Dưới đây là 75 câu hỏi và câu trả lời chi tiết về luận văn của bạn:

**1-10: Tổng quan về đề tài**

1. **Tại sao bạn chọn đề tài này?**
   * Đề tài xuất phát từ nhu cầu thực tế về chuyển đổi số trong y tế, đặc biệt là các phòng khám tư nhân, nơi quản lý truyền thống còn gặp nhiều khó khăn như lưu trữ giấy tờ và xử lý thông tin thủ công.
2. **Đề tài giải quyết vấn đề gì?**
   * Đề tài giúp số hóa quy trình quản lý phòng khám, bao gồm đặt lịch khám, quản lý bệnh nhân, bác sĩ, và hồ sơ y tế.
3. **Lý do bạn chọn công nghệ ASP.NET Core?**
   * ASP.NET Core có khả năng xử lý hiệu quả các API, hỗ trợ đa nền tảng và đảm bảo bảo mật cao.
4. **Những công nghệ nào bạn thấy có thể thay thế công nghệ hiện tại?**
   * Node.js hoặc Django có thể thay thế ASP.NET Core trong backend, và Angular hoặc Vue.js có thể thay ReactJS ở frontend.
5. **Lợi ích của hệ thống với người dùng là gì?**
   * Giảm thời gian chờ, tăng tiện ích khi đặt lịch khám online, và cung cấp dữ liệu khám bệnh chính xác.
6. **Những yếu tố quan trọng nhất để phát triển hệ thống này là gì?**
   * Hiểu rõ nghiệp vụ y tế, sử dụng công nghệ phù hợp và thiết kế giao diện thân thiện.
7. **Hệ thống hiện tại giải quyết được bao nhiêu phần trăm yêu cầu thực tế?**
   * Hệ thống hiện tại đáp ứng khoảng 80%, với một số tính năng thực tế cần được phát triển thêm.
8. **Nếu được làm lại, bạn sẽ thay đổi gì trong đề tài?**
   * Tôi sẽ tập trung hơn vào tối ưu hiệu suất và tích hợp thêm trí tuệ nhân tạo để hỗ trợ quyết định.
9. **Bạn thấy hệ thống này phù hợp với đối tượng nào?**
   * Các phòng khám tư nhân có quy mô vừa và nhỏ, muốn nâng cao hiệu quả quản lý.
10. **Bạn đã tham khảo những tài liệu nào trong quá trình làm?**
    * Các tài liệu về ReactJS, ASP.NET Core, quản lý cơ sở dữ liệu SQL Server, và các trang web phòng khám thực tế

**11-20: Về thiết kế hệ thống**

1. **Kiến trúc của hệ thống được thiết kế như thế nào?**
   * Hệ thống theo mô hình client-server: frontend sử dụng ReactJS và backend là ASP.NET Core, giao tiếp qua API.
2. **Bạn đã tối ưu dữ liệu trong SQL Server như thế nào?**
   * Sử dụng các chỉ số (indexes) và tối ưu hóa truy vấn bằng cách phân tích kế hoạch thực thi (execution plan).
3. **Tại sao bạn chọn ReactJS thay vì Angular hoặc Vue?**
   * ReactJS linh hoạt, hiệu suất cao, và phù hợp với việc xây dựng giao diện động.
4. **Bạn đã thiết kế giao diện người dùng dựa trên nguyên tắc nào?**
   * Dựa trên nguyên tắc thân thiện, dễ sử dụng, và tập trung vào trải nghiệm người dùng (UX/UI).
5. **Hệ thống có sử dụng các mẫu thiết kế phần mềm không?**
   * Có, áp dụng mẫu MVC và Repository Pattern.
6. **Bạn xử lý phân quyền người dùng như thế nào?**
   * Phân quyền dựa trên vai trò (Role-Based Access Control) được lưu trữ trong bảng Role của SQL Server.
7. **Tại sao bạn chọn Bootstrap để xây dựng giao diện?**
   * Bootstrap giúp phát triển giao diện nhanh hơn với các thành phần sẵn có, đồng thời hỗ trợ responsive.
8. **Hệ thống hỗ trợ bao nhiêu vai trò người dùng?**
   * Hỗ trợ bốn vai trò: khách vãng lai, khách hàng, bác sĩ, và nhân viên.
9. **Hệ thống có hỗ trợ tính năng đa ngôn ngữ không?**
   * Hiện tại chưa, nhưng có thể thêm thư viện i18next để hỗ trợ.
10. **Tại sao bạn chọn sử dụng JWT (JSON Web Token) thay vì session?**
    * JWT nhẹ, dễ dàng tích hợp với các ứng dụng frontend, và hỗ trợ tốt trong hệ thống phân tán

**21-35: Về chức năng hệ thống**

1. **Hệ thống cho phép khách hàng thanh toán như thế nào?**
   * Khách hàng có thể thanh toán qua cổng VnPay, với xác thực qua mã OTP.
2. **Chức năng thống kê hoạt động thế nào?**
   * Hệ thống cho phép nhân viên chọn tháng, năm để tạo báo cáo, xuất ra file Excel.
3. **Cách hệ thống xử lý khi có lỗi đặt lịch khám?**
   * Hiển thị thông báo lỗi chi tiết để người dùng điều chỉnh.
4. **Chức năng đăng nhập qua Google được triển khai như thế nào?**
   * Sử dụng Google OAuth2 để xác thực người dùng.
5. **Bạn đã kiểm tra tính chính xác của dữ liệu nhập vào như thế nào?**
   * Áp dụng kiểm tra trên frontend và backend, với các validate rules.
6. **Làm sao để thêm bác sĩ hoặc dịch vụ mới vào hệ thống?**
   * Nhân viên có thể thêm thông tin qua bảng điều khiển.
7. **Hệ thống có thể xóa dữ liệu nhạy cảm không?**
   * Có, sử dụng cơ chế xóa mềm (soft delete) để bảo vệ dữ liệu quan trọng.
8. **Bạn đã triển khai chức năng tìm kiếm như thế nào?**
   * Sử dụng truy vấn SQL với từ khóa nhập từ frontend.
9. **Làm sao để hệ thống hỗ trợ đặt lịch khám trùng giờ?**
   * Áp dụng kiểm tra logic trong backend để từ chối các yêu cầu trùng lặp.
10. **Bác sĩ có thể xem thông tin bệnh nhân như thế nào?**
    * Bác sĩ truy cập bảng điều khiển, nơi hiển thị danh sách bệnh nhân và lịch sử khám bệnh.
11. **Hệ thống có hỗ trợ gửi thông báo qua email không?**
    * Hiện tại chưa, nhưng có thể tích hợp SMTP để gửi thông báo tự động.
12. **Chức năng lưu trữ hồ sơ bệnh nhân hoạt động ra sao?**
    * Lưu trữ trong bảng Medical Record và liên kết với các bảng khác như Checkup.
13. **Hệ thống có hỗ trợ quản lý thuốc không?**
    * Có, quản lý số lượng thuốc tồn kho và ngày hết hạn.
14. **Bạn xử lý giao diện khi hệ thống tải chậm như thế nào?**
    * Sử dụng thanh tải (loading bar) để cải thiện trải nghiệm người dùng.
15. **Bạn làm sao để người dùng đặt lịch mà không cần tài khoản?**
    * Chức năng này chưa hỗ trợ, cần xây dựng thêm.

**36-50: Về kiểm thử**

1. **Bạn đã kiểm thử hệ thống như thế nào?**
   * Thực hiện kiểm thử đơn vị (unit testing) cho từng thành phần, kiểm thử tích hợp (integration testing) giữa frontend và backend, và kiểm thử chức năng (functional testing) để đảm bảo hệ thống hoạt động như mong đợi.
2. **Làm sao bạn đảm bảo rằng hệ thống hoạt động chính xác?**
   * So sánh kết quả kiểm thử với các yêu cầu đã đặt ra và sử dụng các kịch bản kiểm thử để đánh giá tất cả các chức năng.
3. **Công cụ nào bạn sử dụng trong quá trình kiểm thử?**
   * Sử dụng Postman để kiểm thử API, Selenium để kiểm thử giao diện, và SQL Profiler để kiểm tra hiệu suất cơ sở dữ liệu.
4. **Bạn đã thiết kế các kịch bản kiểm thử như thế nào?**
   * Các kịch bản kiểm thử được thiết kế dựa trên các trường hợp sử dụng (use cases) đã mô tả trong luận văn, bao gồm cả các trường hợp thông thường và ngoại lệ.
5. **Có bao nhiêu trường hợp kiểm thử bạn đã thực hiện?**
   * Khoảng 50 trường hợp, bao gồm kiểm thử đăng nhập, đặt lịch khám, quản lý dữ liệu, tìm kiếm và thống kê.
6. **Bạn đã gặp những lỗi nào khi kiểm thử?**
   * Một số lỗi liên quan đến xử lý lỗi người dùng nhập sai, truy vấn SQL không tối ưu, và hiển thị giao diện trên các thiết bị khác nhau.
7. **Làm sao bạn đảm bảo tính chính xác của các API?**
   * Sử dụng Postman để gửi yêu cầu đến API và kiểm tra phản hồi đúng không
8. **Bạn đã kiểm tra hiệu suất hệ thống như thế nào?**
   * Tạo các yêu cầu tải lớn (stress testing) và kiểm tra thời gian phản hồi của hệ thống với số lượng lớn người dùng đồng thời.
9. **Kiểm thử chức năng đặt lịch khám như thế nào?**
   * Kiểm tra các tình huống đặt lịch thành công, nhập thông tin thiếu hoặc sai, và đặt lịch trùng thời gian.
10. **Bạn đã thực hiện kiểm thử giao diện trên những thiết bị nào?**
    * Thực hiện kiểm thử trên máy tính (desktop), máy tính bảng, và điện thoại di động để đảm bảo giao diện responsive.
11. **Hệ thống có bị ảnh hưởng bởi các lỗi hiển thị trên trình duyệt khác nhau không?**
    * Có, nhưng đã khắc phục bằng cách kiểm tra giao diện trên các trình duyệt phổ biến như Chrome, Firefox, và Edge.
12. **Bạn đã thử nghiệm với dữ liệu lớn chưa?**
    * Có, đã nhập dữ liệu của 10.000 bản ghi vào cơ sở dữ liệu và kiểm tra thời gian truy xuất, hiệu suất không bị ảnh hưởng đáng kể.
13. **Làm sao bạn biết rằng giao diện người dùng hoạt động chính xác?**
    * Kiểm tra từng thao tác trên giao diện và so sánh kết quả với yêu cầu đề ra, đồng thời sử dụng Selenium để tự động hóa kiểm thử.
14. **Bạn có kiểm thử tính năng xuất file thống kê không?**
    * Có, đã kiểm tra việc tạo và xuất file Excel với dữ liệu thực tế, đảm bảo file không bị lỗi và đúng định dạng.
15. **Kết quả kiểm thử của bạn có đạt yêu cầu không?**
    * Hầu hết các trường hợp kiểm thử đều đạt yêu cầu, chỉ một số ít vấn đề về giao diện cần tinh chỉnh thêm để cải thiện trải nghiệm người dùng

**51-75: Về mở rộng và ứng dụng**

1. **Làm sao để hệ thống áp dụng được cho bệnh viện lớn?**
   * Nâng cấp hệ thống để quản lý nhiều chi nhánh và mở rộng các chức năng như quản lý tài chính.
2. **Bạn thấy tiềm năng phát triển AI trong hệ thống này như thế nào?**
   * AI có thể được tích hợp để dự đoán bệnh hoặc tư vấn sức khỏe tự động.
3. **Cách tích hợp thêm các cổng thanh toán khác?**
   * Xây dựng API linh hoạt và tích hợp thêm SDK của các cổng thanh toán.
4. **Hệ thống có thể dùng để quản lý nhà thuốc không?**
   * Có, nhưng cần bổ sung thêm các chức năng quản lý hóa đơn bán lẻ.
5. **Bạn sẽ làm gì để cải thiện giao diện hệ thống?**
   * Sử dụng thêm các thư viện UI như Material-UI hoặc Ant Design.
6. **Hệ thống có thể xuất file PDF không?**
   * Có thể, sử dụng thư viện như iTextSharp trong .NET Core.
7. **Bạn có dự định mở rộng hệ thống cho các ngành khác không?**
   * Có, hệ thống có thể điều chỉnh để quản lý phòng gym hoặc nhà hàng.
8. **Hệ thống hỗ trợ tích hợp API bên thứ ba không?**
   * Có, như tích hợp API Google Maps để hiển thị vị trí phòng khám.
9. **Bạn sẽ cải thiện hiệu suất hệ thống như thế nào?**
   * Áp dụng cache cho dữ liệu ít thay đổi và tối ưu hóa các truy vấn SQL.
10. **Bạn có kế hoạch triển khai hệ thống trên đám mây không?**
    * Có, sử dụng dịch vụ như Azure hoặc AWS.
11. **Hệ thống có hỗ trợ báo cáo tự động không?**
    * Hiện tại chưa, nhưng có thể xây dựng tính năng này trong tương lai.
12. **Làm sao để hệ thống phù hợp với quy định y tế?**
    * Áp dụng tiêu chuẩn bảo mật y tế như HIPAA nếu triển khai quốc tế.
13. **Bạn nghĩ hệ thống này có tiềm năng thương mại hóa không?**
    * Có, vì quản lý số hóa phòng khám đang là nhu cầu lớn.
14. **Bạn làm sao để tích hợp chatbot vào hệ thống?**
    * Sử dụng thư viện như Dialogflow để tích hợp chatbot vào frontend.
15. **Làm sao để xử lý lỗi xảy ra trên server?**
    * Sử dụng logging tools như Serilog để theo dõi lỗi.
16. **Bạn có thể cải thiện tốc độ phản hồi của API không?**
    * Tối ưu truy vấn SQL và sử dụng Redis Cache.
17. **Hệ thống có khả năng mở rộng theo địa lý không?**
    * Có, chỉ cần bổ sung cấu hình địa lý trong backend.
18. **Làm sao để hệ thống hỗ trợ cả mobile app?**
    * Sử dụng React Native để phát triển ứng dụng di động.
19. **Hệ thống có tích hợp mạng xã hội không?**
    * Hiện chưa, nhưng có thể thêm chức năng chia sẻ thông tin qua mạng xã hội.
20. **Bạn đã học được gì từ dự án này?**
    * Kỹ năng sử dụng công nghệ mới và khả năng giải quyết vấn đề thực tế.
21. **Bạn có kế hoạch phát triển gì tiếp theo?**
    * Phát triển thêm các module AI để hỗ trợ chẩn đoán bệnh.
22. **Nếu có thêm thời gian, bạn sẽ làm gì?**
    * Bổ sung thêm chức năng đặt thuốc và tư vấn trực tuyến.
23. **Làm sao để cải thiện bảo mật hệ thống?**
    * Triển khai cơ chế xác thực hai lớp (2FA).
24. **Hệ thống có hỗ trợ kiểm tra lịch sử thay đổi không?**
    * Hiện chưa, nhưng có thể thêm bảng log để lưu vết các thay đổi.
25. **Bạn thấy thách thức lớn nhất khi phát triển hệ thống là gì?**
    * Hiểu rõ nghiệp vụ của phòng khám và đảm bảo hệ thống vận hành ổn định.