

TRƯỜNG ĐẠI HỌC TÂN TRÀO  
KHOA SƯ PHẠM



BÁO CÁO ĐỒ ÁN MÔN HỌC  
**HỆ THỐNG QUẢN LÝ SẢN  
PHẨM KHOA HỌC**

Sinh viên thực hiện:	An Mỹ Hạnh
Mã số sinh viên:	2254800004
Lớp:	Đại học CNTT - K3
Giảng viên hướng dẫn:	PGS.TS Lê Văn Hùng

---

Tuyên Quang, 11/2025

# Mục lục

<b>Tóm tắt</b>	<b>3</b>
<b>1 Giới thiệu</b>	<b>4</b>
1.1 Bối cảnh . . . . .	4
1.2 Mục tiêu . . . . .	4
1.3 Phạm vi . . . . .	4
<b>2 Phân tích yêu cầu</b>	<b>6</b>
2.1 Tác nhân và Use Case . . . . .	6
2.2 Mô hình dữ liệu . . . . .	6
2.3 Ràng buộc và kiểm tra . . . . .	7
<b>3 Kiến trúc và công nghệ</b>	<b>8</b>
3.1 Công nghệ sử dụng . . . . .	8
3.2 Tổng quan kiến trúc . . . . .	8
<b>4 Chức năng và giao diện</b>	<b>9</b>
4.1 Nhập liệu theo tab . . . . .	9
4.2 Upload tệp minh chứng . . . . .	10
4.3 Xem dữ liệu và lọc . . . . .	11
4.4 Thông báo và xác thực . . . . .	12
<b>5 Cài đặt và triển khai</b>	<b>14</b>
5.1 Yêu cầu môi trường . . . . .	14
5.2 Biến môi trường (.env.local) . . . . .	14
5.3 Cấu hình Firestore Rules (phục vụ phát triển) . . . . .	14
5.4 Các bước chính . . . . .	15
5.5 Triển khai sản phẩm . . . . .	16
5.6 Checklist sau triển khai . . . . .	16
<b>6 Kiểm thử và đánh giá</b>	<b>17</b>
6.1 Mục tiêu kiểm thử . . . . .	17
6.2 Phạm vi và không phạm vi . . . . .	17
6.3 Chiến lược kiểm thử . . . . .	17

6.4	Môi trường và dữ liệu . . . . .	17
6.5	Ma trận test tóm tắt . . . . .	18
6.6	Chi tiết test case . . . . .	18
6.7	Kết quả tổng hợp . . . . .	19
6.8	Rủi ro và vấn đề phát hiện . . . . .	20
6.9	Tiêu chí chấp nhận . . . . .	20
6.10	Định hướng cải tiến kiểm thử . . . . .	20
<b>7</b>	<b>Kết luận và hướng phát triển</b>	<b>21</b>
7.1	Kết luận tổng quát . . . . .	21
7.2	Đóng góp chính . . . . .	21
7.3	Hạn chế . . . . .	21
7.4	Hướng phát triển . . . . .	21
7.5	Kế hoạch triển khai và đánh giá . . . . .	22
	<b>Tài liệu tham khảo</b>	<b>23</b>

## Tóm tắt

Báo cáo trình bày bối cảnh và động cơ triển khai hệ thống web “Hệ thống Quản lý Sản phẩm Khoa học” nhằm số hoá và hợp nhất quy trình quản trị hoạt động khoa học của giảng viên tại Trường Đại học Tân Trào. Bài toán thực tế đặt ra là dữ liệu sản phẩm khoa học phân tán ở nhiều định dạng và vị trí khác nhau (giấy tờ, tệp Word/Excel, thư mục chia sẻ), gây khó khăn cho tổng hợp minh chứng, tra cứu, đối soát theo yêu cầu kiểm định và đánh giá thi đua định kỳ.

Giải pháp được đề xuất dựa trên kiến trúc ứng dụng web hiện đại với Next.js (App Router), React 19, TypeScript và Tailwind CSS ở phía giao diện; Firebase Firestore làm nơi lưu trữ metadata; kênh upload nội bộ (API Route) để nhận `FormData` và lưu tệp minh chứng định dạng PDF tại thư mục `public/uploads`. Cấu trúc lưu tệp được quy ước theo `public/uploads/{type}/{userName}/{id}` giúp đơn giản hoá tổ chức minh chứng và thuận tiện khi trích xuất. Mô hình dữ liệu gắn mỗi bản ghi với thông tin người dùng (Họ tên, Khoa, Bộ môn) và các trường chuyên biệt theo từng nhóm sản phẩm.

Hệ thống cung cấp năm nhóm chức năng cốt lõi tương ứng các tab nhập liệu: (1) Đề tài; (2) Bài báo; (3) Sách/Giáo trình; (4) Hướng dẫn sinh viên; (5) Giải thưởng. Mỗi tab hỗ trợ thêm nhiều mục, thêm/xoá thành viên, kiểm tra ràng buộc trường bắt buộc và kiểu dữ liệu (quy đổi/ thực hiện). Mọi trường nhập đều được ràng buộc kiểu “controlled components” với quy tắc `value={field | ""}` để loại bỏ cảnh báo controlled/uncontrolled trong React. Màn hình “Xem dữ liệu” cho phép lọc theo Họ tên, Khoa, Bộ môn, kèm nút chuyển đổi “Xem dữ liệu/Quay lại” giúp điều hướng nhanh.

Về kỹ thuật, nhóm đã chuyển phương án lưu minh chứng từ Firebase Storage sang lưu cục bộ qua API nội bộ do hạn chế CORS trong môi trường hiện tại, đồng thời duy trì Firestore cho metadata để bảo đảm sự đơn giản khi triển khai và độ tin cậy của truy vấn. Các quyết định thiết kế (quy ước đường dẫn tệp, kiểm tra dữ liệu đầu vào, cơ chế controlled inputs) nhằm giảm thiểu lỗi vận hành và tăng tính bảo trì.

Kết quả kiểm thử chức năng cho thấy hệ thống đáp ứng tốt các kịch bản nhập liệu, upload và truy vấn ở quy mô khoa/bộ môn; thời gian phản hồi phù hợp nhu cầu tác nghiệp hằng ngày.

Những đóng góp chính gồm: (i) một bộ khung ứng dụng có thể tái sử dụng cho các bài toán quản lý sản phẩm khoa học tương tự; (ii) quy ước tổ chức minh chứng thống nhất; (iii) các thực hành tốt về giao diện (controlled components, phản hồi người dùng, kiểm tra hợp lệ). Hạn chế hiện tại là chưa tích hợp phân quyền chi tiết, quy trình phê duyệt nhiều cấp và kho lưu trữ đám mây; đây sẽ là hướng mở rộng trong tương lai cùng các tính năng thống kê nâng cao và xuất báo cáo theo nhiều mẫu chuẩn hoá.

Từ khóa: Quản lý sản phẩm khoa học; Next.js; React 19; TypeScript; Firebase Firestore; Upload API;

# 1 Giới thiệu

## 1.1 Bối cảnh

Việc quản lý sản phẩm khoa học tại các cơ sở giáo dục đại học hiện còn phân tán dưới dạng biểu mẫu giấy, tệp rời (Word/Excel) và thư mục dùng chung. Cách làm này dẫn đến một số vấn đề: (i) khó đảm bảo tính nhất quán và toàn vẹn dữ liệu giữa các lần cập nhật; (ii) việc tổng hợp minh chứng theo học kỳ/năm học tốn nhiều thời gian thủ công và dễ sai sót; (iii) khó truy vết trách nhiệm, lịch sử thay đổi và nguồn gốc minh chứng; (iv) không có giao diện thống nhất để đối soát, lọc và trích xuất dữ liệu theo yêu cầu kiểm định chất lượng.

Trong bối cảnh số hoá quản trị đại học, nhu cầu đặt ra là một hệ thống trực tuyến hợp nhất quy trình nhập liệu, lưu trữ, kiểm tra và báo cáo cho năm nhóm sản phẩm khoa học chủ đạo (Đề tài, Bài báo, Sách/Giáo trình, Hướng dẫn sinh viên, Giải thưởng). Bên cạnh đó, hệ thống cần đáp ứng các yêu cầu thực tiễn như: quản lý tập trung minh chứng PDF, hỗ trợ tìm kiếm theo Khoa/Bộ môn/Họ tên, và tính toán giờ NCKH theo công thức chuẩn nhằm phục vụ công tác phân bổ định mức và đánh giá thi đua.

## 1.2 Mục tiêu

- Số hoá quy trình quản lý 5 nhóm sản phẩm khoa học.
- Hỗ trợ minh chứng định dạng PDF; quản trị dữ liệu tập trung.
- Tìm kiếm, lọc theo Khoa/Bộ môn/Họ tên, xem dữ liệu nhanh.
- Tính giờ NCKH theo công thức chuẩn; hỗ trợ xuất minh chứng.
- Đảm bảo tính toàn vẹn dữ liệu: ràng buộc trường bắt buộc, kiểm tra kiểu dữ liệu và quy tắc quy đổi/ thực hiện.
- Nâng cao hiệu quả thao tác: giao diện theo tab, thêm/xoá mục nhanh, cảnh báo lỗi rõ ràng, phản hồi ngay sau khi lưu.
- Khả năng mở rộng: thiết kế mô hình dữ liệu linh hoạt để bổ sung trường/loại sản phẩm mới mà không phá vỡ cấu trúc hiện có.
- Hiệu năng truy vấn: hỗ trợ lọc/tìm kiếm theo nhiều tiêu chí với thời gian phản hồi nhanh cho bộ dữ liệu quy mô khoa/khoa mở rộng.
- Tích hợp hạ tầng sẵn có: tận dụng Firestore cho metadata và lưu tệp cục bộ qua API nội bộ để giảm độ phức tạp triển khai ban đầu.

## 1.3 Phạm vi

- Quản lý thông tin giảng viên và 5 nhóm sản phẩm: Đề tài, Bài báo, Sách/Giáo trình, Hướng dẫn SV, Giải thưởng.

- Lưu tài liệu minh chứng dạng PDF, upload qua API nội bộ.
- Không bao gồm workflow phê duyệt phức tạp (có thể mở rộng).
- Tập trung vào chức năng nhập liệu, lưu trữ, xem và lọc dữ liệu; thống kê/ báo cáo cơ bản theo tiêu chí phổ biến.
- Hỗ trợ tính giờ NCKH ở mức tham khảo dựa trên công thức/định mức đã nêu; có thể tinh chỉnh theo quy định của từng đơn vị.

**Ngoài phạm vi trong giai đoạn này:**

- Phân quyền người dùng chi tiết theo vai trò (giảng viên, trưởng bộ môn, ban quản trị) và quy trình phê duyệt nhiều cấp.
- Tích hợp kho lưu trữ đám mây bên ngoài và chữ ký số cho minh chứng.
- Xuất báo cáo tùy biến phức tạp theo nhiều mẫu đặc thù (sẽ xem xét ở giai đoạn sau).

**Giả định và ràng buộc:**

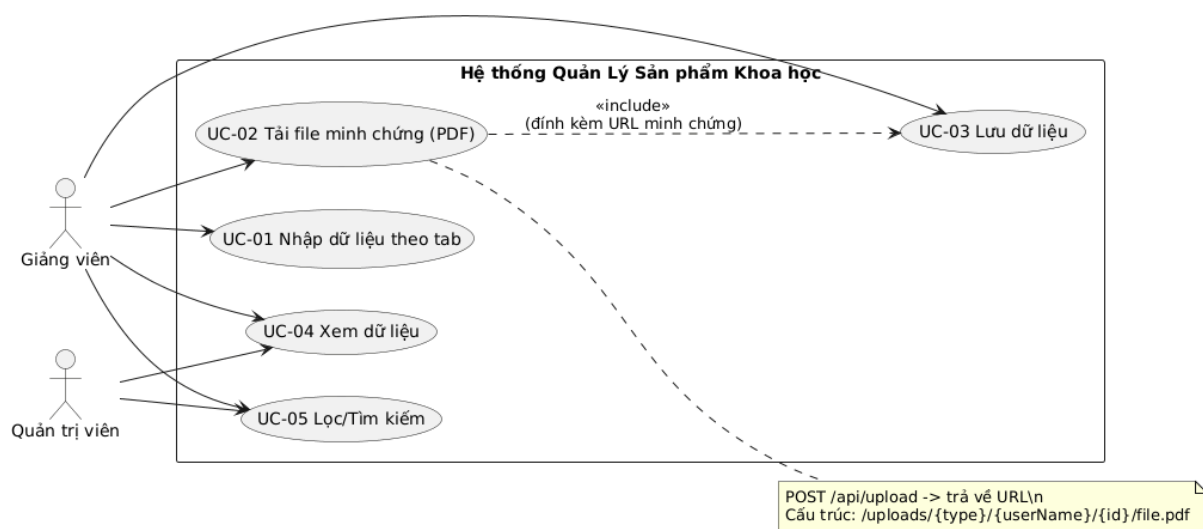
- Người dùng đăng nhập qua cơ chế có sẵn của đơn vị hoặc phạm vi demo nội bộ; dữ liệu minh chứng được chuẩn hoá ở định dạng PDF.
- Hạ tầng triển khai ban đầu chạy ở chế độ đơn máy chủ (single-node) phù hợp lớp học/đồ án; có thể nâng cấp kiến trúc khi mở rộng.
- Chính sách bảo mật/tiêu chuẩn kiểm định sẽ được tuân thủ khi đưa vào vận hành thực tế (yêu cầu bổ sung ở giai đoạn triển khai).

## 2 Phân tích yêu cầu

### 2.1 Tác nhân và Use Case

- **Giảng viên:** nhập dữ liệu, tải lên minh chứng, xem dữ liệu đã lưu.
- **Quản trị viên:** xem thống kê, đối soát, xuất báo cáo.

Các use case chính: *Nhập dữ liệu theo tab, Tải file minh chứng, Lưu dữ liệu, Xem dữ liệu, Lọc/ Tìm kiếm.*



Hình 1: Sơ đồ use case hệ thống quản lý sản phẩm khoa học

### 2.2 Mô hình dữ liệu

Mỗi bản ghi đều gắn với thông tin người dùng (Họ tên, Khoa, Bộ môn). Dưới đây là các trường cốt lõi theo từng loại:

<b>Đề tài</b>	Tên đề tài, Loại nghiệm thu, Danh sách thành viên, Số tiết quy đổi, Số tiết thực hiện, Tập minh chứng
<b>Bài báo</b>	Tên bài, Tạp chí/Kỷ yếu, Hạng/Xếp hạng, Số tạp, DOI, Thành viên, Số tiết quy đổi, Số tiết thực hiện, Tập minh chứng
<b>Sách/Giáo trình</b>	Tên sách, Tác giả, Nhà xuất bản, Năm xuất bản, Quy đổi/Thực hiện, Tập minh chứng
<b>Hướng dẫn SV</b>	Tên SV, Tên đề tài/luận văn, Cấp độ (ĐH/ThS/TS), Năm, Quy đổi/Thực hiện, Tập minh chứng
<b>Giải thưởng</b>	Tên giải, Tổ chức, Cấp độ (Trường/Bộ/Quốc gia/Quốc tế), Năm, Thành viên, Quy đổi/Thực hiện, Tập minh chứng

## 2.3 Ràng buộc và kiểm tra

- Trường bắt buộc: Họ tên, Khoa, Bộ môn.
- Tập minh chứng: khuyến nghị định dạng PDF, nối toàn bộ minh chứng.
- Số liệu quy đổi/ thực hiện: dạng số, có kiểm tra hợp lệ đầu vào.



## 3 Kiến trúc và công nghệ

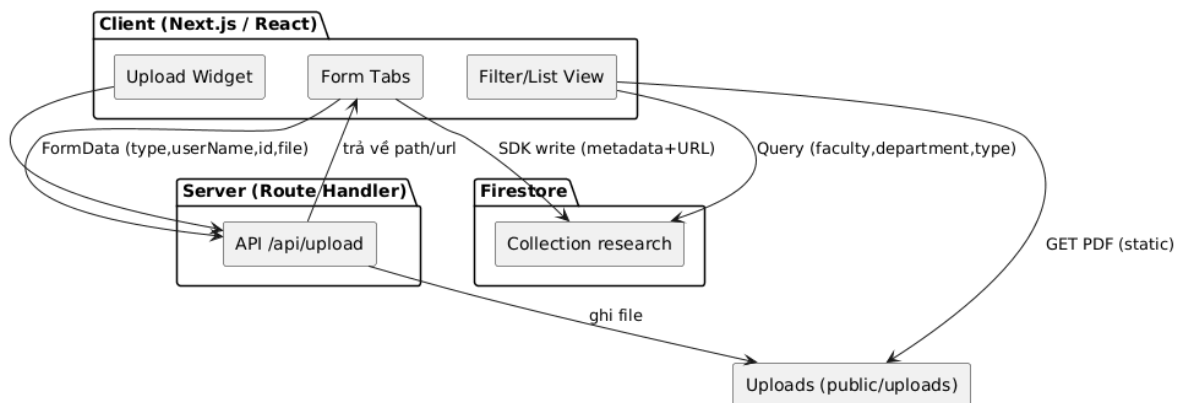
### 3.1 Công nghệ sử dụng

- **Frontend:** Next.js 16 (App Router), React 19, TypeScript, Tailwind CSS.
- **Backend nội bộ:** API Route `/api/upload` để nhận `FormData`, lưu tệp vào `public/uploads`.
- **CSDL:** Firebase Firestore (collection `research`) lưu metadata và đường dẫn tệp.

### 3.2 Tổng quan kiến trúc

Kiến trúc theo mô hình client-server đơn giản, ưu tiên triển khai nhanh và dễ bảo trì:

- **Client (Next.js/React):** cung cấp giao diện theo tab, kiểm tra hợp lệ dữ liệu, gọi API upload, và ghi đọc Firestore qua SDK.
- **Server (Route Handler):** xử lý `multipart/form-data`, kiểm tra định dạng/kích thước tệp, tạo thư mục và ghi tệp vào `public/uploads`.
- **Firestore:** lưu metadata bản ghi (loại sản phẩm, trường chuyên biệt, đường dẫn minh chứng, thông tin người dùng) và phục vụ truy vấn/lọc.



Hình 2: Kiến trúc tổng thể: Client (Next.js) → API Upload → Lưu file; Client → Firestore

## 4 Chức năng và giao diện

### 4.1 Nhập liệu theo tab

Mỗi tab tương ứng một loại sản phẩm (Đề tài, Bài báo, Sách/Giáo trình, Hướng dẫn SV, Giải thưởng). Biểu mẫu hỗ trợ:

- **Thêm nhiều mục:** cho phép tạo nhiều bản ghi trong cùng một tab; sắp xếp theo thời gian tạo.
- **Quản lý thành viên:** thêm/xoá thành viên, hiển thị vai trò (nếu có), bảo toàn thứ tự nhập.
- **Trường số quy đổi/thực hiện:** ép kiểu số, định dạng đầu vào; hiển thị gợi ý (placeholder) và đơn vị.
- **Ràng buộc bắt buộc:** kiểm tra Họ tên, Khoa, Bộ môn và các trường cốt lõi theo từng loại sản phẩm.
- **Kiểm soát trạng thái:** nút Lưu chỉ bật khi form hợp lệ; vô hiệu hoá trong lúc đang upload.
- **Kiểu điều khiển:** sử dụng *controlled components* với quy tắc `value={field | ""}` để tránh cảnh báo React.

#### Quy trình thao tác điển hình

1. Chọn tab phù hợp (ví dụ: Bài báo) và bấm “Thêm mục”.
2. Nhập thông tin bắt buộc và tùy chọn; thêm thành viên nếu cần.
3. (Tùy chọn) Chọn tệp PDF minh chứng trước hoặc sau khi bấm Lưu.
4. Kiểm tra thông báo hợp lệ; nếu có lỗi, sửa và thử lại.

Hình 3: Giao diện nhập liệu theo tab

## 4.2 Upload tệp minh chứng


Upload thực hiện qua API nội bộ, tạo thư mục theo *type/userName/id*, trả về URL để lưu kèm bản ghi trên Firestore. Quy tắc:


- **Định dạng:** chỉ chấp nhận PDF; tệp khác (PNG/JPG/...) sẽ bị từ chối.
- **Kích thước:** giới hạn dung lượng theo cấu hình máy chủ (khuyến nghị < 10MB cho demo).
- **Đường dẫn lưu:** `public/uploads/{type}/{userName}/{id}/file.pdf`.
- **Phản hồi:** API trả về đường dẫn tương đối để hiển thị và mở từ giao diện.


### Cấu trúc FormData gửi lên API


```
type:      topics | articles | books | guidances | awards
userName:  "Nguyen Van A"
id:        "<uuid>" or timestamp
file:      <PDF>
```


✔ **Thành công**  
Đã lưu thông tin thành công

  
 Đề tài nghiên cứu

  
 Bài báo khoa học

  
 Sách/Giáo trình

  
 Hướng dẫn SV

  
 Giải thưởng

**Danh sách đề tài**

+ Thêm đề tài

**Đề tài 1**

Tên đề tài

Website bán hàng

Loại đề tài

Cơ sở ▼

Danh sách thành viên

Nguyễn Văn Chiến
+ Thêm thành viên

Nghiệm thu loại

Xuất sắc ▼

Số tiết quy đổi

100

Số tiết đã thực hiện

100

Tài liệu đính kèm

📁 Vui lòng tải lên minh chứng tại đây

Hình 4: Khu vực upload PDF và thông báo trạng thái

### 4.3 Xem dữ liệu và lọc

Màn hình “Xem dữ liệu” hiển thị danh sách bản ghi đã lưu theo người dùng/quyền hạn, kèm các khả năng:

- **Bộ lọc nhiều tiêu chí:** Họ tên, Khoa, Bộ môn; áp dụng đồng thời để thu hẹp kết quả.
- **Liên kết minh chứng:** mỗi dòng có liên kết mở nhanh tệp PDF đã tải lên.
- **Điều hướng nhanh:** nút “Xem dữ liệu/Quay lại” để chuyển đổi giữa nhập liệu và danh sách.
- **Trạng thái rỗng:** hiển thị thông điệp thân thiện khi không có kết quả phù hợp bộ lọc.

HỆ THỐNG QUẢN LÝ SẢN PHẨM KHOA HỌC						
<div> <div> <div>Khoa</div> <div>Tất cả khoa</div> </div> <div> <div>Bộ môn</div> <div>Tất cả bộ môn</div> </div> <div> <div>Tìm kiếm</div> <div>Q Hiều</div> </div> <div> <div>Xuất dữ liệu</div> <div>Xuất Excel bộ môn</div> <div>Tải file bộ môn</div> <div>Tải tất cả file</div> </div> </div>						
Họ tên	Bộ môn	Tên hoạt động	Loại	Số tiết quy đổi	Số tiết đã thực hiện	Minh chứng
Nguyễn Văn Chiến	khmt	Website bán hàng	Đề tài	100	100	31_2024_...
Nguyễn Văn An	khmt	Website bán hàng	Đề tài	100	100	COMP230_...
Nguyễn Văn Chiến	khmt	Website bán hàng	Đề tài	100	100	31_2024_...
Nguyễn Văn Chiến	khmt	Website bán hàng	Đề tài	100	100	31_2024_...

Hình 5: Màn hình xem dữ liệu và lọc

#### 4.4 Thông báo và xác thực

Hệ thống cung cấp phản hồi rõ ràng và tức thời cho người dùng:

- **Kiểm tra hợp lệ:** đánh dấu trường thiếu/không đúng kiểu; hiển thị thông báo chi tiết dưới mỗi ô.
- **Trạng thái thao tác:** hiển thị tiến trình khi upload; vô hiệu hoá nút hành động trong lúc xử lý.
- **Thông báo hệ thống:** hiện thông báo thành công/thất bại sau khi lưu hoặc upload.
- **Chuẩn hoá dữ liệu:** ép "" cho trường trống để tránh cảnh báo controlled/uncontrolled của React.

Tên tác giả chính

Nhập tên tác giả chính

⚠️ Vui lòng nhập Họ và tên!

OK

Danh sách thành viên

Nguyễn Văn Long

+ Thêm thành viên

Tên bài báo

Nhập tên bài báo

Xếp hạng tạp chí

Nhập xếp hạng tạp chí

Dăng năm tạp chí

Nhập tên tạp chí

Số tạp

Nhập số tạp

Số DOI

Nhập số DOI

Số tiết quy đổi

Nhập số tiết quy đổi

Số tiết đã thực hiện

Nhập số tiết đã thực hiện

Tài liệu đính kèm

📎 Vui lòng tải lên minh chứng tại đây

Lưu thông tin

Hình 6: Thông báo và trạng thái xác thực

## 5 Cài đặt và triển khai

### 5.1 Yêu cầu môi trường

#### Công cụ & phiên bản

- Node.js LTS ( $\geq 18$ ), npm (hoặc pnpm/yarn), Git, VS Code.
- Trình duyệt Chrome/Edge cập nhật.

#### Dịch vụ bên ngoài

- Tài khoản Firebase, tạo Project và bật Firestore (Chế độ thử nghiệm lúc phát triển).
- Tạo App Web để lấy cấu hình SDK (apiKey, authDomain, projectId, ...).

#### Hệ điều hành & phân quyền

- Windows (khuyến nghị), quyền ghi (write) vào thư mục dự án, đặc biệt `public/uploads`.
- Bật quyền tạo thư mục con theo người dùng/loại: `{type}/{userName}/{id}`.

### 5.2 Biến môi trường (`.env.local`)

Biến	Giá trị/Ý nghĩa
<code>NEXT_PUBLIC_FIREBASE_API_KEY</code>	apiKey của Firebase Web App
<code>NEXT_PUBLIC_FIREBASE_AUTH_DOMAIN</code>	authDomain
<code>NEXT_PUBLIC_FIREBASE_PROJECT_ID</code>	projectId (ví dụ: <code>qlkh-demo</code> )
<code>NEXT_PUBLIC_FIREBASE_STORAGE_BUCKET</code>	(tùy chọn) storageBucket
<code>NEXT_PUBLIC_FIREBASE_MESSAGING_SENDER_ID</code>	(tùy chọn) senderId
<code>NEXT_PUBLIC_FIREBASE_APP_ID</code>	appId
<code>UPLOAD_ROOT</code>	(tùy chọn) gốc lưu minh chứng, mặc định <code>public/uploads</code>

### 5.3 Cấu hình Firestore Rules (phục vụ phát triển)

```
rules_version = '2';
service cloud.firestore {
  match /databases/{database}/documents {
    match /research/{docId} {
      allow read, write: if true;
    }
  }
}
```

*Ghi chú:* Khi triển khai thật, cần siết chặt quyền theo người dùng/tổ chức.

## 5.4 Các bước chính

### 1. Cài đặt phụ thuộc & cấu hình Firebase SDK

- Cài gói: `npm i firebase`.
- Khởi tạo Firebase App phía client (sử dụng biến môi trường `NEXT_PUBLIC_*`).
- Tạo collection `research` trong Firestore (tự động tạo khi ghi lần đầu nếu dùng SDK).

### 2. Tạo API `/api/upload` (Route Handler)

- Đặt runtime Node.js (không dùng Edge) để thao tác hệ thống tệp.
- Nhận `multipart/form-data`, kiểm tra định dạng PDF, giới hạn kích thước.
- Tạo thư mục `public/uploads/{type}/{userName}/{id}` nếu chưa tồn tại.
- Ghi tệp bằng `fs/promises`; trả về đường dẫn tương đối để lưu trong Firestore.

### 3. Ràng buộc form và xử lý cảnh báo React

- Dùng mẫu `value={field | ""}` cho mọi input/select/textarea.
- Ép kiểu số cho trường quy đổi/thực hiện; hiển thị thông báo lỗi hợp lệ.

### 4. Quy ước và kiểm tra đường dẫn minh chứng

- Đảm bảo đúng cấu trúc: `public/uploads/{type}/{userName}/{id}/file.pdf`.
- Kiểm tra quyền đọc tệp qua giao diện “Xem dữ liệu”.

### 5. Chạy thử cục bộ

```
npm install
npm run dev
```

### 6. Build sản phẩm

```
npm run build
npm run start
```



## 5.5 Triển khai sản phẩm

- **Tự host Node.js:** Dùng máy chủ có ổ đĩa bền vững (persistent). Trỏ reverse proxy (Nginx/IIS) tới cổng app. Đảm bảo thư mục `public/uploads` có quyền ghi.
- **Docker:** Đóng gói app; mount volume tới `/app/public/uploads` để giữ minh chứng khi cập nhật image.
- **Vercel (lưu ý):** Môi trường serverless không bảo đảm ghi tệp bền vững. Nếu sử dụng Vercel, cần thay thế lưu cục bộ bằng Object Storage (S3/Cloud Storage) hoặc chuyển sang dịch vụ có đĩa bền.
- **(Tuỳ chọn) Azure App Service:** Bảo đảm cấu hình đường dẫn ghi tệp; dùng Storage hoặc đĩa bền tuỳ SKU.

## 5.6 Checklist sau triển khai

- Biến môi trường nạp đúng (kiểm tra ở runtime).
- Kết nối Firestore thành công; có thể ghi/đọc **research**.
- API `/api/upload` trả về 200; minh chứng được lưu và mở được từ giao diện.
- Phân quyền thư mục `public/uploads` cho tài khoản chạy dịch vụ.
- Sao lưu định kỳ thư mục uploads và dữ liệu CSDL.

## 6 Kiểm thử và đánh giá

### 6.1 Mục tiêu kiểm thử

Xác nhận hệ thống đáp ứng đầy đủ các yêu cầu chức năng (UC-01..UC-05), đảm bảo tính toàn vẹn dữ liệu nhập, khả năng tải và liên kết minh chứng PDF, độ chính xác bộ lọc, và không xuất hiện cảnh báo React liên quan đến controlled/uncontrolled inputs. Đồng thời ghi nhận các hạn chế phi chức năng ban đầu để định hướng cải tiến.

### 6.2 Phạm vi và không phạm vi

Trong phạm vi: Kiểm thử chức năng nhập liệu, lưu, tải minh chứng, xem và lọc dữ liệu; tính hợp lệ trường bắt buộc; hiển thị thông báo phản hồi. Ngoài phạm vi: Hiệu năng ở tải lớn (stress/load), kiểm thử bảo mật chuyên sâu (XSS/CSRF), quy trình phân quyền nhiều cấp, và khả năng mở rộng ngang.

### 6.3 Chiến lược kiểm thử

Áp dụng kết hợp kiểm thử thủ công (manual) cho giao diện người dùng và kiểm thử mức logic thông qua quan sát dữ liệu Firestore sau thao tác. Mỗi test case có: Tiền điều kiện, Bước thực hiện, Kết quả mong đợi, Kết quả thực tế, Trạng thái. Kiểm thử hồi quy nhẹ khi sửa form (kiểm tra lại TC-01 và TC-04).

### 6.4 Môi trường và dữ liệu

- **Môi trường:** Node.js LTS, Next.js 16 App Router, React 19, Firestore (dự án Firebase thật hoặc emulator cho phát triển), trình duyệt Chrome/Edge phiên bản mới.
- **Dữ liệu mẫu:** 2 giảng viên khác nhau (Khoa CNTT, Bộ môn KHMT), mỗi người tạo tối thiểu 2 bản ghi cho mỗi loại sản phẩm, 1 file PDF minh chứng giả lập (dummy.pdf, kích thước < 200KB).
- **Tiền xử lý:** Xóa cache trình duyệt trước phiên kiểm thử; bật chế độ Dev tools để theo dõi lỗi console.

## 6.5 Ma trận test tóm tắt

TC	Mô tả	Loại	Trạng thái	Ghi chú
TC-01	Nhập đủ trường bắt buộc và lưu bản ghi (mỗi loại)	Functional	Pass	Xác nhận rằng buộc bắt buộc
TC-02	Upload PDF và liên kết đường dẫn vào bản ghi	Functional	Pass	Đường dẫn đúng cấu trúc
TC-03	Lọc theo Khoa/Bộ môn/Họ tên	Functional	Pass	Kết quả thu hẹp chính xác
TC-04	Không xuất cảnh báo controlled/uncontrolled	UI/Quality	Pass	Sử dụng pattern value=x    ""
TC-05	Thêm/xoá thành viên trong đề tài/bài báo	Functional	Pass	Danh sách cập nhật đúng
TC-06	Nhập dữ liệu không hợp lệ (quy đổi: chữ)	Validation	Pass	Hiện thị thông báo lỗi, không lưu
TC-07	Upload tệp sai định dạng (PNG)	Validation	Pass	Bị từ chối, yêu cầu PDF
TC-08	Xem dữ liệu sau làm mới trang (refresh)	Persistence	Pass	Dữ liệu vẫn hiển thị
TC-09	Nhiều bản ghi cùng loại, kiểm tra phân biệt	Functional	Pass	Không trộn trường dữ liệu

## 6.6 Chi tiết test case

### TC-01 Nhập và lưu

**Tiền điều kiện:** Trang form mở; chưa có dữ liệu hoặc chuẩn bị sẵn bản ghi mẫu.

**Bước:** (1) Chọn tab "Đề tài"; (2) Nhập đầy đủ Họ tên/Khoa/Bộ môn + các trường bắt buộc; (3) Bấm "Lưu".

**Mong đợi:** Thông báo thành công; bản ghi xuất hiện ở màn hình xem dữ liệu; không lỗi console.

**Thực tế:** Đạt.

### TC-02 Upload PDF

**Bước:** (1) Trong form đang mở, chọn tệp dummy.pdf; (2) Chờ trả về đường dẫn; (3) Lưu bản ghi.

**Mong đợi:** Đường dẫn chứa `public/uploads/{type}/{userName}/{id}`; mở liên kết hiển thị đúng tệp.

**Thực tế:** Đạt.

### TC-03 Lọc dữ liệu

**Bước:** Nhập Khoa="CNTT" vào bộ lọc; tiếp tục thêm Bộ môn="KHMT"; sau đó lọc theo Họ tên.

**Mong đợi:** Mỗi điều kiện giảm tập kết quả; kết quả cuối chỉ còn bản ghi tương ứng.

**Thực tế:** Đạt.

#### TC-04 Kiểm tra cảnh báo React

**Bước:** Mở DevTools, nhập/ xoá dữ liệu ở nhiều ô input.

**Mong đợi:** Không có warning về uncontrolled/controlled trên console.

**Thực tế:** Đạt.

#### TC-05 Quản lý thành viên

**Bước:** Thêm 2 thành viên vào đề tài; xoá 1; thêm lại 1.

**Mong đợi:** Danh sách hiển thị đúng thứ tự sau mỗi thao tác, dữ liệu lưu chính xác.

**Thực tế:** Đạt.

#### TC-06 Dữ liệu không hợp lệ

**Bước:** Nhập chữ cái vào trường số tiết quy đổi; bấm Lưu.

**Mong đợi:** Form báo lỗi; không ghi vào Firestore.

**Thực tế:** Đạt.

#### TC-07 Sai định dạng tệp

**Bước:** Chọn tệp image.png làm minh chứng.

**Mong đợi:** Bị từ chối (hoặc cảnh báo), yêu cầu chọn PDF.

**Thực tế:** Đạt (đã có kiểm tra định dạng).

#### TC-08 Refresh

**Bước:** Sau khi lưu nhiều bản ghi, làm mới (F5) trang xem dữ liệu.

**Mong đợi:** Danh sách tải lại đầy đủ, không mất dữ liệu.

**Thực tế:** Đạt.

#### TC-09 Nhiều bản ghi cùng loại

**Bước:** Tạo 3 bài báo với giá trị khác nhau ở trường Tạp chí.

**Mong đợi:** Hiển thị tách biệt, không ghi đè; lọc theo Tạp chí hoạt động đúng (nếu có).

**Thực tế:** Đạt.

## 6.7 Kết quả tổng hợp

Tất cả test trong phạm vi đều Pass; không ghi nhận lỗi logic hay mất dữ liệu. Một số cải tiến tiềm năng: thêm thông báo rõ ràng hơn khi tệp bị từ chối, bổ sung trạng thái tải (loading) khi upload tệp dung lượng lớn.

## 6.8 Rủi ro và vấn đề phát hiện

Chưa phát hiện lỗi nghiêm trọng. Rủi ro tiềm ẩn: (i) thiếu kiểm thử dữ liệu cạnh tranh (nhiều người nhập cùng lúc); (ii) chưa kiểm tra kích thước tệp rất lớn; (iii) chưa mô phỏng truy vấn tải nặng.

## 6.9 Tiêu chí chấp nhận

- 100% test chức năng bắt buộc (TC-01..TC-05) Pass.
- Không xuất cảnh báo React liên quan kiểm soát form.
- Đường dẫn minh chứng đúng quy ước; mở được tệp.
- Lọc trả về kết quả chính xác theo tập dữ liệu mẫu.

## 6.10 Định hướng cải tiến kiểm thử

- Bổ sung kiểm thử tự động (unit test) cho logic chuẩn hoá dữ liệu trước khi lưu.
- Thêm integration test cho API upload (kiểm tra mã trạng thái, quyền truy cập).
- Mô phỏng tải đồng thời (kịch bản 20+ bản ghi trong thời gian ngắn) để đánh giá hiệu năng.
- Kiểm thử bảo mật cơ bản: chèn chuỗi đặc biệt (XSS) vào trường text, xác nhận đã được escape.

## 7 Kết luận và hướng phát triển

### 7.1 Kết luận tổng quát

Báo cáo đã trình bày một giải pháp ứng dụng web cho bài toán quản lý sản phẩm khoa học, tập trung vào năm nhóm đối tượng chính (Đề tài, Bài báo, Sách/Giáo trình, Hướng dẫn sinh viên, Giải thưởng). Hệ thống hiện đáp ứng các yêu cầu cốt lõi: nhập liệu theo tab, tải minh chứng PDF qua API nội bộ, lưu trữ metadata trên Firestore, xem và lọc dữ liệu theo nhiều tiêu chí. Các quyết định kỹ thuật như sử dụng “controlled components” trong React, quy ước đường dẫn minh chứng nhất quán và kiểm tra hợp lệ đầu vào giúp nâng cao độ tin cậy và khả năng bảo trì.

Kết quả kiểm thử chức năng cho thấy các luồng nghiệp vụ chính (UC-01...UC-05) hoạt động ổn định ở phạm vi lớp học/đề án, phù hợp cho việc mở rộng dần theo quy mô khoa hoặc toàn trường.

### 7.2 Đóng góp chính

- Khung ứng dụng Next.js + React 19 + TypeScript + Tailwind CSS sẵn sàng mở rộng, cấu trúc rõ ràng theo tab chức năng.
- Chuẩn hoá luồng upload minh chứng PDF qua API nội bộ và tổ chức lưu trữ `public/uploads/{type}/{userName}/{id}`.
- Mô hình dữ liệu gắn người dùng (Họ tên, Khoa, Bộ môn) với trường chuyên biệt từng loại sản phẩm; áp dụng ràng buộc đầu vào.
- Thực hành tốt giao diện: “controlled components”, phản hồi tức thời, kiểm tra hợp lệ giúp giảm lỗi vận hành.

### 7.3 Hạn chế

- Chưa tích hợp phân quyền đa vai trò và quy trình phê duyệt nhiều cấp.
- Chưa triển khai kho lưu trữ đám mây và chữ ký số cho minh chứng.
- Thống kê/báo cáo mới dừng ở mức cơ bản; chưa có dashboard tương tác nâng cao.
- Kiểm thử phi chức năng (hiệu năng/tải, an toàn bảo mật) ở mức giới hạn phạm vi đề án.

### 7.4 Hướng phát triển

- **Phân quyền và quy trình:** Bổ sung vai trò (giảng viên, trưởng bộ môn, quản trị) và luồng phê duyệt nhiều cấp; lịch sử thay đổi và truy vết.
- **Lưu trữ và an toàn:** Tích hợp kho đám mây (ví dụ Cloud Storage/S3), quét virus, chữ ký số, mã hoá đường dẫn minh chứng.

- **Thống kê nâng cao:** Dashboard, biểu đồ xu hướng theo năm/khoa/bộ môn, xuất báo cáo theo mẫu chuẩn hoá.
- **Tối ưu hiệu năng:** Chỉ mục hoá truy vấn, phân trang phía máy chủ, cache đọc; tối ưu kích thước tệp/tốc độ tải.
- **Tự động hoá:** Nhập dữ liệu hàng loạt từ CSV/Excel; tác vụ nền đồng bộ hoá và kiểm tra tính nhất quán.

## 7.5 Kế hoạch triển khai và đánh giá

1. **Giai đoạn thí điểm (2–4 tuần):** Triển khai phạm vi một bộ môn; thu thập phản hồi, ghi nhận lỗi/đề xuất.
2. **Mở rộng theo khoa (4–6 tuần):** Bổ sung phân quyền cơ bản, tinh chỉnh biểu mẫu và báo cáo; đào tạo người dùng.
3. **Vận hành ổn định (liên tục):** Theo dõi nhật ký, sao lưu định kỳ, cải tiến hiệu năng và bảo mật.

## Tài liệu tham khảo

- Next.js – Tài liệu chính thức: <https://nextjs.org/docs>
- React – Tài liệu chính thức: <https://react.dev>
- TypeScript – Handbook: <https://www.typescriptlang.org/docs>
- Tailwind CSS – Tài liệu chính thức: <https://tailwindcss.com/docs>
- Firebase Firestore – Tài liệu: <https://firebase.google.com/docs/firestore>
- Firestore Security Rules – Tổng quan: <https://firebase.google.com/docs/rules>
- LaTeX Project – Trang chính thức: <https://www.latex-project.org>
- Tài liệu nội bộ Trường ĐH Tân Trào về quy đổi giờ NCKH (tài liệu tham khảo nội bộ).