Khi xây dựng phần mềm quản lý hướng dẫn đồ án tốt nghiệp dưới dạng một ứng dụng web, cần xem xét các yêu cầu chi tiết hơn về công nghệ, kiến trúc, giao diện, và các yếu tố liên quan. Dưới đây là phân tích chi tiết:

### 1. \*\*Yêu cầu nghiệp vụ (Business Requirements)\*\*

- \*\*Quản lý tài khoản người dùng\*\*:

- \*\*Sinh viên\*\*:

- Đăng ký tài khoản với thông tin cá nhân (họ tên, mã sinh viên, lớp, email).

- Quản lý thông tin cá nhân, thay đổi mật khẩu.

- Xem danh sách đề tài đã đăng ký, theo dõi tiến trình thực hiện đồ án.

- \*\*Giảng viên\*\*:

- Đăng ký tài khoản với thông tin cá nhân (họ tên, email, bộ môn).

- Quản lý sinh viên hướng dẫn, theo dõi tiến độ đồ án.

- Đưa ra nhận xét, đánh giá và chấm điểm đồ án.

- \*\*Quản trị viên\*\*:

- Quản lý toàn bộ hệ thống (thêm/xóa/cập nhật tài khoản người dùng).

- Phân quyền truy cập cho các nhóm người dùng khác nhau.

- \*\*Quản lý đề tài\*\*:

- Tạo mới, cập nhật, và xóa đề tài.

- Phân công giảng viên hướng dẫn cho từng đề tài.

- Cho phép sinh viên đăng ký đề tài và giảng viên phê duyệt.

- Theo dõi tiến độ của từng đề tài: cập nhật thời gian thực về các mốc hoàn thành.

- \*\*Quản lý tiến trình\*\*:

- Sinh viên cập nhật tiến trình làm đồ án, đính kèm các tài liệu liên quan.

- Giảng viên nhận xét và đánh giá tiến trình theo từng giai đoạn.

- Tự động gửi thông báo nhắc nhở về các mốc thời gian quan trọng.

- \*\*Quản lý tài liệu\*\*:

- Hỗ trợ sinh viên và giảng viên tải lên và tải xuống tài liệu.

- Lưu trữ tài liệu theo từng đề tài, hỗ trợ các định dạng phổ biến (PDF, DOCX, ZIP, v.v.).

- Quản lý phiên bản tài liệu để theo dõi lịch sử thay đổi.

- \*\*Đánh giá và chấm điểm\*\*:

- Giảng viên chấm điểm và nhận xét cho đồ án của sinh viên.

- Lưu trữ kết quả đánh giá và tạo báo cáo tổng kết cho từng sinh viên.

### 2. \*\*Yêu cầu chức năng (Functional Requirements)\*\*

- \*\*Hệ thống quản lý đăng nhập và đăng ký\*\*:

- Hỗ trợ đăng nhập qua email và mật khẩu.

- Xác thực người dùng qua email khi đăng ký.

- Quên mật khẩu: Gửi email để người dùng đặt lại mật khẩu.

- \*\*Dashboard quản lý\*\*:

- \*\*Dashboard cho sinh viên\*\*: Xem danh sách đồ án, trạng thái hiện tại, tài liệu đính kèm và lịch trình.

- \*\*Dashboard cho giảng viên\*\*: Xem danh sách sinh viên đang hướng dẫn, trạng thái đồ án, và các mốc đánh giá.

- \*\*Dashboard cho quản trị viên\*\*: Thống kê số lượng đề tài, sinh viên, giảng viên, và các hoạt động liên quan.

- \*\*Quản lý thông báo\*\*:

- Tự động gửi email và thông báo trên hệ thống khi có sự kiện quan trọng (như cập nhật tiến trình, nhận xét của giảng viên, deadline).

- Tích hợp với hệ thống thông báo qua SMS nếu cần.

- \*\*Tìm kiếm và lọc dữ liệu\*\*:

- Chức năng tìm kiếm theo từ khóa (ví dụ: tên đề tài, tên sinh viên, tên giảng viên).

- Lọc theo trạng thái đồ án, mốc thời gian, và tên giảng viên.

### 3. \*\*Yêu cầu phi chức năng (Non-Functional Requirements)\*\*

- \*\*Hiệu năng\*\*:

- Hệ thống cần đáp ứng nhanh khi có nhiều người dùng truy cập đồng thời.

- Tối ưu hóa cơ sở dữ liệu để xử lý các truy vấn tìm kiếm và cập nhật nhanh chóng.

- \*\*Bảo mật\*\*:

- Mã hóa dữ liệu nhạy cảm (mật khẩu, thông tin cá nhân).

- Bảo vệ chống lại các cuộc tấn công phổ biến như SQL Injection, Cross-Site Scripting (XSS), Cross-Site Request Forgery (CSRF).

- Phân quyền truy cập dựa trên vai trò (Role-Based Access Control).

- \*\*Khả năng mở rộng\*\*:

- Thiết kế hệ thống để có thể dễ dàng mở rộng về số lượng người dùng và chức năng trong tương lai.

- Hỗ trợ tích hợp với các hệ thống quản lý khác trong trường học nếu cần.

- \*\*Tính ổn định và bảo trì\*\*:

- Đảm bảo hệ thống hoạt động liên tục, hạn chế tối đa thời gian ngừng hoạt động.

- Cung cấp công cụ giám sát và ghi log để dễ dàng phát hiện và sửa lỗi.

### 4. \*\*Yêu cầu về giao diện người dùng (UI/UX Requirements)\*\*

- \*\*Thân thiện và trực quan\*\*:

- Giao diện cần dễ hiểu, dễ sử dụng cho cả sinh viên và giảng viên.

- Đảm bảo tính tương thích trên các trình duyệt phổ biến (Chrome, Firefox, Safari).

- \*\*Thiết kế đáp ứng (Responsive Design)\*\*:

- Hỗ trợ truy cập từ nhiều thiết bị khác nhau (PC, máy tính bảng, điện thoại di động).

- Sử dụng các framework front-end như Bootstrap hoặc Tailwind CSS để tạo giao diện đáp ứng.

- \*\*Giao diện tối giản nhưng đầy đủ chức năng\*\*:

- Tập trung vào trải nghiệm người dùng, giảm thiểu số lần nhấp chuột để hoàn thành một công việc.

- Hỗ trợ đa ngôn ngữ nếu cần thiết.

### 5. \*\*Yêu cầu về kiến trúc và công nghệ (Architecture and Technology Requirements)\*\*

- \*\*Kiến trúc phần mềm\*\*:

- \*\*Frontend\*\*:

- Sử dụng các framework hiện đại như React.js, Angular, hoặc Vue.js để xây dựng giao diện người dùng.

- \*\*Backend\*\*:

- Sử dụng các ngôn ngữ và framework phổ biến như Node.js với Express, Django (Python), hoặc Spring Boot (Java) để phát triển API và xử lý logic nghiệp vụ.

- \*\*Cơ sở dữ liệu\*\*:

- Lựa chọn cơ sở dữ liệu quan hệ như MySQL, PostgreSQL, hoặc cơ sở dữ liệu NoSQL như MongoDB dựa trên yêu cầu lưu trữ và truy vấn dữ liệu.

- \*\*API\*\*:

- Phát triển các API RESTful để giao tiếp giữa frontend và backend.

- Có thể cân nhắc sử dụng GraphQL nếu cần các truy vấn dữ liệu phức tạp.

- \*\*Triển khai và vận hành (Deployment and Operations)\*\*:

- Triển khai trên các nền tảng cloud phổ biến như AWS, Azure, hoặc Google Cloud.

- Sử dụng Docker để đóng gói ứng dụng, giúp triển khai dễ dàng hơn.

- Thiết lập CI/CD để tự động hóa quá trình triển khai và cập nhật hệ thống.

### 6. \*\*Yêu cầu kiểm thử (Testing Requirements)\*\*

- \*\*Kiểm thử chức năng (Functional Testing)\*\*:

- Đảm bảo tất cả các chức năng nghiệp vụ và giao diện hoạt động đúng như yêu cầu.

- \*\*Kiểm thử hiệu năng (Performance Testing)\*\*:

- Kiểm thử khả năng chịu tải của hệ thống, đặc biệt là khi có nhiều người dùng truy cập đồng thời.

- \*\*Kiểm thử bảo mật (Security Testing)\*\*:

- Kiểm thử các biện pháp bảo mật, đảm bảo dữ liệu người dùng và hệ thống an toàn trước các mối đe dọa.

- \*\*Kiểm thử tích hợp (Integration Testing)\*\*:

- Đảm bảo các thành phần của hệ thống (frontend, backend, database) hoạt động mượt mà khi tích hợp với nhau.

### 7. \*\*Yêu cầu triển khai (Deployment Requirements)\*\*

- \*\*Môi trường triển khai\*\*:

- Phát triển và thử nghiệm trên môi trường staging trước khi triển khai lên môi trường production.

- Đảm bảo các bản sao lưu dữ liệu thường xuyên và cơ chế khôi phục trong trường hợp xảy ra sự cố.

- \*\*Hỗ trợ sau triển khai\*\*:

- Cung cấp tài liệu hướng dẫn sử dụng chi tiết cho người dùng.

- Đảm bảo có kế hoạch bảo trì, nâng cấp hệ thống định kỳ.

- Hỗ trợ kỹ thuật khi người dùng gặp sự cố hoặc có yêu cầu.

Với các yêu cầu chi tiết này, phần mềm quản lý hướng dẫn đồ án tốt nghiệp sẽ được xây dựng trên nền tảng web, cung cấp đầy đủ các chức năng và đảm bảo hiệu quả, an toàn và thân thiện với người dùng.