

VIETNAM NATIONAL UNIVERSITY HO CHI MINH CITY
HO CHI MINH CITY UNIVERSITY OF TECHNOLOGY
FACULTY OF COMPUTER SCIENCE AND ENGINEERING



Programming Intergration Project (CO3103)

CC02 - Group 01

File Sharing Web Frontend

Advisor(s): Lê Đình Thuận

Student(s): Lê Nguyễn Bảo Minh ID: 2352742

Cao Nguyễn Gia Khánh ID: 2352514

Mai Trung Kiên ID: 2352641

Thi Minh Thức ID: D

HO CHI MINH CITY, DECEMBER 2025



MEMBER LIST
CLASS CC02 - GROUP 01

No.	Fullname	Student ID	Work load	% done
1	Lê Nguyễn Bảo Minh	2352742	Work A	100%
2	Cao Nguyễn Gia Khánh	2352514	Work B	100%
3	Mai Trung Kiên	2352641	Work C	100%
4	Thi Minh Thúc	D	Work D	100%

Table 0.1: Team members' contributions



Contents

1	Tổng quan dự án	4
1.1	Vai trò	4
1.2	Mục tiêu	4
2	Technology Stack	5
3	Project Specification	6
3.1	Giới hạn dự án	6
3.1.1	Phạm vi triển khai của Frontend	6
3.1.2	Hợp tác với Backend	6
4	Functional requirement	7
4.1	Xác thực & Phân quyền	7
4.2	Quản lý Tập tin (Tải lên)	7
4.3	Bảng điều khiển Người dùng	8
4.4	Quản trị	8
5	Non-functional Requirements	9
5.1	Hiệu năng	9
5.2	Tính dễ sử dụng	9
5.3	Khả năng bảo trì	9
6	UI Design & screen flow	10
6.1	Screen Flow	10
6.1.1	Public	10
6.1.2	Authentication Screens	10
6.1.3	User Screens	12
6.1.4	Public File Access	13
6.1.5	Admin Screens	14

List of Tables

0.1	Team members' contributions	2
-----	---------------------------------------	---



1 Tổng quan dự án

1.1 Vai trò

Lập trình viên Frontend

1.2 Mục tiêu

Mục tiêu chính của dự án là thiết kế và xây dựng một giao diện web phản hồi nhanh (responsive), bảo mật và lấy người dùng làm trung tâm cho hệ thống chia sẻ tệp tin tạm thời. Hệ thống được xây dựng dựa trên repository `file-sharing-fe-web`, đóng vai trò là cầu nối mượt mà giữa người dùng cuối và các logic bảo mật phức tạp được định nghĩa trong *File Sharing System API* ở backend.

Các trách nhiệm chính và mục tiêu triển khai bao gồm:

- **Quy trình Tải lên Bảo mật:** Phát triển giao diện upload động, thực thi các quy tắc validation ngay tại phía client (ví dụ: đảm bảo ngày *availableFrom* luôn trước ngày *availableTo*) và cấu hình quyền riêng tư có điều kiện (ẩn/hiện trường mật khẩu và danh sách email dựa trên trạng thái public/private).
- **Xác thực Nâng cao:** Triển khai bộ tính năng xác thực toàn diện, vượt ra ngoài việc đăng nhập thông thường để hỗ trợ quy trình Mật khẩu Một lần theo Thời gian (TOTP), bao gồm hiển thị mã QR để thiết lập và xác minh hai bước (2FA) khi đăng nhập.
- **Dashboard & Quản lý:** Xây dựng bảng điều khiển trung tâm cho phép người dùng đã đăng nhập theo dõi vòng đời các tệp tin của họ, xem trạng thái thời gian thực (Đang hoạt động, Chờ xử lý, Đã hết hạn) và thống kê chi tiết lượt tải xuống.
- **Xử lý Truy cập Công khai:** Xây dựng trang đích (landing page) tải xuống an toàn, có khả năng diễn giải các mã trạng thái HTTP để phản hồi phù hợp với người dùng, chẳng hạn như xử lý lỗi "Gone" (410) cho file hết hạn, "Locked" (423) cho file chưa đến giờ mở, và các yêu cầu nhập mật khẩu bảo vệ.



2 Technology Stack

Dự án sử dụng các công nghệ sau:

- **Framework cốt lõi:** `Next.js 16` (`React 19`) - Tận dụng khả năng Server-Side Rendering (SSR) và tối ưu hóa hiệu năng mới nhất.
- **Ngôn ngữ lập trình:** `TypeScript` (chiếm 77.9% source code) - Đảm bảo tính chặt chẽ về kiểu dữ liệu và giảm thiểu lỗi runtime.
- **Styling (Giao diện):** `Tailwind CSS v4` - Sử dụng kết hợp với `clsx` và `tailwind-merge` để xử lý các class động một cách linh hoạt.
- **HTTP Client:** `Axios` - Quản lý các request API, xử lý Interceptors cho việc đính kèm Token và xử lý lỗi toàn cục.
- **Quản lý trạng thái (State Management):** Sử dụng `React Hooks` và `Context API` (tích hợp sẵn trong `React/Next.js`) để quản lý trạng thái phiên làm việc và dữ liệu người dùng.
- **Tiện ích khác:**
 - `js-cookie`: Quản lý lưu trữ Token bảo mật trong Cookie.
 - `sonner`: Hiển thị thông báo (toast notifications) cho trải nghiệm người dùng mượt mà.
 - `lucide-react`: Bộ icon hiện đại, nhẹ nhàng cho giao diện.

3 Project Specification

3.1 Giới hạn dự án

Phần Frontend của hệ thống File Sharing Web được xây dựng nhằm cung cấp giao diện người dùng hiện đại, dễ sử dụng và bảo mật, cho phép người dùng tải lên, chia sẻ và quản lý các tệp tin tạm thời. Phạm vi của Frontend được xác định rõ ràng trong bối cảnh hệ thống bao gồm cả nhóm Backend (Go) và nhóm Frontend (Web).

3.1.1 Phạm vi triển khai của Frontend

Frontend chịu trách nhiệm cho toàn bộ các nhiệm vụ liên quan đến giao diện và tương tác người dùng, bao gồm:

- Thiết kế và hiện thực UI/UX cho toàn bộ luồng hoạt động của người dùng: đăng nhập, thiết lập TOTP, tải lên file, truy cập file chia sẻ, quản lý file trong dashboard.
- Thực hiện toàn bộ xử lý logic ở phía client: validation dữ liệu, quản lý trạng thái, thông báo lỗi/thành công, điều hướng người dùng, hiển thị thời gian hiệu lực file.
- Tương tác trực tiếp với Backend bằng REST API để gọi authentication, upload file, truy xuất metadata, tải file xuống, cập nhật hoặc xóa file.
- Xây dựng cấu trúc hệ thống Frontend (routing, service layer, component structure) theo Next.js App Router.
- Đảm bảo hệ thống chạy được trong môi trường production bằng Docker Compose và tuân thủ yêu cầu CI/CD của dự án.

3.1.2 Hợp tác với Backend

- Frontend sử dụng API mà nhóm Backend cung cấp, không tự định nghĩa logic riêng.
- Giao diện phải tương thích đa nền tảng và hỗ trợ SSR theo kiến trúc Next.js.
- Hệ thống phải được containerized hoàn toàn và deploy bằng GitHub Actions sử dụng self-hosted runners.

4 Functional requirement

4.1 Xác thực & Phân quyền

- **Đăng ký:** Người dùng có thể đăng ký tài khoản mới bằng email, tên đăng nhập và mật khẩu.
- **Đăng nhập:** Người dùng có thể xác thực bằng email và mật khẩu.
- **Xác thực Hai yếu tố (TOTP):**
 - Người dùng có thể kích hoạt TOTP cho tài khoản của họ (yêu cầu quét mã QR và xác minh).
 - Nếu được kích hoạt, việc đăng nhập yêu cầu mã TOTP hợp lệ.
 - Người dùng có thể vô hiệu hóa TOTP (yêu cầu xác minh mã TOTP hiện tại).
- **Quản lý Mật khẩu:** Người dùng có thể thay đổi mật khẩu. Việc này yêu cầu mật khẩu cũ hoặc mã TOTP (nếu đã kích hoạt).
- **Đăng xuất:** Người dùng có thể đăng xuất khỏi hệ thống một cách an toàn.

4.2 Quản lý Tập tin (Tải lên)

- **Tải lên Tập tin:** Người dùng có thể tải lên tập tin với các ràng buộc sau:
 - **Định dạng hỗ trợ:** pdf, doc, docx, xls, xlsx, ppt, pptx, txt, csv, jpg, jpeg, png, gif, zip, rar, mp4, mp3.
 - **Giới hạn kích thước tập:** Được thực thi bởi chính sách hệ thống (mặc định 50MB).
- **Cấu hình Bảo mật:**
 - **Bảo vệ bằng Mật khẩu:** Người dùng có thể đặt mật khẩu để truy cập tập tin (độ dài tối thiểu được thực thi bởi chính sách).
 - **Thời gian Hiệu lực:** Người dùng có thể đặt thời gian "Có hiệu lực từ" (Available From) và "Có hiệu lực đến" (Available To). Hệ thống kiểm tra thời gian bắt đầu phải trước thời gian kết thúc.



- **Chia sẻ Riêng tư:** Người dùng có thể giới hạn quyền truy cập cho một danh sách các địa chỉ email cụ thể.
- **Logic Hiển thị:** Tập tin sẽ tự động được đánh dấu là "Riêng tư" nếu bật bảo vệ bằng mật khẩu hoặc chia sẻ với các email cụ thể. Ngược lại, chúng mặc định là "Công khai" (bất kỳ ai có liên kết đều có thể truy cập).

4.3 Bảng điều khiển Người dùng

- **Xem Hồ sơ:** Người dùng có thể xem thông tin hồ sơ của họ.
- **Thống kê Tập tin:** Người dùng có thể xem tóm tắt về các tập tin của họ, bao gồm số lượng tập tin Đang hoạt động (Active), Đang chờ (Pending), Đã hết hạn (Expired) và Đã xóa (Deleted).
- **Danh sách Tập tin:** Người dùng có thể xem danh sách phân trang các tập tin đã tải lên.
 - **Lọc:** Lọc tập tin theo trạng thái (ví dụ: Đang hoạt động, Đã hết hạn).
 - **Sắp xếp:** Sắp xếp tập tin theo ngày tạo hoặc các tiêu chí khác.
- **Thao tác với Tập tin:** Người dùng có thể xóa các tập tin đã tải lên của họ.

4.4 Quản trị

- **Quản lý Chính sách Hệ thống:** Quản trị viên có thể xem và cập nhật các chính sách hệ thống toàn cầu ảnh hưởng đến tất cả người dùng:
 - **maxFileSizeMB:** Kích thước tập tin tối đa cho phép (MB).
 - **minValidityHours:** Thời gian hiệu lực tối thiểu (giờ).
 - **maxValidityDays:** Thời gian hiệu lực tối đa (ngày).
 - **defaultValidityDays:** Thời gian hiệu lực mặc định (ngày).
 - **requirePasswordMinLength:** Độ dài tối thiểu yêu cầu cho mật khẩu tập tin.
- **Dọn dẹp Hệ thống:** Quản trị viên có thể kích hoạt quy trình dọn dẹp để xóa vĩnh viễn các tập tin đã hết hạn hoặc đã xóa mềm khỏi hệ thống.

5 Non-functional Requirements

5.1 Hiệu năng

- **Thời gian tải trang:** Các trang chính (Login, Dashboard, Upload, Download) tải trong vòng ≤ 2 giây trong điều kiện mạng ổn định.
- **Phản hồi giao diện:** Các thao tác UI như mở modal, chuyển trang nội bộ, và hiển thị toast phải phản hồi **gần tức thời** (thường dưới 200–300 ms).
- **Upload tệp:** Khi người dùng chọn tệp, tiến trình upload phải hiển thị progress bar trong vòng < 0.5 giây để tạo cảm giác phản hồi nhanh.

5.2 Tính dễ sử dụng

- **Rõ ràng:** Trạng thái file (Active, Pending, Expired) phải được phân biệt bằng màu sắc, icon nhất quán, trực quan.
- **Nhất quán:** Các toast thông báo (success / warning / error) phải sử dụng chung một style và thời gian hiển thị (mặc định 3-5 giây).

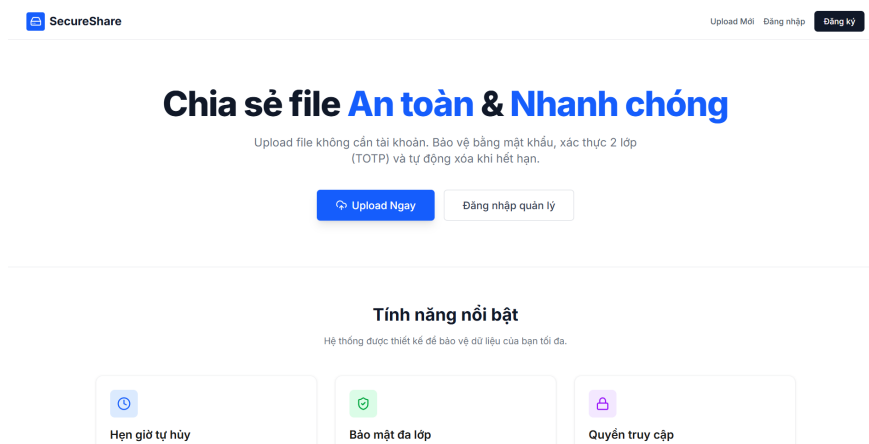
5.3 Khả năng bảo trì

- **Hệ thống mã nguồn rõ ràng:** Mã nguồn được chia theo module (auth, dashboard, upload, admin).
- **Tái sử dụng:** Các UI component phải được gom vào thư mục components và được tái sử dụng thay vì viết lại.
- **Dễ mở rộng:** Hệ thống phải được thiết kế để dễ dàng bổ sung thêm kiểu bảo mật, loại file hoặc chính sách mới.

6 UI Design & screen flow

6.1 Screen Flow

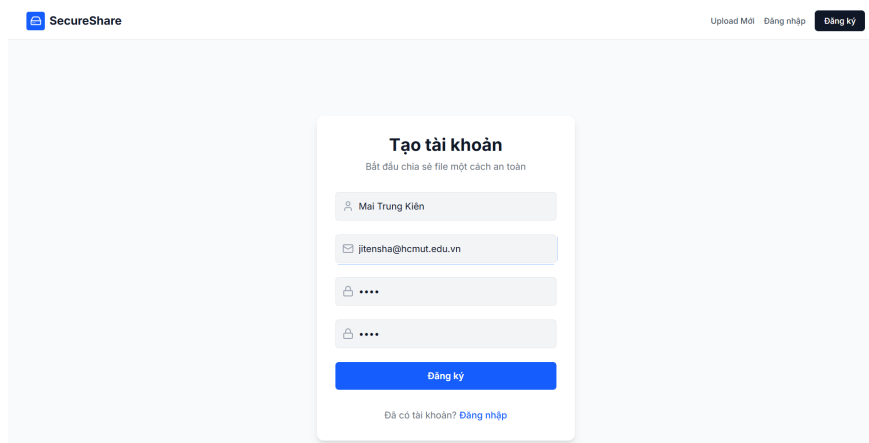
6.1.1 Public



Home Page

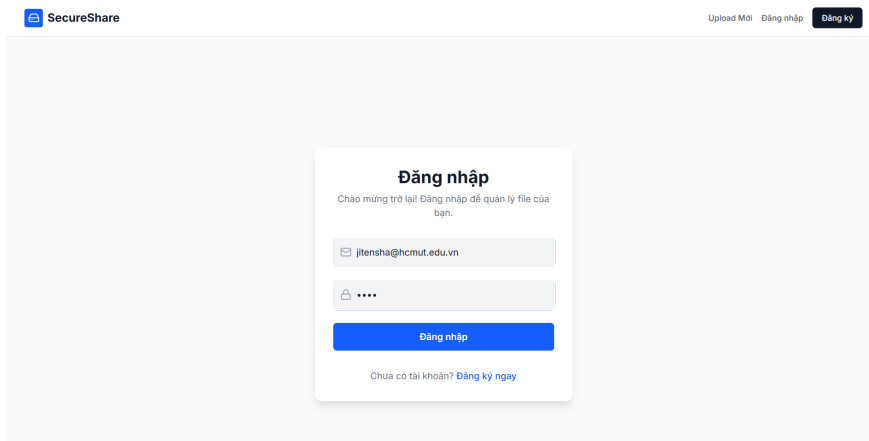
- Route: /
- File: app/page.tsx

6.1.2 Authentication Screens



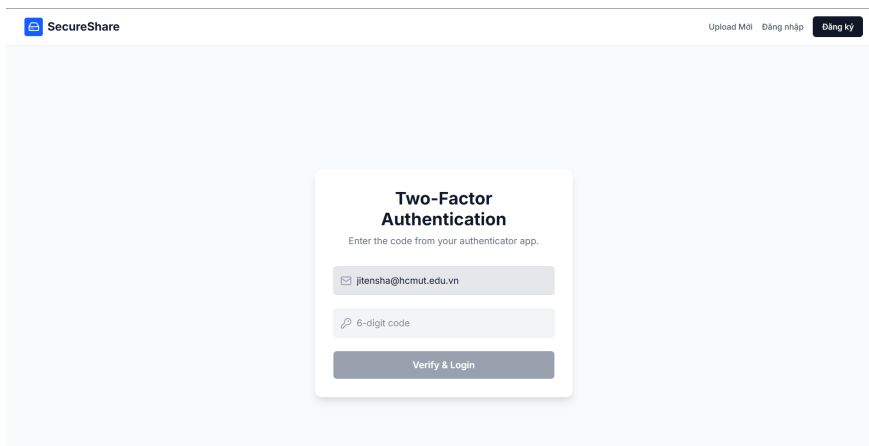
Đăng ký

- Route: /register
- File: app/(auth)/register/page.tsx



Đăng nhập

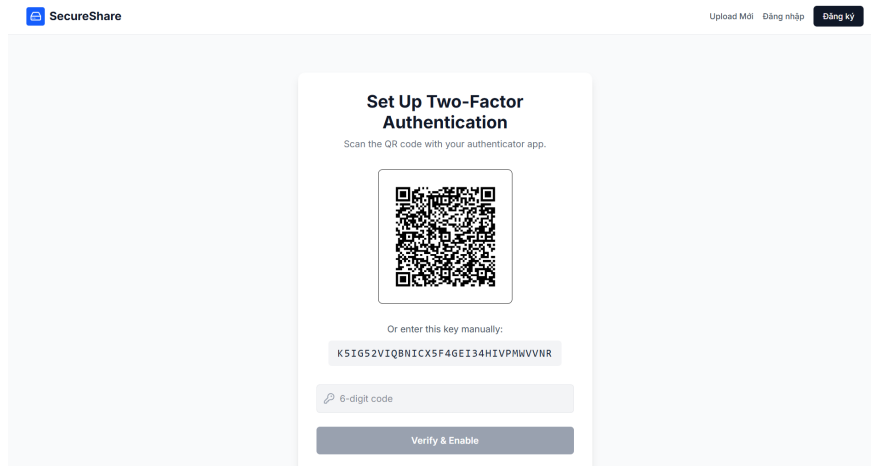
- Route: /login
- File: app/(auth)/login/page.tsx



Đăng nhập với TOTP

- Route: /login/totp

- File: `app/(auth)/login/totp/page.tsx`

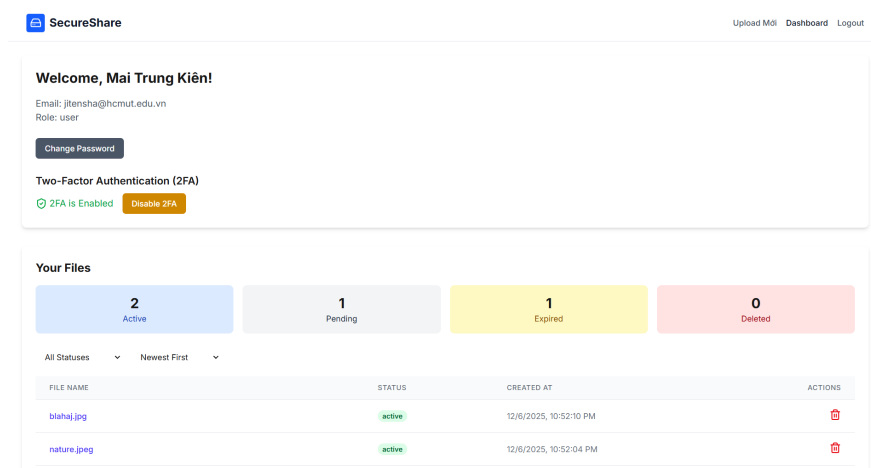


Thiết lập TOTP

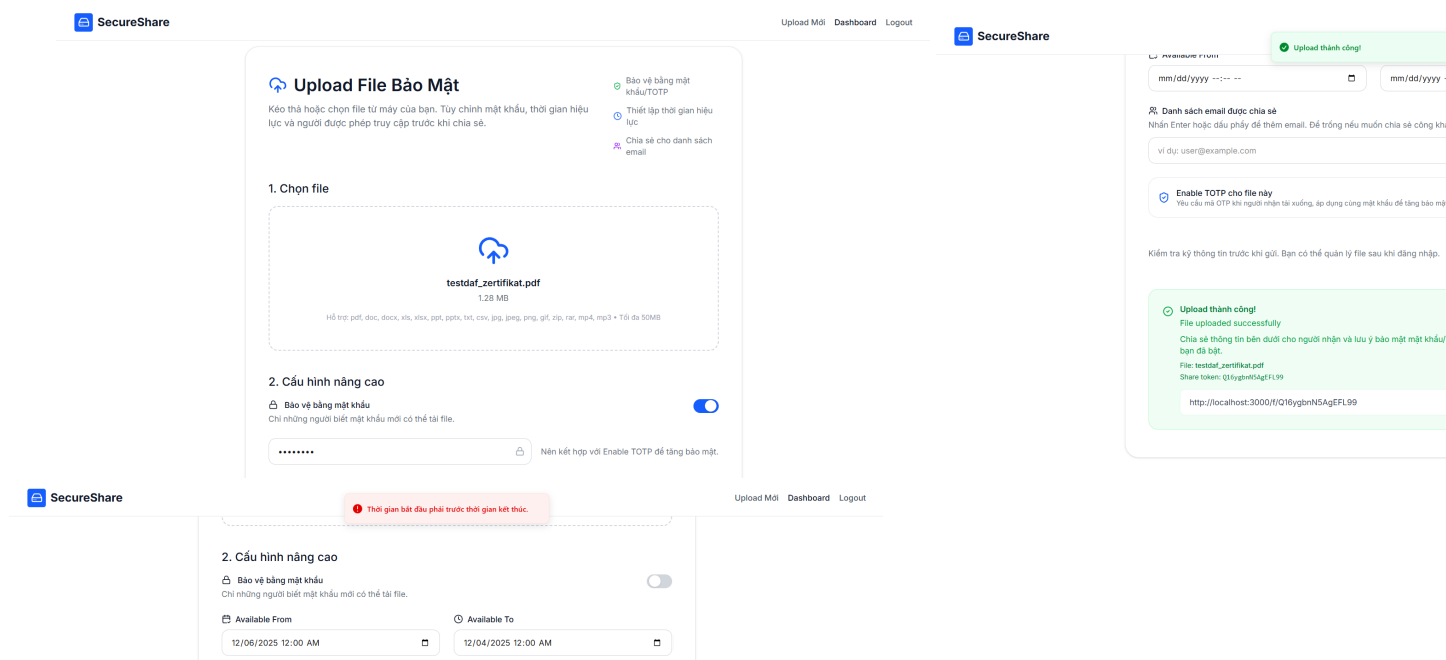
- Route: `/totp-setup`
- File: `app/(auth)/totp-setup/page.tsx`

6.1.3 User Screens

Bảng điều khiển



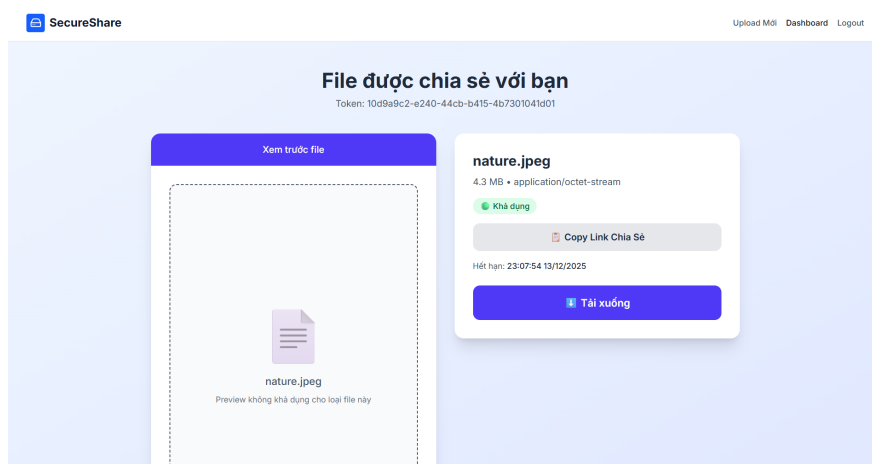
- Route: `/dashboard`
- File: `app/dashboard/page.tsx`



Upload

- Route: /upload
- File: app/upload/page.tsx

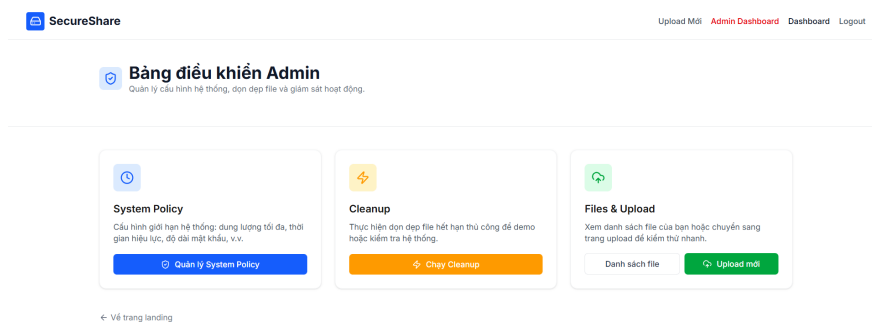
6.1.4 Public File Access



Trang truy cập công khai

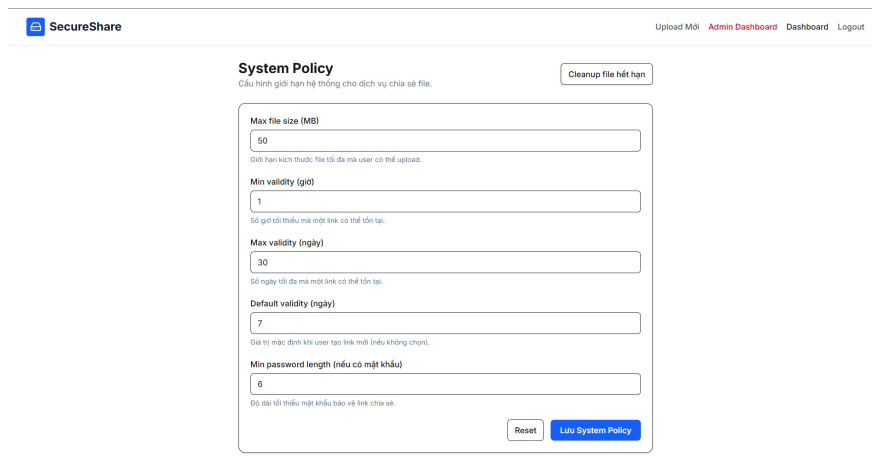
- Route: `/f/[token]`
- File: `app/(public)/f/[token]/page.tsx`

6.1.5 Admin Screens



Trang chính Admin

- Route: `/admin`
- File: `app/admin/page.tsx`



Admin Policy

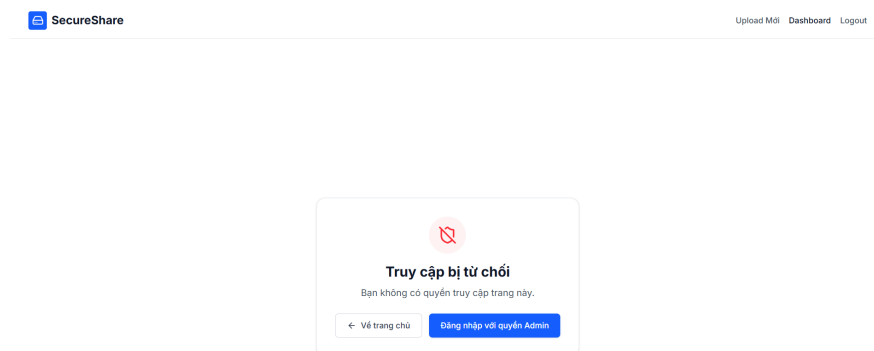
- Route: `/admin/policy`

- File: app/admin/policy/page.tsx



Admin Cleanup

- Route: /admin/cleanup
- File: app/admin/cleanup/page.tsx



Not Admin

- Route: /admin/templates/notadmin
- File: app/admin/templates/notadmin.tsx