chưa đặt ngân sách



Hình 4.4.5 Giao diện trang quản lý ngân sách, hiển thị tổng ngân sách, số tiền đã chi tiêu, số dư còn lại và cảnh báo vượt ngân sách theo từng danh mục.

4.4.6 Giao diện báo cáo thống kê Nội dung hiển thị:

- **Tổng thu nhập:** 10.500.000.000 ₫ (+0.35% so với tháng trước)
- **Tổng chi tiêu:** 63.212.314 ₫ (-0.02% so với tháng trước)
- **Số dư ròng:** 10.436.787.686 ₫ (tích lũy được)
- **Giao dịch:** 5 (tổng số giao dịch)
- **Biểu đồ cột thu chi theo tháng** (T1 đến T12), thể hiện thu nhập và chi tiêu từng tháng.
- Các phần khác (bên dưới – chưa hiển thị đầy đủ trong hình): Chi tiêu theo danh mục, Xu hướng số dư.



Hình 4.4.6. Giao diện trang báo cáo thống kê, thể hiện biểu đồ thu chi theo tháng cùng các thông tin tổng quan về tình hình tài chính của người dùng.

5. Quản lý dự án trên Jira

Lập kế hoạch Sprint và phân công công việc:

Trong quá trình phát triển phần mềm, nhóm dự án chia toàn bộ lộ trình thành nhiều Sprint – mỗi Sprint thường kéo dài từ 1 đến 2 tuần, tùy vào quy mô và yêu cầu cụ thể. Mỗi Sprint được xem là một giai đoạn phát triển độc lập, với mục tiêu rõ ràng và sản phẩm có thể kiểm thử được. Dựa vào Product Backlog – danh sách tập hợp tất cả các tính năng, yêu cầu và đề xuất của dự án, nhóm lựa chọn các mục tiêu ưu tiên để đưa vào Sprint Backlog.

Mỗi Issue đều được phân công rõ ràng cho các thành viên trong nhóm kèm theo thời hạn hoàn thành (deadline), mức độ ưu tiên (Priority), và có thể đính kèm tài liệu, hình ảnh liên quan để thuận tiện cho việc thực hiện. Việc phân công này giúp đảm bảo trách nhiệm cá nhân và tối ưu hiệu suất làm việc trong nhóm.

Theo dõi tiến độ

Để trực quan hóa quy trình làm việc và kiểm soát tiến độ, nhóm sử dụng Scrum Board trong Jira, với các cột chính: To Do → In Progress → Done. Mỗi thành viên sẽ chủ động cập nhật trạng thái công việc của mình khi bắt đầu hoặc hoàn thành một tính năng, giúp toàn nhóm nắm được tiến độ tổng thể của Sprint.

Công cụ Jira còn hỗ trợ tự động tạo các báo cáo trực quan như Burndown Chart, thể hiện số lượng công việc còn lại theo thời gian, giúp nhóm dễ dàng theo dõi tiến độ và phát hiện sớm các nguy cơ chậm trễ. Ngoài ra, sau mỗi Sprint, Jira cung cấp Sprint Report để tổng hợp kết quả đạt được, so sánh giữa kế hoạch và thực tế, từ đó nhóm có

thể rút kinh nghiệm, điều chỉnh chiến lược và ước lượng tốt hơn cho các Sprint tiếp theo.

Việc kết hợp quy trình Scrum và công cụ Jira giúp nhóm làm việc hiệu quả, minh bạch và kiểm soát chất lượng sản phẩm một cách chuyên nghiệp.

Phân công nhiệm vụ

Dự án web được hoàn thành trong 5 Sprint với mỗi nhiệm vụ phân công cho từng thành viên cụ thể như sau:

Key	Summary	Status	Comments	Sprint	Assignee
SCRUM-1	Lên kế hoạch và phân tích yêu cầu	DONE	1 comment		MaiHongLoi
SCRUM-2	Thiết kế hệ thống và giao diện	DONE	Add comment	SCRUM Spri...	MaiHongLoi
SCRUM-3	Lập kế hoạch dự án Website Quản Lý Tài Chính	DONE	Add comment	SCRUM Sprint 1	Huynh Quoc Kiet
SCRUM-4	Thu thập yêu cầu người dùng	DONE	Add comment	SCRUM Sprint 1	Dminnhhieu2408
SCRUM-5	Thiết kế sơ đồ Use Case	DONE	Add comment	SCRUM Sprint 1	MaiHongLoi
SCRUM-6	Tạo cơ sở dữ liệu MySQL	DONE	Add comment	SCRUM Sprint 2	Huynh Quoc Kiet
SCRUM-7	Viết API đăng ký user	DONE	Add comment	SCRUM Sprint 3	Huynh Quoc Kiet
SCRUM-8	Viết API đăng nhập user	DONE	Add comment	SCRUM Sprint 3	Huynh Quoc Kiet
SCRUM-9	Thiết kế giao diện login	DONE	Add comment	SCRUM Sprint 2	Dminnhhieu2408

Hình 5. Board Jira quản lý dự án

CHƯƠNG 5 : ĐÁNH GIÁ VÀ KẾT LUẬN

5.1 . Đánh giá kết quả

Sau thời gian thực hiện, nhóm đã hoàn thành xây dựng website quản lý tài chính với các chức năng chính:

- Đăng ký, đăng nhập tài khoản người dùng.
- Ghi nhận các khoản thu, khoản chi.
- Hiển thị danh sách thu chi theo ngày, tháng.
- Thống kê và vẽ biểu đồ chi tiêu theo tháng, giúp người dùng dễ dàng theo dõi và cân đối tài chính.
- Giao diện website đơn giản, dễ sử dụng, thân thiện với người dùng.

Website được phát triển theo kiến trúc client/server, sử dụng các công nghệ như HTML, CSS, JavaScript, Node.js, Express, và cơ sở dữ liệu MongoDB. Quy trình phát triển được quản lý qua Jira, thiết kế giao diện trên Figma, lập trình và quản lý mã nguồn trên GitHub, đảm bảo tính khoa học và chuyên nghiệp.

5.2 . Khó khăn

Một trong những khó khăn lớn nhất là việc lựa chọn công nghệ phù hợp và làm chủ chúng trong thời gian ngắn. Đề tài yêu cầu áp dụng nhiều công cụ và công nghệ hiện đại như Docker, Swagger, GitHub Actions, Postman, kiến trúc RESTful API,… Trong khi đó, phần lớn thành viên trong nhóm chưa từng sử dụng những công cụ này trước đây. Vì vậy, nhóm phải tự học, tra cứu tài liệu, xem hướng dẫn từ nhiều nguồn khác nhau, vừa học vừa áp dụng vào dự án thực tế. Điều này gây áp lực khá lớn về mặt thời gian và dễ dẫn đến sai sót kỹ thuật ở các giai đoạn đầu.

Ngoài ra, nhóm cũng gặp thách thức trong việc phối hợp làm việc nhóm và quản lý tiến độ. Mỗi thành viên đều có lịch học và công việc cá nhân khác nhau, nên không dễ để sắp xếp thời gian làm việc chung. Dù đã sử dụng Jira để phân chia công việc và quản lý sprint, nhưng vẫn có thời điểm tiến độ bị chậm hoặc trễ do không đồng bộ được nhịp làm việc giữa các thành viên.

5.3 . Bài học rút ra

Thông qua dự án này, nhóm đã rút ra nhiều bài học kinh nghiệm quý báu:

- Củng cố vững chắc kiến thức về lập trình frontend và backend, kết nối cơ sở dữ liệu MongoDB.

- Rèn luyện kỹ năng làm việc nhóm, phân công công việc hợp lý, quản lý tiến độ bằng Jira hiệu quả hơn.
- Học cách sử dụng Figma để thiết kế giao diện chuyên nghiệp trước khi code.
- Nâng cao kỹ năng git – GitHub, giúp quản lý source code tốt và làm việc nhóm dễ dàng hơn.
- Nhận thức rõ tầm quan trọng của việc lên kế hoạch và thiết kế hệ thống chi tiết trước khi bắt tay vào lập trình.
- Hiểu rõ quy trình phát triển phần mềm theo hướng công nghiệp, từ phân tích yêu cầu, thiết kế, triển khai đến kiểm thử và đánh giá.

5.4 Kết luận

Đề tài " Xây dựng website quản lý tài chính" đã đạt được mục tiêu ban đầu, xây dựng thành công một hệ thống quản lý thu chi dễ sử dụng, trực quan và tiện lợi. Tuy vẫn còn một số hạn chế, nhưng dự án đã giúp nhóm củng cố kiến thức lập trình web, quy trình phát triển phần mềm, cũng như rèn luyện kỹ năng làm việc nhóm – những yếu tố cần thiết để đáp ứng yêu cầu công việc trong tương lai.

Trong thời gian tới, nhóm mong muốn tiếp tục phát triển website này bằng cách:

- Thêm chức năng quản lý tiết kiệm, quản lý đầu tư.
- Cải thiện giao diện responsive phù hợp cho điện thoại và máy tính bảng.
- Triển khai website trên các hosting chuyên nghiệp để phục vụ người dùng thực tế.

TÀI LIỆU THAM KHẢO

1 . Atlassian. (2024). ***Jira Software Documentation*****. Truy cập tại:https://support.atlassian.com/jira-software-cloud/**

2. Figma. (2024). ***Getting Started with Figma*****. Truy cập tại:https://www.figma.com/resources/learn-design/**

3. Atlassian. (2023). ***Agile project management with Jira Software*****. Truy cập tại:https://www.atlassian.com/agile/project-management**

4. Figma. (2022). ***Designing with Figma: Introduction*****. Truy cập tại: https://www.figma.com/resources/learn-design/designing-with-figma-introduction**

/