

HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG
KHOA CÔNG NGHỆ THÔNG TIN



ĐỀ CƯƠNG THỰC TẬP CƠ SỞ

**ĐỀ TÀI: PHÁT TRIỂN WEB ĐẶT LỊCH HẸN TRỰC TUYẾN
CHO PHÒNG KHÁM HOẶC BỆNH VIỆN**

Tên sinh viên	Trần Nhật Minh
Mã sinh viên	B22DCVT350
Lớp	E22CQCN02-B

Giáo viên phụ trách: Kim Ngọc Bách

Hà Nội 2025

I. TỔNG QUAN VỀ ĐỀ TÀI

1. Bối cảnh và lý do thực hiện

-Trong bối cảnh chuyển đổi số ngày càng được chú trọng, lĩnh vực y tế cũng cần ứng dụng công nghệ để tối ưu quy trình khám chữa bệnh.

-Thực trạng thường gặp ở các phòng khám, bệnh viện là tình trạng quá tải, bệnh nhân phải xếp hàng lâu và gây lãng phí thời gian lẫn nguồn lực.

-Việc đặt lịch hẹn trực tuyến sẽ giúp bệnh nhân chủ động chọn thời gian rảnh, hạn chế chờ đợi quá lâu; đồng thời phòng khám/bệnh viện cũng có thể sắp xếp lịch cho bác sĩ, nâng cao hiệu quả quản lý.

2. Mục tiêu của đề tài

-Xây dựng một nền tảng web giúp bệnh nhân đặt lịch khám với bác sĩ thông qua một vài thao tác đơn giản.

-Giúp bác sĩ và nhân viên y tế dễ dàng quản lý lịch hẹn, quản lý trạng thái khám, đồng thời giảm sai sót so với quy trình thủ công.

-Đảm bảo bảo mật dữ liệu và trải nghiệm người dùng (UI/UX) tốt, có tiềm năng mở rộng sang các dịch vụ khác (hồ sơ bệnh án điện tử, thanh toán trực tuyến...).

3. Phạm vi và đối tượng áp dụng

-Phạm vi: Đề tài hướng đến một phòng khám hoặc bệnh viện quy mô vừa, với số lượng bác sĩ và bệnh nhân tương đối lớn, cần một giải pháp đơn giản, nhưng hiệu quả và có thể nâng cấp.

-Đối tượng:

+Bệnh nhân: Tìm kiếm, chọn bác sĩ theo chuyên khoa, đặt lịch và nhận thông báo.

+Bác sĩ: Quản lý lịch hẹn, cập nhật trạng thái, xem thông tin bệnh nhân.

+Quản trị viên (Admin): Quản lý toàn bộ hệ thống, phân quyền người dùng, xem báo cáo hoạt động.

4. Lợi ích dự kiến

-Tiết kiệm thời gian cho cả bệnh nhân và nhân viên y tế.

-Tạo uy tín cho phòng khám/bệnh viện nhờ quy trình chuyên nghiệp, ít sai sót.

-Thu thập dữ liệu phục vụ mục đích thống kê, hỗ trợ quyết định quản lý (ví dụ: xem giờ cao điểm, tỷ lệ bệnh nhân hủy hẹn...).

II. CÁC CHỨC NĂNG CỦA HỆ THỐNG

1. Quản lý tài khoản và phân quyền

-Đăng nhập, đăng ký: Bệnh nhân, bác sĩ và quản trị viên có quy trình đăng ký, đăng nhập khác nhau (có thể yêu cầu xác thực qua email/SMS).

Phân quyền:

- Bệnh nhân chỉ được đặt lịch, xem lịch của chính mình.
- Bác sĩ chỉ xem lịch của mình, cập nhật trạng thái khám.
- Admin có quyền cao nhất, quản lý tất cả thông tin.

2. Đặt lịch hẹn trực tuyến

Danh sách bác sĩ: Người dùng (bệnh nhân) có thể lọc theo chuyên khoa, xem thời gian làm việc của bác sĩ.

Chọn ngày/giờ trống: Hệ thống hiển thị khung giờ khả dụng (available) để bệnh nhân đặt lịch.

Xác nhận lịch hẹn: Bệnh nhân nhập thông tin cơ bản (lý do khám, ghi chú). Sau khi đặt, hệ thống ghi nhận, gửi thông báo xác nhận.

3. Quản lý lịch hẹn (Bác sĩ, Quản trị viên)

Bác sĩ: Xem chi tiết những bệnh nhân nào đã đặt trong ngày, cập nhật trạng thái (chờ khám, đã khám, hủy).

Admin: Có thể tạo, chỉnh sửa, hoặc hủy lịch hẹn trong tình huống đặc biệt (bác sĩ bận đột xuất).

Theo dõi các lịch hẹn trong quá khứ và kế hoạch sắp tới.

4. Gửi thông báo (Email/SMS)

-Tích hợp gửi email để nhắc nhở bệnh nhân trước giờ khám (có thể 1–2 ngày, hoặc vài tiếng).

-Tùy nhu cầu, có thể thêm SMS API để gửi tin nhắn (tốn chi phí), hoặc thông báo qua ứng dụng di động nếu hệ thống hỗ trợ.

5. Chức năng quản trị hệ thống

-Quản lý danh mục bác sĩ, chuyên khoa, phòng ban.

-Quản lý người dùng (tạo mới, khóa tạm thời...).

-Thống kê, báo cáo: số lượng lịch hẹn theo tháng, tỷ lệ hủy hẹn, tỷ lệ khám đúng giờ...

III. CÔNG NGHỆ SỬ DỤNG

1. Nền tảng Back-end

Node.js:

-Thích hợp cho xây dựng ứng dụng web **thời gian thực** và **API** nhẹ.

-Có hệ sinh thái **npm** phong phú, cộng đồng lớn, dễ tìm tài liệu.

Express.js:

-Framework phổ biến trên nền Node.js, hỗ trợ RESTful nhanh.

-Dễ cấu hình, có nhiều middleware tích hợp sẵn.

2. Cơ sở dữ liệu

-Dùng MySQL hoặc PostgreSQL (quan hệ), tùy theo kinh nghiệm hoặc cơ sở hạ tầng hiện có.

-Tạo các bảng chính như Users, Doctors, Appointments, Specialties...

-(Tùy chọn) Sử dụng Sequelize (ORM) để thao tác CSDL dễ dàng, có tính năng migration và association.

3. Front-end

HTML5, CSS3, JavaScript: Nền tảng căn bản để xây dựng giao diện.

Bootstrap/Tailwind hoặc **React/Vue** (nếu muốn tách biệt front-end/back-end) để tăng tốc độ phát triển, đảm bảo **responsive** cho di động.

Sử dụng **Axios** hoặc **Fetch API** để gọi các **API** từ server Node.js.

4. Thông báo Email/SMS

Tích hợp các dịch vụ như **Nodemailer** (SMTP) để gửi email.

(Tùy chọn) **Twilio**, **Nexmo** hoặc cổng SMS của nhà mạng để gửi tin nhắn SMS nhắc hẹn.

5. Công cụ và môi trường triển khai

GitHub để quản lý mã nguồn, theo dõi lịch sử thay đổi.

IV. KẾ HOẠCH THỰC HIỆN DỰ ÁN (bắt đầu từ 1/4)

1. Giai đoạn 1: Phân tích và Thiết kế (Tuần 1 – 2)

Phân tích hệ thống: Vẽ sơ đồ Use Case, ERD, Activity Diagram, xác định quy trình nghiệp vụ.

Thiết kế giao diện: Xây dựng bằng Figma cho trang chính, trang đặt lịch, trang quản trị.

2. Giai đoạn 2: Triển khai Back-end và CSDL (Tuần 3-4)

Cấu trúc dự án **Node.js + Express**, thiết lập **Sequelize** (hoặc ORM tương tự) để kết nối CSDL.

Xây dựng các API chính: đăng nhập, đăng ký, đặt lịch, quản lý lịch, cập nhật bác sĩ...

Áp dụng **bảo mật** (mã hóa mật khẩu, phân quyền).

3. Giai đoạn 3: Xây dựng Front-end & Tích hợp (Tuần 5-6)

Xây dựng giao diện (HTML/CSS/JS) tương ứng với mockup đã thiết kế.

Gọi **API** từ server Back-end, hiển thị dữ liệu (danh sách bác sĩ, lịch hẹn...).

Tích hợp **chức năng đặt lịch** hoàn chỉnh, kiểm tra luồng xem – đặt – hủy.

4. Giai đoạn 4: Kiểm thử, Tối ưu và Hoàn thiện (Tuần 7-8)

Test chức năng frontend + backend: hoàn thiện các chức năng Đăng nhập, đăng ký, Đặt lịch hẹn trực tuyến, Quản lý lịch hẹn, Gửi thông báo,

Kiểm thử hiệu năng (nếu có điều kiện): đánh giá tốc độ phản hồi khi nhiều người dùng.

Soạn thảo **tài liệu hướng dẫn** (cài đặt, sử dụng), viết **báo cáo** tổng kết.