

**TRƯỜNG ĐẠI HỌC CÔNG NGHIỆP TP HỒ CHÍ MINH
KHOA CÔNG NGHỆ THÔNG TIN**



**BÁO CÁO CUỐI KÌ
MÔN HỌC: PHÁT TRIỂN ỨNG DỤNG**

**ĐỀ TÀI: WEBSITE QUẢN LÝ HỒ SƠ Y TẾ TRỰC
TUYẾN - MEDICAL RECORDS**

Người thực hiện: **TRƯƠNG CÔNG ĐẠT - 22685561**

NGUYỄN GIA LÂM - 22685611

PHAN TẤN TÀI - 22684181

Lớp : **420300350103**

Khoá : **18**

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

DH TRẦN TẤN THÀNH

THÀNH PHỐ HỒ CHÍ MINH, NĂM 2025

**TRƯỜNG ĐẠI HỌC CÔNG NGHIỆP TP HỒ CHÍ MINH
KHOA CÔNG NGHỆ THÔNG TIN**



**BÁO CÁO CUỐI KÌ
MÔN HỌC: PHÁT TRIỂN ỨNG DỤNG**

**ĐỀ TÀI: WEBSITE QUẢN LÝ HỒ SƠ Y TẾ TRỰC
TUYẾN - MEDICAL RECORDS**

Người thực hiện: **TRƯƠNG CÔNG ĐẠT - 22685561**

NGUYỄN GIA LÂM - 22685611

PHAN TẤN TÀI - 22684181

Lớp : 420300350103

Khoá : 18

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

DH TRẦN TẤN THÀNH

THÀNH PHỐ HỒ CHÍ MINH, NĂM 2025

LỜI CẢM ƠN

Trước hết, em xin gửi lời cảm ơn chân thành và sâu sắc đến THS. Trương Vĩnh Linh, người đã luôn tận tâm hướng dẫn, hỗ trợ và đồng hành cùng chúng em trong suốt quá trình thực hiện đồ án. Sự chỉ dẫn tận tình, những góp ý quý báu và tinh thần khích lệ của thầy/cô không chỉ giúp chúng em vượt qua những khó khăn mà còn là động lực lớn lao để em nỗ lực hoàn thành tốt đồ án của mình. Chúng em xin chân thành cảm ơn thầy đã luôn kiên nhẫn, tận tụy và tạo mọi điều kiện thuận lợi để chúng em có thể hoàn thiện đồ án này một cách trọn vẹn nhất.

PHẦN ĐÁNH GIÁ CỦA GIẢNG VIÊN

Tp. Hồ Chí Minh, ngày tháng năm

(kí và ghi họ tên)

THÔNG TIN NHÓM

1. Thông tin các thành viên trong nhóm :

- Thành viên 1:

- + Họ và tên: Trương Công Đạt
- + MSSV: 22685561
- + Email: tdat4926@gmail.com

- Thành viên 2:

- + Họ và tên: Nguyễn Gia Lâm
- + MSSV: 22685611
- + Email: lam2004ha@gmail.com

- Thành viên 3:

- + Họ và tên: Phan Tấn Tài
- + MSSV: 22684181
- + Email: tide.tantai@gmail.com

2. Phân công công việc:

| Họ và tên | Công việc phụ trách |
|-----------------|--|
| Trương Công Đạt | - Hiện thực Front-end và back-end các chức năng của role: Patient - Viết và chỉnh sửa báo cáo - Làm Powerpoint |
| Nguyễn Gia Lâm | - Hiện thực Front-end và back-end các chức năng của role: Doctor và chatbox - Viết và chỉnh sửa báo cáo - Làm Powerpoint |
| Phan Tấn Tài | - Hiện thực Front-end và back-end các chức năng của role: Admin - Xây dựng test case cho project, viết script chạy tự động - Viết và chỉnh sửa báo cáo |

MỤC LỤC

| | |
|---|----|
| CHƯƠNG 1. TỔNG QUAN VỀ DỰ ÁN..... | 1 |
| 1. Giới thiệu về dự án..... | 1 |
| 2. Lý do chọn đề tài | 1 |
| 3. Mục tiêu của dự án | 2 |
| 4. Các vấn đề cần giải quyết:..... | 3 |
| CHƯƠNG 2. CHI TIẾT DỰ ÁN..... | 5 |
| 1. Phân tích yêu cầu | 5 |
| 2. Thiết kế hệ thống | 7 |
| 2.1. <i>Usecase Diagram</i> | 7 |
| 2.2. <i>Database diagram</i> | 8 |
| 3. Công cụ và công nghệ sử dụng | 9 |
| 3.1. <i>Backend</i> | 9 |
| 3.2. <i>Frontend</i> | 10 |
| 3.3. <i>Database</i> | 10 |
| 3.4. <i>Development Tools</i> | 10 |
| 3.5. <i>Deployment</i> | 10 |
| 4. Triển khai ứng dụng | 11 |
| 4.1. <i>Các bước triển khai</i> | 11 |
| 5. Kiểm thử..... | 13 |
| 5.1 <i>Thực hiện kiểm thử chức năng (Functional Testing)</i> | 13 |
| 5.2 <i>Thực hiện kiểm thử hiệu năng (Performance Testing)</i> | 16 |
| CHƯƠNG 3. KẾT QUẢ, HẠN CHẾ VÀ HƯỚNG PHÁT TRIỂN | 19 |
| 1. Kết quả đạt được | 19 |
| 2. Hạn chế và hướng phát triển | 27 |

CHƯƠNG 1. TỔNG QUAN VỀ DỰ ÁN

1. Giới thiệu về dự án

Medical Records là một nền tảng quản lý hồ sơ y tế thông minh, được thiết kế để kết nối bệnh nhân và bác sĩ trong kỷ nguyên số. Với giao diện trực quan và dễ sử dụng, hệ thống cho phép bệnh nhân dễ dàng theo dõi lịch sử sức khỏe của mình thông qua việc lưu trữ và phân tích các chỉ số xét nghiệm máu quan trọng. Bác sĩ có thể truy cập và đánh giá dữ liệu sức khỏe của bệnh nhân một cách nhanh chóng, đồng thời gửi thông báo và khuyến nghị y tế. Đặc biệt, hệ thống được tích hợp công nghệ AI hiện đại, giúp cung cấp thông tin tư vấn y tế chính xác và hỗ trợ phân tích dữ liệu sức khỏe. Medical Records không chỉ là một công cụ quản lý hồ sơ y tế thông thường, mà còn là một giải pháp toàn diện nhằm nâng cao chất lượng chăm sóc sức khỏe, tăng cường tương tác giữa bác sĩ và bệnh nhân, đồng thời góp phần vào việc số hóa ngành y tế.

2. Lý do chọn đề tài

- *Nhu cầu thực tế:*

- + Xu hướng số hóa trong lĩnh vực y tế đang phát triển mạnh mẽ
- + Nhu cầu quản lý hồ sơ y tế hiệu quả ngày càng tăng
- + Đòi hỏi về tính minh bạch và dễ truy xuất thông tin
- + Cần cải thiện chất lượng chăm sóc sức khỏe

- *Tính khả thi:*

- + Công nghệ hiện đại cho phép triển khai hệ thống
- + Có thể tích hợp với các hệ thống y tế hiện có
- + Khả năng mở rộng và phát triển trong tương lai
- + Nguồn lực và công nghệ sẵn có để thực hiện

- Lợi ích mang lại:

- + Cải thiện chất lượng chăm sóc sức khỏe
- + Tiết kiệm thời gian và chi phí cho cả bệnh nhân và bác sĩ
- + Nâng cao hiệu quả quản lý thông tin y tế
- + Hỗ trợ ra quyết định dựa trên dữ liệu
- + Tăng cường sự tương tác giữa bác sĩ và bệnh nhân

- Tính đổi mới:

- + Ứng dụng công nghệ AI trong lĩnh vực y tế
- + Tích hợp các công cụ phân tích dữ liệu hiện đại
- + Cải tiến phương thức quản lý hồ sơ y tế
- + Tạo ra giải pháp toàn diện cho việc chăm sóc sức khỏe

- Tiềm năng phát triển:

- + Có thể mở rộng sang các lĩnh vực y tế khác
- + Khả năng tích hợp với các hệ thống y tế lớn
- + Tiềm năng phát triển ứng dụng di động
- + Cơ hội mở rộng thị trường trong tương lai

3. Mục tiêu của dự án

Và đây là những mục tiêu mà nhóm chúng em đã đặt ra khi xây dựng website Medical Records

- Mục tiêu 1: Xây dựng một hệ thống quản lý hồ sơ y tế tập trung và an toàn cho phép người dùng (Bệnh nhân, Bác sĩ, Quản trị viên) truy cập và quản lý thông tin sức khỏe hiệu quả. Cụ thể như sau:

+ Bệnh nhân có thể đăng ký, đăng nhập, quản lý thông tin cá nhân, tự nhập và theo dõi lịch sử kết quả xét nghiệm máu (bao gồm 10 chỉ số: HGB, RBC, WBC, PLT, HCT, Glucose, Creatinine, ALT, Cholesterol, CRP).

+ Bác sĩ có thể tìm kiếm bệnh nhân, xem chi tiết hồ sơ bệnh án, bao gồm cả dạng bảng và biểu đồ trực quan hóa dữ liệu theo thời gian.

+ Quản trị viên có thể quản lý tài khoản người dùng (thay đổi vai trò, đặt lại mật khẩu) và xuất dữ liệu hệ thống.

+ Đảm bảo an toàn dữ liệu thông qua mã hóa mật khẩu, bảo vệ chống CSRF, và phân quyền truy cập dựa trên vai trò người dùng.

- **Mục tiêu 2:** Nâng cao trải nghiệm người dùng thông qua giao diện trực quan, dễ sử dụng và cung cấp các công cụ hỗ trợ thông minh. Cụ thể như sau:

+ Cung cấp biểu đồ trực quan (sử dụng Plotly.js) giúp bệnh nhân và bác sĩ dễ dàng theo dõi xu hướng thay đổi của các chỉ số xét nghiệm theo thời gian.

+ Triển khai hệ thống thông báo cho phép bác sĩ gửi khuyến nghị, nhận xét đến bệnh nhân và bệnh nhân có thể xem, quản lý các thông báo này.

+ Tích hợp trợ lý AI (Gemini) để cung cấp thông tin, giải đáp thắc mắc về y tế cho người dùng, có khả năng tham khảo dữ liệu bệnh án cá nhân (nếu được phép) để đưa ra tư vấn phù hợp hơn.

+ Cho phép bệnh nhân tìm kiếm thông tin bác sĩ trong hệ thống.

+ Cho phép bác sĩ tải xuống hồ sơ bệnh án của bệnh nhân dưới dạng file CSV và quản trị viên xuất dữ liệu người dùng, bệnh án dưới dạng file Excel.

4. Các vấn đề cần giải quyết:

- **Vấn đề quản lý thông tin:**

+ Hồ sơ y tế truyền thống dễ bị mất, hư hỏng

- + Khó khăn trong việc tìm kiếm và truy xuất thông tin
- + Khó khăn trong việc chia sẻ thông tin giữa các bên liên quan

- Vấn đề giao tiếp:

- + Bác sĩ gặp khó khăn trong việc theo dõi và nhắc nhở bệnh nhân
- + Khó khăn trong việc cập nhật thông tin, tình hình sức khỏe cho bệnh nhân

- Vấn đề phân tích dữ liệu:

- + Khó khăn trong việc theo dõi xu hướng các chỉ số sức khỏe
- + Thiếu công cụ trực quan hóa dữ liệu
- + Khó khăn trong việc đưa ra quyết định dựa trên dữ liệu lịch sử

- Vấn đề bảo mật:

- + Đảm bảo tính bảo mật của thông tin y tế nhạy cảm
- + Tuân thủ các quy định về bảo vệ dữ liệu cá nhân
- + Kiểm soát quyền truy cập thông tin
- + Bảo vệ dữ liệu khỏi các rủi ro bảo mật

CHƯƠNG 2. CHI TIẾT DỰ ÁN

1. Phân tích yêu cầu

Sau đây là các yêu cầu chức năng (*Functional Requirements - FR*) và yêu cầu phi chức năng (*Non-Functional Requirements – NFR*) của dự án xây dựng website Medical Records.

- *Yêu cầu chức năng (Functional Requirements - FR)*

+ FR1: Quản lý tài khoản người dùng - Hệ thống cần cung cấp các chức năng liên quan đến tài khoản người dùng, bao gồm đăng ký tài khoản mới, đăng nhập, đăng xuất, cập nhật thông tin cá nhân, và có thể bao gồm cả đặt lại mật khẩu.

+ FR2: Chức năng cho Bệnh nhân - Đây là các tính năng dành riêng cho người dùng có vai trò là bệnh nhân, ví dụ như thêm, xem, chỉnh sửa, xóa hồ sơ bệnh án (kết quả xét nghiệm), xem biểu đồ trực quan hóa dữ liệu sức khỏe, nhận thông báo từ bác sĩ và tìm kiếm thông tin bác sĩ.

+ FR3: Chức năng cho Bác sĩ - Đây là các tính năng dành riêng cho người dùng có vai trò là bác sĩ, ví dụ như tìm kiếm bệnh nhân, xem hồ sơ bệnh án và biểu đồ của bệnh nhân, gửi thông báo hoặc khuyến nghị cho bệnh nhân, và có thể là tải xuống dữ liệu bệnh nhân.

+ FR4: Chức năng cho Quản trị viên - Đây là các tính năng dành riêng cho người dùng có vai trò quản trị hệ thống, ví dụ như quản lý tất cả tài khoản người dùng (xem, thêm, sửa, xóa), cập nhật vai trò người dùng (phân quyền), và các công việc quản trị hệ thống khác.

- *Yêu cầu phi chức năng (Non-Functional Requirements – NFR)*

+ NFR1: Tính Bảo mật (Security) - Hệ thống phải đảm bảo an toàn cho dữ liệu người dùng, đặc biệt là thông tin y tế nhạy cảm. Điều này bao gồm bảo vệ chống truy cập trái

phép, mã hóa dữ liệu, bảo vệ chống tấn công mạng, và xác thực người dùng an toàn.

+ NFR2: Tính Khả dụng (Usability) - Hệ thống phải dễ sử dụng, giao diện trực quan và thân thiện với người dùng ở tất cả các vai trò. Người dùng có thể dễ dàng tìm thấy các chức năng cần thiết và hoàn thành tác vụ một cách hiệu quả.

+ NFR3: Hiệu năng (Performance) - Hệ thống phải hoạt động nhanh chóng và hiệu quả, xử lý các yêu cầu của người dùng mà không bị chậm trễ đáng kể, ngay cả khi có nhiều người truy cập cùng lúc hoặc lượng dữ liệu lớn.

+ NFR4: Tính Tin cậy (Reliability) - Hệ thống phải hoạt động ổn định và đáng tin cậy, ít gặp lỗi hoặc sự cố. Khi có lỗi xảy ra, hệ thống cần có cơ chế phục hồi hoặc xử lý lỗi phù hợp để giảm thiểu ảnh hưởng đến người dùng.

+ NFR5: Khả năng Bảo trì (Maintainability) - Mã nguồn và cấu trúc hệ thống cần được tổ chức tốt, dễ hiểu, giúp cho việc sửa lỗi, cập nhật và mở rộng các tính năng trong tương lai trở nên thuận tiện và ít tốn kém thời gian, công sức.

+ NFR6: Khả năng Mở rộng (Scalability) - Hệ thống cần có khả năng mở rộng để đáp ứng được số lượng người dùng và dữ liệu ngày càng tăng trong tương lai mà không ảnh hưởng tiêu cực đến hiệu năng.

2. Thiết kế hệ thống

2.1. Usecase Diagram



Hình 2.1. Sơ đồ Usecase diagram của dự án

Đây là use case diagram mô tả các tương tác chính giữa người dùng và hệ thống "Medical Records System". Biểu đồ thể hiện ba tác nhân (actor) chính: Bác sĩ (Doctor), Quản trị viên (Admin), và Bệnh nhân (Patient).

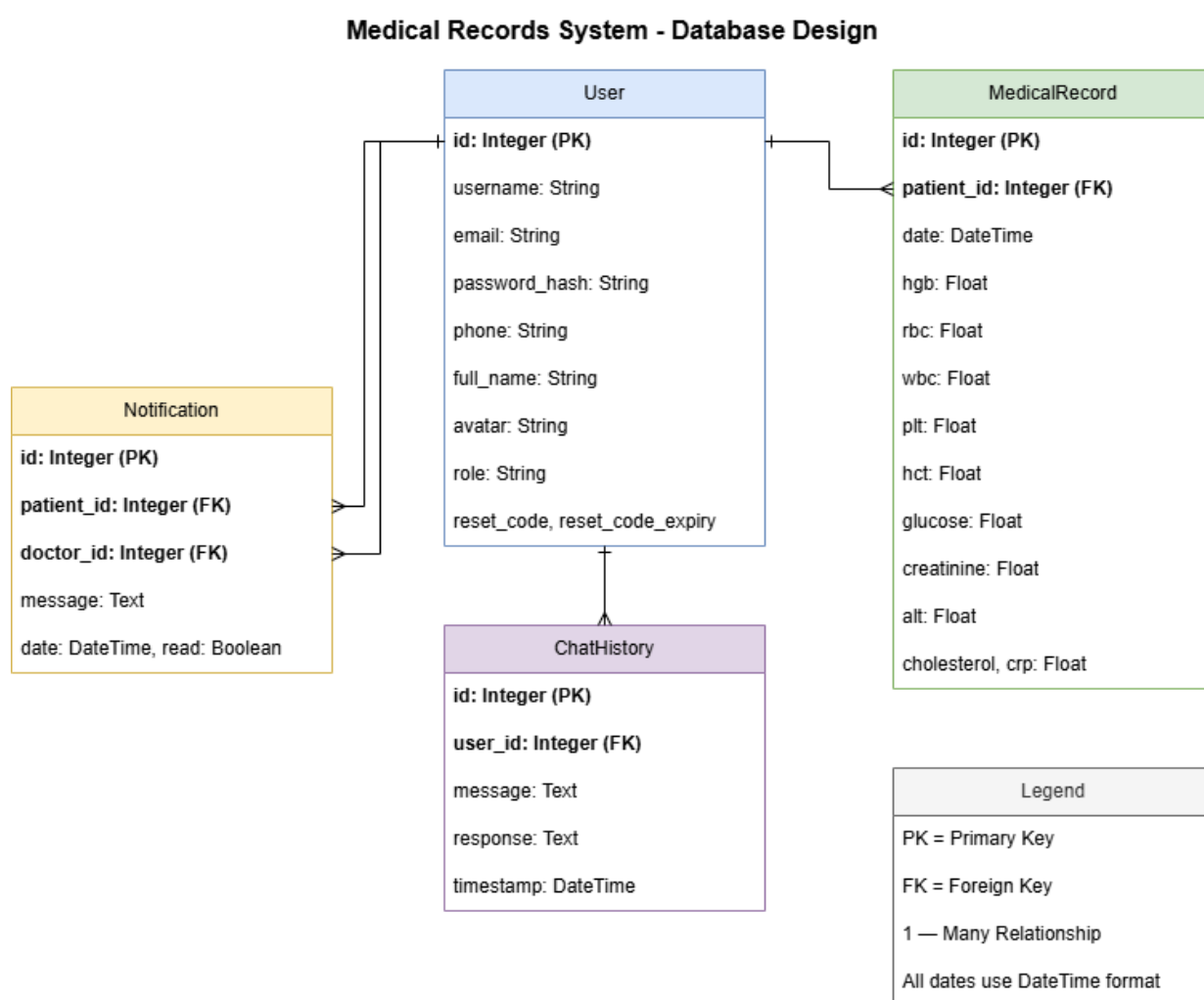
- Bệnh nhân có thể thực hiện các chức năng cốt lõi như đăng ký, đăng nhập, đăng xuất, cập nhật thông tin cá nhân, thêm và xem lịch sử bản ghi y tế, xem biểu đồ trực quan hóa dữ liệu sức khỏe, quản lý thông báo và tìm kiếm thông tin bác sĩ.

- Bác sĩ tương tác với hệ thống để đăng nhập, đăng xuất, cập nhật thông tin cá nhân

(ngầm định), tìm kiếm bệnh nhân, xem bản ghi và biểu đồ y tế của bệnh nhân, gửi thông báo và xuất dữ liệu bệnh nhân.

- Quản trị viên có các quyền hạn liên quan đến quản lý người dùng như đặt lại mật khẩu và cập nhật vai trò người dùng, cùng với các chức năng chung như đăng nhập và đăng xuất.

2.2. Database diagram



Hình 2.2 Database diagram của dự án

- Hệ thống bao gồm bốn bảng chính:

+ User: Lưu trữ thông tin về tất cả người dùng, bao gồm id (khóa chính), username, email, password_hash (mật khẩu đã băm), phone, full_name, avatar (đường dẫn ảnh đại diện), role (vai trò: bệnh nhân, bác sĩ, admin), cùng các trường hỗ trợ đặt lại mật khẩu (reset_code, reset_code_expiry).

+ MedicalRecord: Chứa thông tin chi tiết về các bản ghi y tế của bệnh nhân, với id (khóa chính), patient_id (khóa ngoại, liên kết đến bảng User), date (ngày ghi nhận) và các chỉ số sức khỏe cụ thể như hgb, rbc, glucose, v.v.

+ Notification: Dùng để lưu các thông báo được gửi trong hệ thống. Bảng này có id (khóa chính), patient_id và doctor_id (đều là khóa ngoại, liên kết đến bảng User, cho biết bệnh nhân nhận và bác sĩ gửi), message (nội dung thông báo), date (ngày gửi), và read (trạng thái đã đọc).

+ ChatHistory: Ghi lại lịch sử các cuộc trò chuyện của người dùng (thường là với AI), bao gồm id (khóa chính), user_id (khóa ngoại, liên kết đến bảng User), message (tin nhắn của người dùng), response (phản hồi), và timestamp (thời gian chat).

- Các mối quan hệ chính bao gồm:

+ Một User có thể có nhiều MedicalRecord.

+ Một User (bệnh nhân) có thể nhận nhiều Notification, và một User (bác sĩ) có thể gửi nhiều Notification.

+ Một User có thể có nhiều bản ghi ChatHistory.

3. Công cụ và công nghệ sử dụng

3.1. Backend

- Ngôn ngữ lập trình: Python 3.8+

- Framework: Flask
- ORM: SQLAlchemy
- Authentication: Flask-Login
- Email: Flask-Mail
- Database Migration: Flask-Migrate

3.2. Frontend

- HTML5/CSS3
- JavaScript
- Bootstrap 5
- Plotly.js (trực quan hóa dữ liệu)
- jQuery

3.3. Database

- SQLite (development)
- PostgreSQL (production)

3.4. Development Tools

- IDE: Visual Studio Code
- Version Control: Git
- Browser DevTools: Chrome DevTools

3.5. Deployment

- Render

4. Triển khai ứng dụng

4.1. Các bước triển khai

- Bước 1: Chuẩn bị môi trường:

- + Cài đặt Python 3.8 trở lên
- + Cài đặt Git
- + Cài đặt Visual Studio Code

- Bước 2: Tải mã nguồn:

- + Clone repository từ GitHub
- + Kiểm tra cấu trúc thư mục
- + Xác nhận các file cần thiết

- Bước 3: Thiết lập môi trường ảo:

- + Tạo môi trường ảo Python
- + Kích hoạt môi trường ảo
- + Cài đặt các thư viện cần thiết

- Bước 4: Cấu hình database:

- + Khởi tạo database
- + Tạo tài khoản admin mặc định
- + Kiểm tra kết nối

- Bước 5: Chạy ứng dụng:

- + Khởi động server
- + Kiểm tra các chức năng

- + Xử lý lỗi nếu có

4.2. Cách thức cài đặt

- Cài đặt Python:

- + Tải Python từ python.org
- + Chọn phiên bản 3.8 trở lên
- + Cài đặt với tùy chọn "Add Python to PATH"

- Cài đặt Git:

- + Tải Git từ git-scm.com
- + Cài đặt với các tùy chọn mặc định
- + Kiểm tra cài đặt bằng lệnh "git --version"

- Clone project:

- + Mở Command Prompt hoặc PowerShell
- + Chạy lệnh: `git clone https://github.com/your-repo/medical-records.git`
- + Di chuyển vào thư mục project: `cd medical-records`

- Tạo môi trường ảo:

- + Chạy lệnh: `python -m venv venv`
- + Kích hoạt môi trường:
- + Windows: `venv\Scripts\activate`
- + Linux/Mac: `source venv/bin/activate`

- Cài đặt dependencies:

- + Chạy lệnh: `pip install -r requirements.txt`

- + Đợi quá trình cài đặt hoàn tất
- + Kiểm tra các package đã cài đặt

4.3. Chạy ứng dụng

Có 2 cách để chạy ứng dụng:

- Cách 1: Chạy lệnh ``python run.py``

- + Lệnh này sẽ tự động:
- + Tạo database
- + Tạo các bảng cần thiết
- + Tạo tài khoản admin mặc định (username: admin, password: admin)
- + Ứng dụng sẽ chạy tại <http://localhost:5000>

- Cách 2: chạy file “setup-and-run.ps1”

5. Kiểm thử

5.1 Thực hiện kiểm thử chức năng (Functional Testing)

- Mục tiêu: Kiểm thử chức năng (Functional Testing) được thực hiện để xác minh rằng mọi chức năng của hệ thống "Medical Records System" hoạt động đúng như mong đợi và đáp ứng các yêu cầu đã được định nghĩa. Quá trình này tập trung vào việc kiểm tra các kịch bản sử dụng chính của từng vai trò người dùng (Bệnh nhân, Bác sĩ, Quản trị viên) cũng như các tính năng chung của hệ thống.

- Phương pháp và Công cụ:

- + Các ca kiểm thử chức năng được viết trong file test/test_functional.py.
- + Sử dụng thư viện unittest của Python làm nền tảng để xây dựng các ca kiểm thử.
- + Pytest được sử dụng làm trình chạy kiểm thử, cung cấp kết quả chi tiết về trạng

thái của từng ca kiểm thử.

+ Môi trường kiểm thử được thiết lập với cấu hình riêng (TestingConfig) để đảm bảo tính độc lập và không ảnh hưởng đến dữ liệu thực tế, ví dụ như sử dụng cơ sở dữ liệu SQLite trong bộ nhớ (sqlite:///memory:).

- Phạm vi Kiểm thử:

+ Quản lý Truy cập Hệ thống:

- test_home_page: Kiểm tra khả năng truy cập trang chủ.
- test_user_registration: Kiểm tra quy trình đăng ký tài khoản người dùng mới.
- test_user_login: Kiểm tra quy trình đăng nhập của người dùng.
- test_user_logout: Kiểm tra quy trình đăng xuất của người dùng.

+ Chức năng của Quản trị viên (Admin):

- test_admin_dashboard: Kiểm tra khả năng truy cập vào trang quản trị.
- test_admin_login: Kiểm tra đăng nhập với tài khoản admin (mặc định hoặc được tạo).
- test_admin_manage_doctors: Kiểm tra chức năng quản lý tài khoản bác sĩ (ví dụ: thay đổi vai trò).
- test_admin_manage_patients: Kiểm tra chức năng quản lý tài khoản bệnh nhân.
- test_admin_manage_users: Kiểm tra chức năng quản lý người dùng nói chung.
- test_admin_registration: Kiểm tra quy trình đăng ký hoặc tạo tài khoản admin.

- test_admin_system_settings: Kiểm tra các cài đặt hệ thống mà admin có thể quản lý (nếu có, hoặc các trang quản trị chung).

+ Chức năng của Bác sĩ (Doctor):

- test_doctor_add_comment (hoặc send_notification): Kiểm tra chức năng gửi thông báo/nhận xét cho bệnh nhân.
- test_doctor_login: Kiểm tra đăng nhập với tài khoản bác sĩ.
- test_doctor_registration: Kiểm tra quy trình đăng ký hoặc tạo tài khoản bác sĩ.
- test_doctor_view_patient_records: Kiểm tra chức năng xem hồ sơ bệnh án của bệnh nhân.
- test_doctor_view_patients: Kiểm tra chức năng xem danh sách bệnh nhân.
- Chức năng của Bệnh nhân (Patient):
- test_notifications: Kiểm tra chức năng hiển thị và tương tác với thông báo.
- test_patient_creation (liên quan đến new_record): Kiểm tra chức năng tạo mới hồ sơ bệnh án.
- test_update_profile: Kiểm tra chức năng cập nhật thông tin cá nhân.
- test_view_charts: Kiểm tra chức năng xem biểu đồ sức khỏe.
- test_view_records: Kiểm tra chức năng xem lại các hồ sơ bệnh án đã nhập.

- Kết quả Thực hiện: Tổng cộng 21 ca kiểm thử chức năng đã được thu thập và thực thi. Tất cả 21 ca kiểm thử đều thành công (PASSED), đạt tỷ lệ 100%.

```
(medical-records-env) PS C:\Users\tontide1\Desktop\ADP\Medical-Records> python -- pytest test/test_functional.py -v
===== test session starts =====
platform win32 -- Python 3.10.11, pytest-7.4.0, pluggy-1.6.0 -- C:\Users\tontide1\scoop\apps\miniconda3\current\envs\medical-records-env\python.exe
cachedir: .pytest_cache
rootdir: C:\Users\tontide1\Desktop\ADP
configfile: pytest.ini
plugins: cov=4.1.0
collected 21 items

test\test_functional.py::FunctionalTest::test_admin_dashboard PASSED [ 4%]
test\test_functional.py::FunctionalTest::test_admin_login PASSED [ 9%]
test\test_functional.py::FunctionalTest::test_admin_manage_doctors PASSED [ 14%]
test\test_functional.py::FunctionalTest::test_admin_manage_patients PASSED [ 19%]
test\test_functional.py::FunctionalTest::test_admin_manage_users PASSED [ 23%]
test\test_functional.py::FunctionalTest::test_admin_registration PASSED [ 28%]
test\test_functional.py::FunctionalTest::test_admin_system_settings PASSED [ 33%]
test\test_functional.py::FunctionalTest::test_doctor_add_comment PASSED [ 38%]
test\test_functional.py::FunctionalTest::test_doctor_login PASSED [ 42%]
test\test_functional.py::FunctionalTest::test_doctor_registration PASSED [ 47%]
test\test_functional.py::FunctionalTest::test_doctor_view_patient_records PASSED [ 52%]
test\test_functional.py::FunctionalTest::test_doctor_view_patients PASSED [ 57%]
test\test_functional.py::FunctionalTest::test_home_page PASSED [ 61%]
test\test_functional.py::FunctionalTest::test_notifications PASSED [ 66%]
test\test_functional.py::FunctionalTest::test_patient_creation PASSED [ 71%]
test\test_functional.py::FunctionalTest::test_update_profile PASSED [ 76%]
test\test_functional.py::FunctionalTest::test_user_login PASSED [ 80%]
test\test_functional.py::FunctionalTest::test_user_logout PASSED [ 85%]
test\test_functional.py::FunctionalTest::test_user_registration PASSED [ 90%]
test\test_functional.py::FunctionalTest::test_view_charts PASSED [ 95%]
test\test_functional.py::FunctionalTest::test_view_records PASSED [100%]
```

Hình 2.3 Kết quả chạy test Functional Testing

5.2 Thực hiện kiểm thử hiệu năng (Performance Testing)

- Mục tiêu: Kiểm thử hiệu năng (Performance Testing) được tiến hành nhằm đánh giá tốc độ xử lý, khả năng đáp ứng và sự ổn định của hệ thống "Medical Records System" dưới các điều kiện tải và khối lượng dữ liệu khác nhau. Mục tiêu là đảm bảo hệ thống hoạt động mượt mà, không gây ra sự chậm trễ đáng kể cho người dùng trong quá trình sử dụng các chức năng chính.

- Phương pháp và Công cụ:

- + Các ca kiểm thử hiệu năng được định nghĩa trong file test/test_performance.py.
- + Sử dụng thư viện unittest của Python làm nền tảng, kết hợp với module time để đo lường thời gian thực thi của các tác vụ cụ thể.
- + pytest được sử dụng làm trình chạy kiểm thử, cung cấp báo cáo về trạng thái của từng ca kiểm thử.
- + Mỗi ca kiểm thử hiệu năng được thiết kế để thực hiện một loạt các hành động (ví dụ: tạo nhiều bản ghi, thực hiện nhiều truy vấn) và sau đó so sánh tổng thời gian thực thi

với một ngưỡng thời gian chấp nhận được (ví dụ: `self.assertLess(execution_time, 5.0)`).

- Phạm vi Kiểm thử: Các ca kiểm thử hiệu năng tập trung vào những khía cạnh quan trọng sau của hệ thống:

+ test_user_registration_performance: Đánh giá hiệu suất khi đăng ký một số lượng người dùng nhất định.

+ test_medical_record_creation_performance: Đánh giá hiệu suất khi tạo một số lượng bản ghi y tế.

+ test_bulk_medical_record_creation: Đánh giá hiệu suất khi tạo đồng thời một khối lượng lớn các bản ghi y tế.

+ test_search_performance: Đánh giá tốc độ phản hồi của chức năng tìm kiếm.

+ test_concurrent_user_operations: Mô phỏng và đánh giá hiệu suất khi có nhiều người dùng thực hiện các thao tác trên hệ thống cùng lúc.

+ test_database_query_performance: Kiểm tra hiệu suất của các truy vấn cơ sở dữ liệu phức tạp, bao gồm tìm kiếm với nhiều điều kiện, phân trang và sắp xếp.

+ test_doctor_operations_performance: Đánh giá hiệu suất các thao tác đặc thù của vai trò Bác sĩ (ví dụ: tìm kiếm bệnh nhân, xem hồ sơ, gửi thông báo).

+ test_patient_operations_performance: Đánh giá hiệu suất các thao tác đặc thù của vai trò Bệnh nhân (ví dụ: xem thông tin cá nhân, xem lịch sử bản ghi, cập nhật thông tin).

+ test_admin_operations_performance: Đánh giá hiệu suất các thao tác đặc thù của vai trò Quản trị viên (ví dụ: quản lý người dùng, cập nhật vai trò, đặt lại mật khẩu).

+ test_export_performance: Đánh giá hiệu suất của chức năng xuất dữ liệu (danh sách người dùng, bản ghi y tế).

- Kết quả Thực hiện: Như hình ảnh bạn cung cấp, quá trình chạy kiểm thử hiệu năng bằng lệnh python -m pytest test/test_performance.py -v cho thấy:

+ Tổng cộng 10 ca kiểm thử hiệu năng đã được thu thập và thực thi.

+ Tất cả 10 ca kiểm thử đều thành công (PASSED), đạt tỷ lệ 100%.

```
(medical-records-env) PS C:\Users\tontide1\Desktop\ADP\Medical-Records> python -m pytest test/test_performance.py -v
test session starts
platform win32 -- Python 3.10.11, pytest-7.4.0, pluggy-1.6.0 -- C:\Users\tontide1\scoop\apps\miniconda3\current\envs\medical-records-env\python.exe
cachedir: .pytest_cache
rootdir: C:\Users\tontide1\Desktop\ADP
configfile: pytest.ini
plugins: cov-4.1.0
collected 10 items

test\test_performance.py::PerformanceTest::test_admin_operations_performance PASSED [ 10%]
test\test_performance.py::PerformanceTest::test_bulk_medical_record_creation PASSED [ 20%]
test\test_performance.py::PerformanceTest::test_concurrent_user_operations PASSED [ 30%]
test\test_performance.py::PerformanceTest::test_database_query_performance PASSED [ 40%]
test\test_performance.py::PerformanceTest::test_doctor_operations_performance PASSED [ 50%]
test\test_performance.py::PerformanceTest::test_export_performance PASSED [ 60%]
test\test_performance.py::PerformanceTest::test_medical_record_creation_performance PASSED [ 70%]
test\test_performance.py::PerformanceTest::test_patient_operations_performance PASSED [ 80%]
test\test_performance.py::PerformanceTest::test_search_performance PASSED [ 90%]
test\test_performance.py::PerformanceTest::test_user_registration_performance PASSED [100%]
```

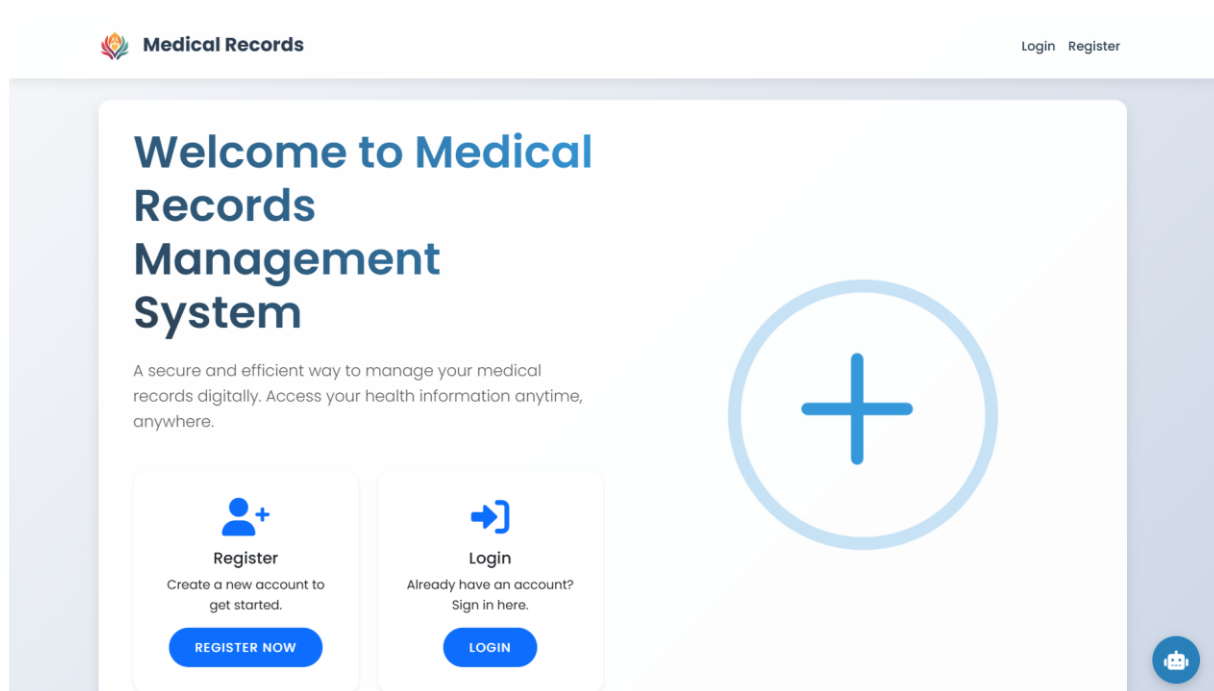
Hình 2.4 Kết quả chạy test Performance Testing

CHƯƠNG 3. KẾT QUẢ, HẠN CHẾ VÀ HƯỚNG PHÁT TRIỂN

1. Kết quả đạt được

Dự án đã mang lại một số kết quả khá tốt. Tuy chưa phải kết quả hoàn thiện nhưng đã đáp ứng phần lớn chức năng cơ bản. Dưới đây là một số kết quả nhóm đã đạt được trong dự án “Medical Records”:

- **Xây dựng thành công hệ thống quản lý hồ sơ y tế trực tuyến:** Dự án đã phát triển một ứng dụng web hoàn chỉnh cho phép quản lý thông tin xét nghiệm máu với ba vai trò người dùng chính: Bệnh nhân, Bác sĩ và Quản trị viên. Hệ thống đáp ứng được các mục tiêu cơ bản đặt ra ban đầu.



Hình 3.1. Giao diện chung trang web

Welcome Back!

Username
user1

Password

☐ Remember Me

Login

[Forgot Password?](#)

Don't have an account? [Register here](#)

Create Account

Username
user1

Email

Phone

Password

Confirm Password

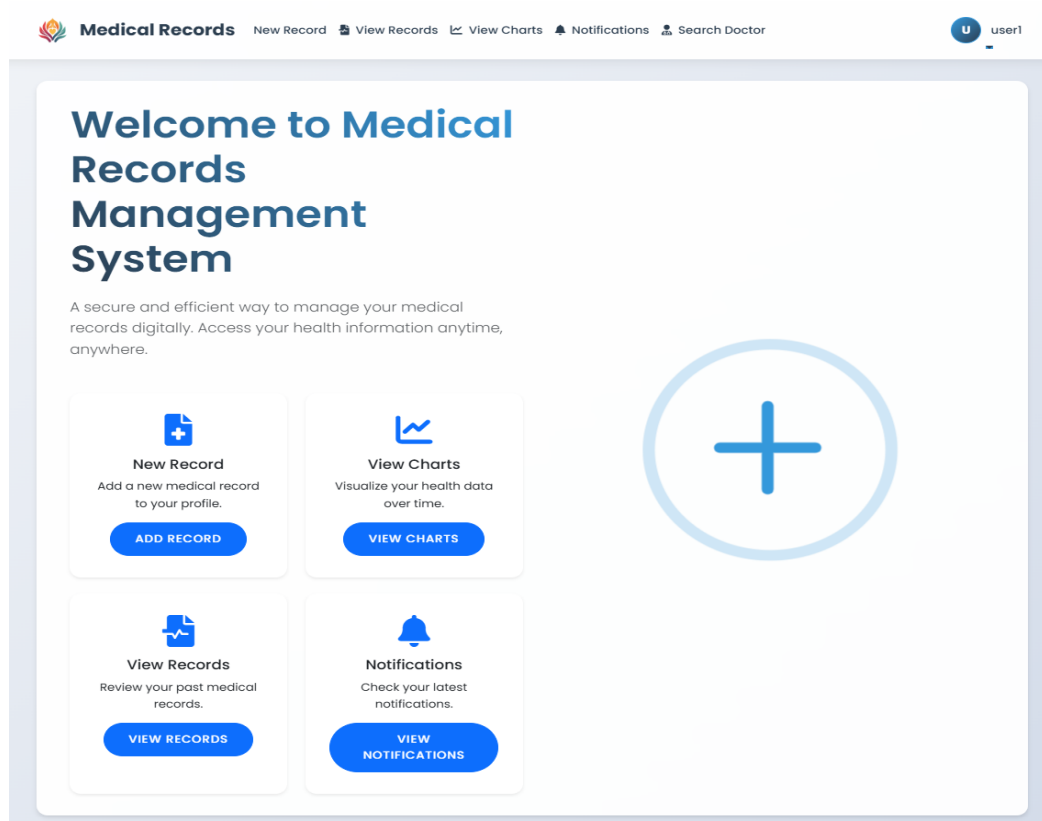
Create Account

Already have an account? [Login here](#)

Hình 3.2. Giao diện chức năng đăng nhập/ đăng kí

- Hoàn thiện các chức năng cốt lõi cho từng vai trò người dùng:

+ Bệnh nhân: Có thể đăng ký, đăng nhập, quản lý thông tin cá nhân, tự nhập và theo dõi lịch sử kết quả xét nghiệm máu (10 chỉ số HGB, RBC, WBC, PLT, HCT, Glucose, Creatinine, ALT, Cholesterol, CRP), xem biểu đồ trực quan hóa dữ liệu, nhận và quản lý thông báo từ bác sĩ, tìm kiếm thông tin bác sĩ, và tương tác với trợ lý AI.



Hình 3.3. Giao diện sau khi người dùng đăng nhập thành công

New Medical Record

Date

HGB

Glucose

RBC

Creatinine

WBC

ALT







PLT

Cholesterol

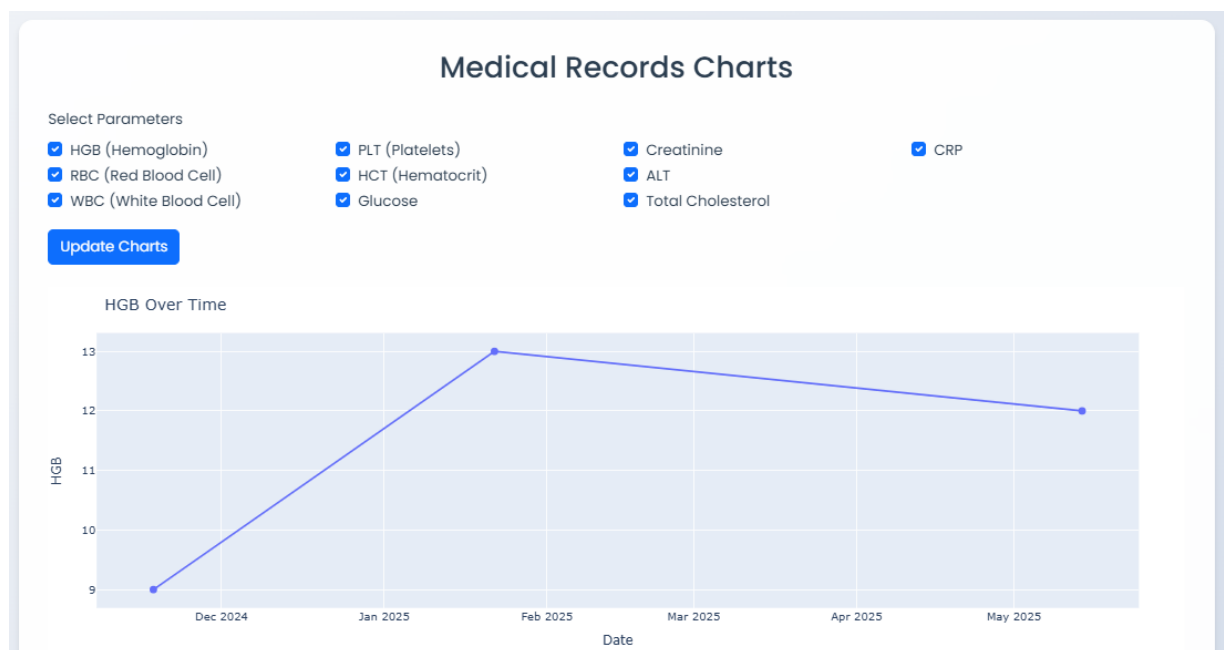
HCT

CRP

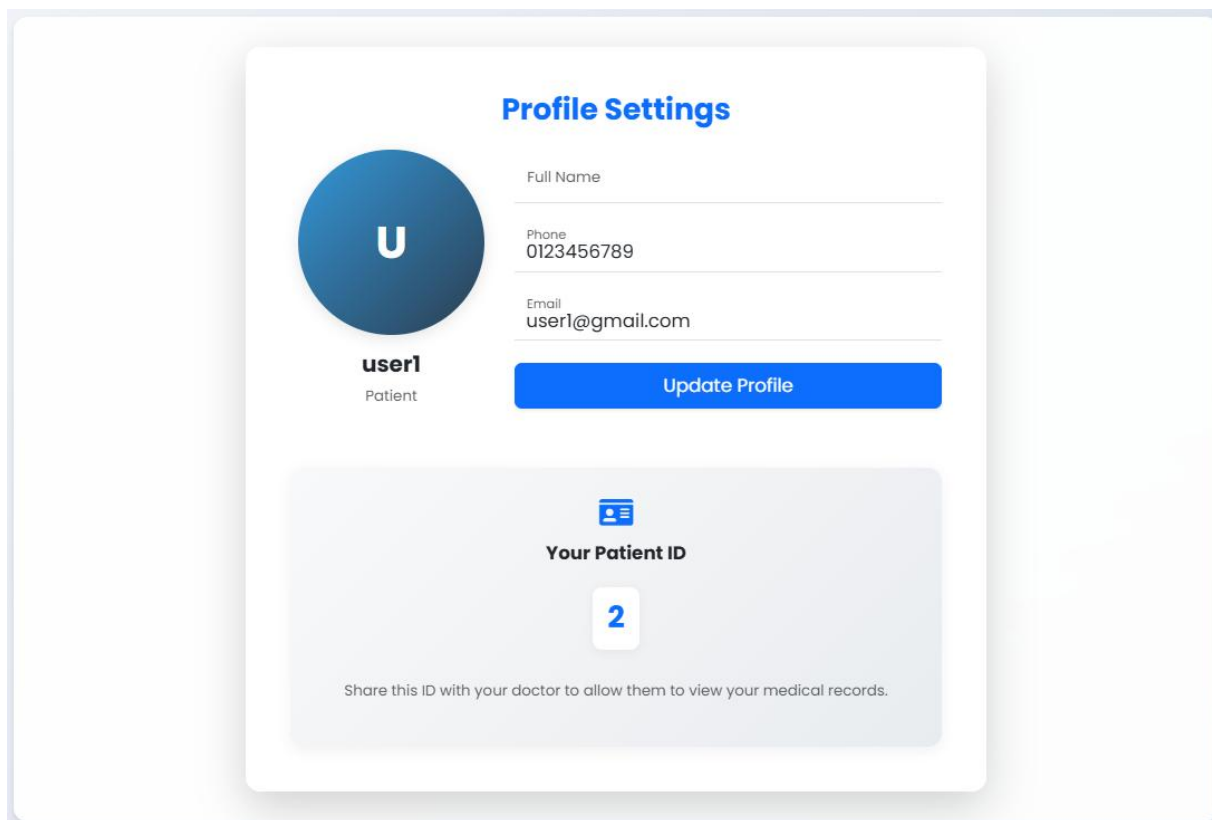
Hình 3.4. Giao diện chức năng thêm bản ghi

| Medical Records – Patient: None (user1) | | | | | | | | | | | |
|---|------------------------|---------------------------|----------------------------|---------------------------|---------------------|------------------------------|----------------------------------|----------------------|---------------------------------|------------------------|---|
| Date | HGB (g/dL) 12–16 | RBC (M/ μL) 4.2–5.4 | WBC (K/ μL) 4.5–11.0 | PLT (K/ μL) 150–450 | HCT (%) 36–46 | Glucose (mg/dL) 70–100 | Creatinine (mg/dL) 0.7–1.3 | ALT (U/L) 7–56 | Cholesterol (mg/dL) < 200 | CRP (mg/L) < 3.0 | Actions |
| 2025-05-14 | 12.0 | 7.0 | 8.0 | 11.0 | 6.0 | 5.0 | 2.0 | 3.0 | 9.0 | 1.0 |   |
| 2025-01-22 | 13.0 | 6.0 | 7.0 | 10.0 | 8.0 | 9.0 | 2.0 | 8.0 | 7.0 | 4.0 |   |
| 2024-11-18 | 9.0 | 8.0 | 6.0 | 7.0 | 10.0 | 7.0 | 4.0 | 4.0 | 12.0 | 2.0 |   |

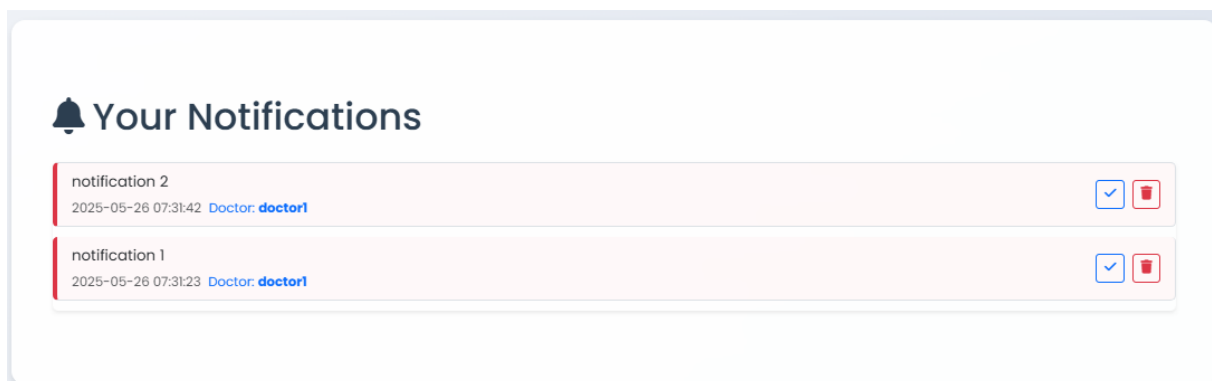
Hình 3.5. Giao diện chức năng xem lại bản ghi của người dùng



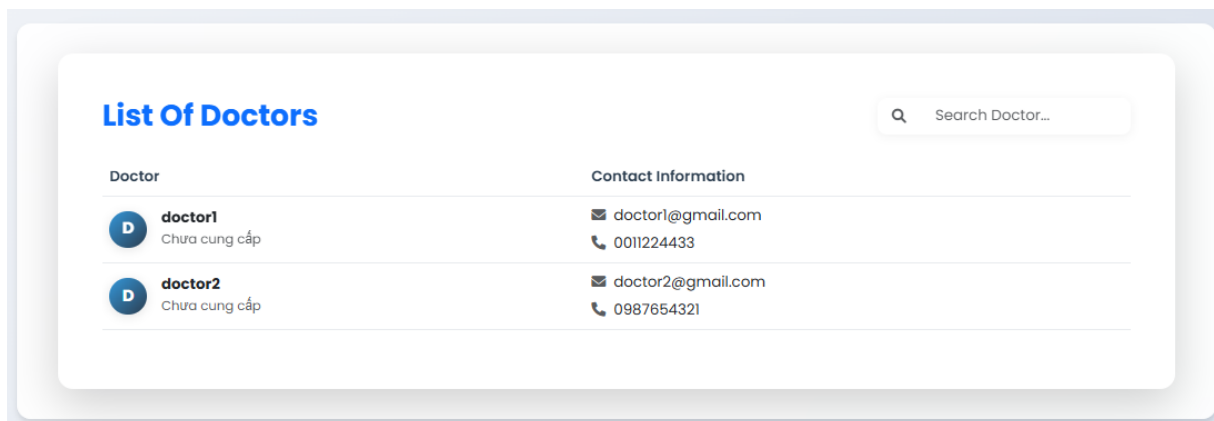
Hình 3.6. Giao diện chức năng xem biểu đồ chỉ số máu



Hình 3.7. Giao diện hồ sơ của người dùng

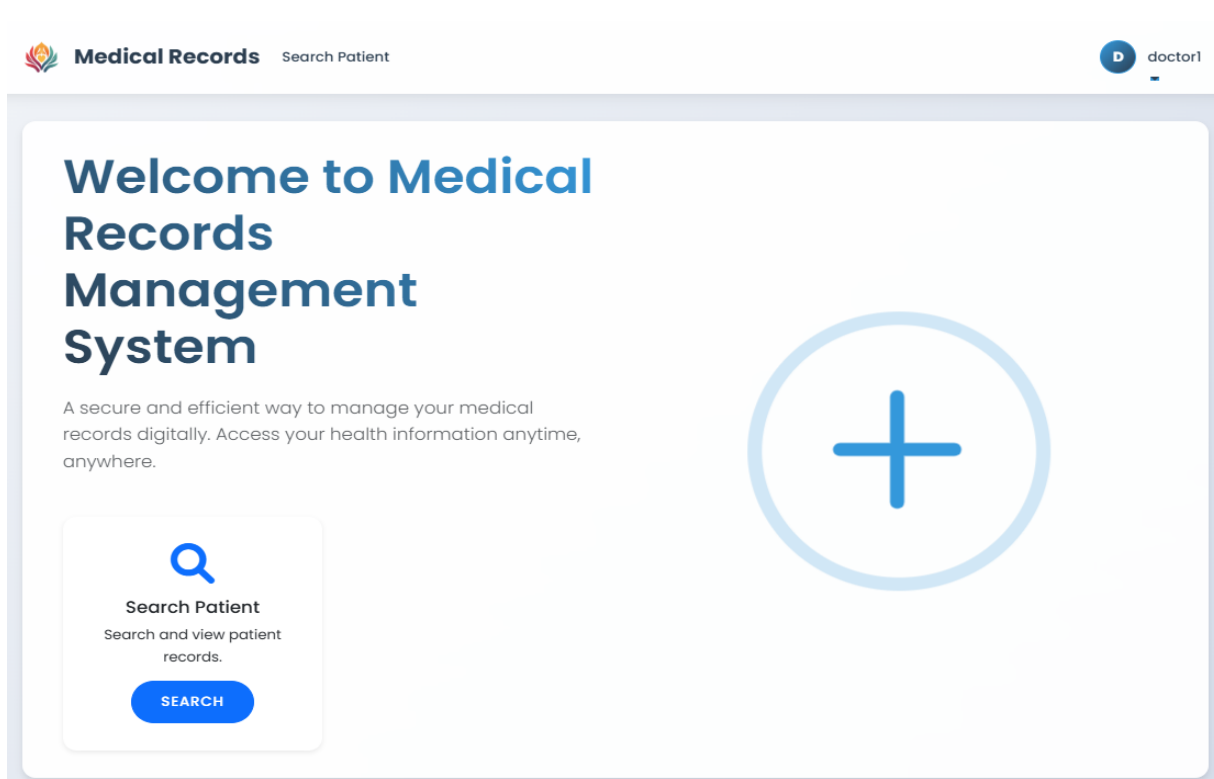


Hình 3.8. Giao diện chức năng xem thông báo của người dùng

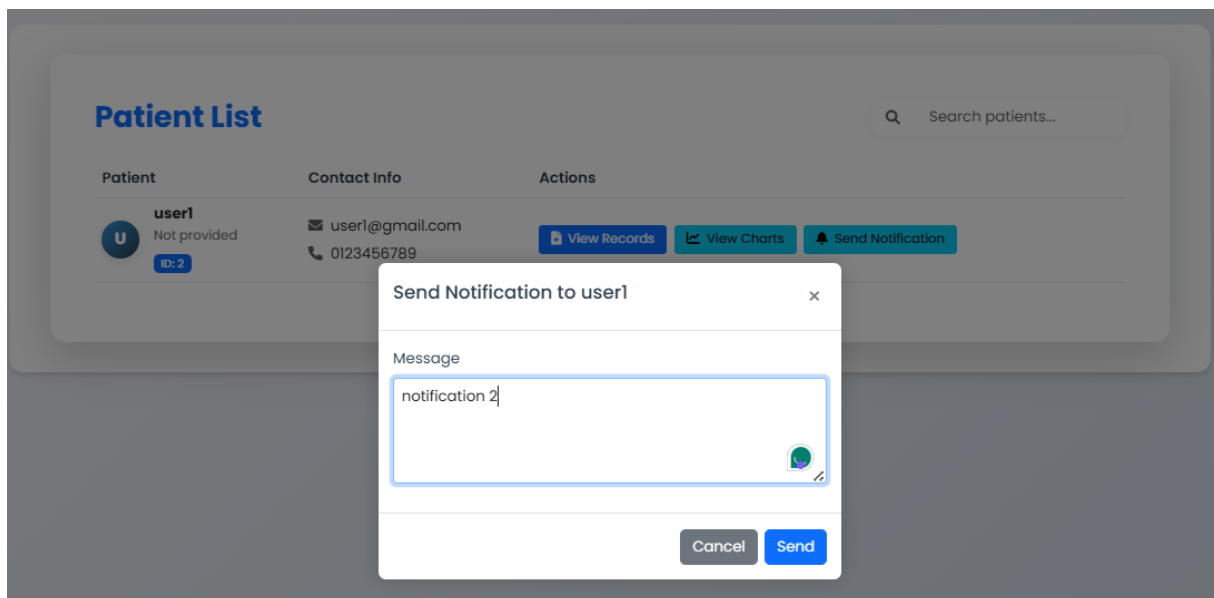


Hình 3.9. Giao diện tìm kiếm thông tin bác sĩ của bệnh nhân

+ Bác sĩ: Có thể tìm kiếm bệnh nhân, xem chi tiết hồ sơ bệnh án (dạng bảng và biểu đồ), gửi thông báo/khuyến nghị cho bệnh nhân, và tải xuống dữ liệu bệnh nhân định dạng CSV.

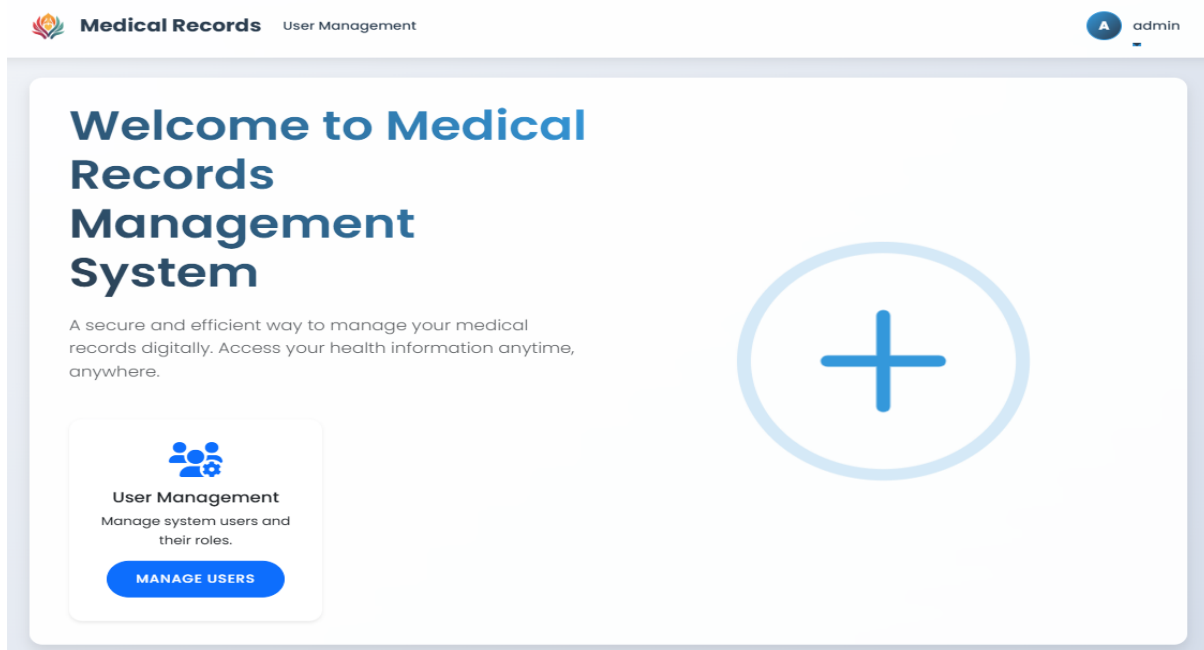


Hình 3.10. Giao diện sau khi bác sĩ đăng nhập thành công

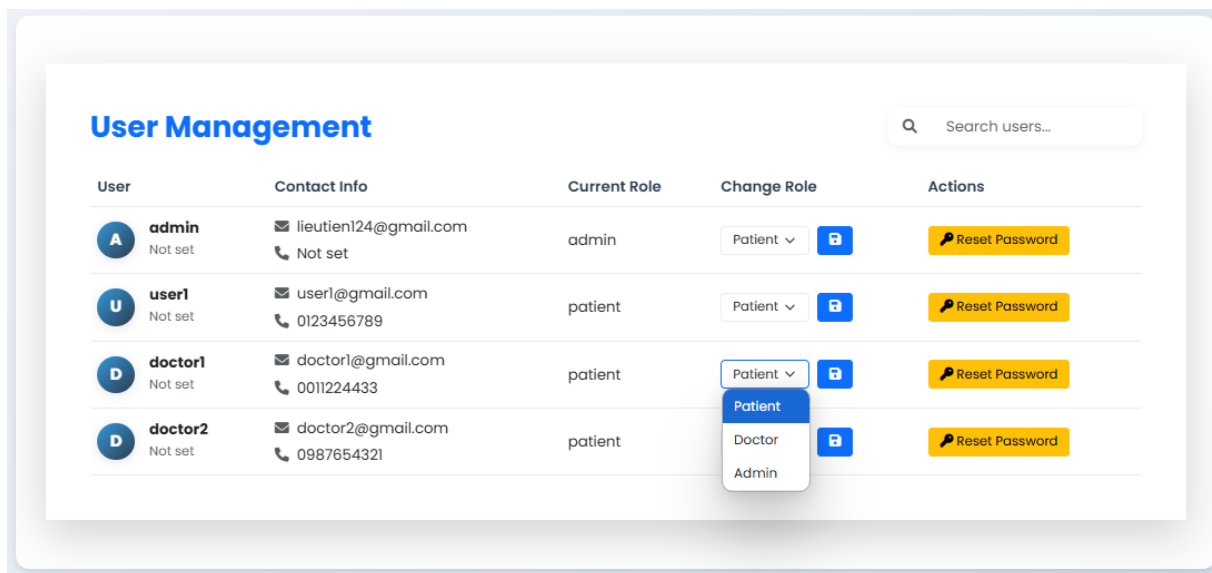


Hình 3.11. Giao diện gửi thông báo cho bệnh nhân của bác sĩ

+ Quản trị viên: Có thể quản lý tài khoản người dùng (xem danh sách, cập nhật vai trò, đặt lại mật khẩu), và xuất dữ liệu người dùng, hồ sơ bệnh án ra file Excel.



Hình 3.12. Giao diện sau khi quản trị viên đăng nhập thành công



Hình 3.13. Giao diện chức năng quản lý người dùng của quản trị viên

- **Giao diện người dùng thân thiện và trực quan:** Hệ thống cung cấp giao diện web dễ sử dụng, được xây dựng dựa trên các template HTML, CSS (có khả năng là Bootstrap) và Jinja2, giúp người dùng dễ dàng tương tác và thực hiện các thao tác.

- **Trực quan hóa dữ liệu sức khỏe hiệu quả:** Chức năng hiển thị biểu đồ các chỉ số xét nghiệm theo thời gian sử dụng Plotly (Python) và Plotly.js đã được triển khai thành công, giúp bệnh nhân và bác sĩ dễ dàng theo dõi và phân tích xu hướng sức khỏe.

- **Tích hợp thành công trợ lý AI (Gemini):** Hệ thống đã kết nối và sử dụng Google Gemini API để cung cấp chức năng chatbot tư vấn y tế cho bệnh nhân, có khả năng tham khảo lịch sử bệnh án để tăng tính cá nhân hóa. Lịch sử chat cũng được lưu lại.

- **Đảm bảo các yêu cầu về bảo mật cơ bản:** Đã triển khai các biện pháp bảo mật như băm mật khẩu người dùng (sử dụng werkzeug.security), bảo vệ chống tấn công CSRF (sử dụng Flask-WTF), và phân quyền truy cập dựa trên vai trò người dùng.

- **Hệ thống hoạt động ổn định và đáng tin cậy:**

+ Tất cả 21 ca kiểm thử chức năng đã thực hiện đều PASSED (như hình ảnh kết quả

kiểm thử functional test bạn cung cấp), cho thấy các chức năng chính hoạt động đúng theo thiết kế.

+ Tất cả 10 ca kiểm thử hiệu năng đã thực hiện đều PASSED (như hình ảnh kết quả kiểm thử performance test bạn cung cấp), cho thấy hệ thống đáp ứng tốt về tốc độ và khả năng chịu tải trong các kịch bản được kiểm thử.

- **Hoàn thành việc xây dựng cơ sở dữ liệu:** Đã thiết kế và triển khai thành công cơ sở dữ liệu với các bảng User, MedicalRecord, Notification, ChatHistory và các mối quan hệ giữa chúng, đáp ứng nhu cầu lưu trữ và truy xuất thông tin của hệ thống. (Như sơ đồ cơ sở dữ liệu bạn cung cấp).

2. Hạn chế và hướng phát triển

Tuy đã đạt được những kết quả khá tốt, song dự án vẫn còn một số hạn chế. Dưới đây là một số hạn chế của dự án cùng với đó là hướng phát triển nhằm khắc phục những hạn chế đó:

- **Khả năng mở rộng với SQLite:** Hiện tại, hệ thống sử dụng SQLite làm cơ sở dữ liệu. Mặc dù phù hợp cho giai đoạn phát triển và các ứng dụng quy mô nhỏ, SQLite có thể gặp hạn chế về hiệu năng và khả năng xử lý đồng thời khi số lượng người dùng và dữ liệu tăng lớn. **Hướng phát triển:** Có thể xem xét chuyển đổi sang các hệ quản trị cơ sở dữ liệu mạnh mẽ hơn như PostgreSQL hoặc MySQL cho các phiên bản triển khai quy mô lớn hơn.

- **Thiếu tính năng tương tác trực tiếp (real-time) giữa Bác sĩ và Bệnh nhân:** Mặc dù có hệ thống thông báo, hệ thống hiện tại chưa hỗ trợ chat trực tiếp theo thời gian thực giữa bác sĩ và bệnh nhân, điều này có thể cải thiện trải nghiệm tư vấn. **Hướng phát triển:** Tích hợp công nghệ WebSocket hoặc các giải pháp chat real-time.

- **Phạm vi tư vấn của AI còn giới hạn:** Chức năng AI hiện tại chủ yếu dựa trên Q&A và thông tin bệnh án cơ bản. Chưa có các tính năng AI nâng cao như phân tích dự đoán,

gợi ý chẩn đoán sâu hoặc cảnh báo sớm dựa trên dữ liệu. **Hướng phát triển:** Nghiên cứu và tích hợp các mô hình AI phức tạp hơn, huấn luyện trên tập dữ liệu lớn hơn (nếu có) để tăng cường khả năng của trợ lý ảo.

- **Chưa có ứng dụng di động riêng biệt:** Hệ thống hiện tại là một ứng dụng web. Mặc dù có thể được thiết kế responsive, việc phát triển ứng dụng di động (native app) chuyên biệt cho Android và iOS sẽ mang lại trải nghiệm tốt hơn và tiện lợi hơn cho người dùng trên thiết bị di động.

- **Tuân thủ các tiêu chuẩn y tế và bảo mật dữ liệu quốc tế:** Đối với một ứng dụng y tế thực tế, việc tuân thủ các tiêu chuẩn nghiêm ngặt về bảo mật và quyền riêng tư dữ liệu y tế như HIPAA (Mỹ) hay GDPR (Châu Âu) là bắt buộc. Dự án hiện tại có các biện pháp bảo mật cơ bản nhưng để đạt được sự tuân thủ đầy đủ cần đầu tư nghiên cứu và triển khai thêm.

- **Thiếu tài liệu hướng dẫn người dùng chi tiết:** Hiện tại, có thể chưa có tài liệu hướng dẫn sử dụng chi tiết cho từng nhóm người dùng, điều này quan trọng để họ có thể khai thác tối đa các tính năng của hệ thống. **Hướng phát triển:** Xây dựng bộ tài liệu hướng dẫn sử dụng (User Manual) và FAQ.