**BỘ GIÁO DỤC VÀ ĐÀO TẠO**

**TRƯỜNG ĐẠI HỌC KINH TẾ - TÀI CHÍNH**

**THÀNH PHỐ HỒ CHÍ MINH**

---------------------------



**BÁO CÁO CUỐI KỲ**

**MÔN HỌC: PHÁT TRIỂN WEB NÂNG CAO**

**CHUYÊN NGÀNH: CÔNG NGHỆ PHẦN MỀM**

**ĐỀ TÀI: WEB DOCUMENT AND FILE SHARING**

**NHÓM 8:**

ĐOÀN NGUYỄN HUY 225052312 251.ITE1278E.B01E

LÊ TRẦN MỸ TÂM 225050282 251.ITE1278E.B01E

MAI ĐOÀN GIA HUY 225052198 251.ITE1278E.B01E

HUỲNH VĨNH SAN 225054102 251.ITE1278E.B01E

GIÁO VIÊN HƯỚNG DẪN: HOÀNG VĂN HIẾU

THÀNH PHỐ HỒ CHÍ MINH, THÁNG 12 NĂM 2025

**BỘ GIÁO DỤC VÀ ĐÀO TẠO**

**TRƯỜNG ĐẠI HỌC KINH TẾ - TÀI CHÍNH**

**THÀNH PHỐ HỒ CHÍ MINH**

---------------------------



**BÁO CÁO CUỐI KỲ**

**MÔN HỌC: PHÁT TRIỂN WEB NÂNG CAO**

**CHUYÊN NGÀNH: CÔNG NGHỆ PHẦN MỀM**

**ĐỀ TÀI: WEB DOCUMENT AND FILE SHARING**

**NHÓM 8:**

ĐOÀN NGUYỄN HUY 225052312 251.ITE1278E.B01E

LÊ TRẦN MỸ TÂM 225050282 251.ITE1278E.B01E

MAI ĐOÀN GIA HUY 225052198 251.ITE1278E.B01E

HUỲNH VĨNH SAN 225054102 251.ITE1278E.B01E

Giáo Viên Hướng Dẫn: HOÀNG VĂN HIẾU

Ho Chi Minh City, December 2025

Mục Lục

[I. Introduction 5](#_Toc584)

[1. Background & Problem Statement 5](#_Toc8413)

[2. Objectives 5](#_Toc23426)

[3. Scope & Limitations 5](#_Toc14804)

[4. Project Significance 5](#_Toc16336)

[II. Literature Review & Requirements 5](#_Toc10229)

[1. Technology Used 5](#_Toc10859)

[2. Requirement Analysis 6](#_Toc14022)

[3. Use-case Diagram 6](#_Toc18735)

[III. System Design 6](#_Toc21368)

[1. Database Design 6](#_Toc30784)

[2. UI / UX Design 6](#_Toc26222)

[3. System Architecture 6](#_Toc20480)

[IV. Implementation & Result 6](#_Toc23014)

[1. Development & Environment Tools 6](#_Toc32656)

[2. Step-By-Step Installation Guide 6](#_Toc16111)

[3. Key Code Snippets & Explanation 6](#_Toc16492)

[4. Screenshots Of Main Features 6](#_Toc19221)

[5. Conclusion & Future Work 6](#_Toc21482)

[V. References 6](#_Toc31851)

[VI. Appendix 6](#_Toc16798)

[1. Github Link 6](#_Toc24276)

[2. Database Dump (.sql) 6](#_Toc1522)

[3. .evn.example 6](#_Toc1222)

[IMAGE 1: USE CASE DIAGRAM 1 10](#_Toc194579091)

[IMAGE 2: USE CASE DIAGRAM 2 10](#_Toc194579092)

[IMAGE 3: CLASS DIAGRAM 13](#_Toc194579093)

[IMAGE 4: SEQUENCE DIAGRAM (ADMIN AND USER) 17](#_Toc194579094)

[IMAGE 5: SEQUENCE DIAGRAM ( CUSTOMER) 19](#_Toc194579095)

[IMAGE 6: SEQUENCE DIAGRAM ( ADMIN) 21](#_Toc194579096)

[IMAGE 7: SEQUENCE DIAGRAM 22](#_Toc194579097)

[IMAGE 8: STATECHART DIAGRAM 23](#_Toc194579098)

[IMAGE 9: ACTIVITY DIAGRAM (ADMIN) 27](#_Toc194579099)

[IMAGE 10: ACTIVITY DIAGRAM (CUSTOMER) 29](#_Toc194579100)

[IMAGE 11: ACTIVITY DIAGRAM (ADMIN AND CUSTOMER) 32](#_Toc194579101)

**TÓM TẮT**

Trong bối cảnh chuyển đổi số và làm việc trực tuyến ngày càng phát triển mạnh mẽ, nhu cầu lưu trữ, quản lý và chia sẻ tài liệu số một cách an toàn, hiệu quả đã trở thành yêu cầu thiết yếu đối với cá nhân, doanh nghiệp và các tổ chức giáo dục. Tuy nhiên, thực tế cho thấy nhiều đơn vị hiện nay vẫn đang sử dụng các phương thức quản lý tài liệu rời rạc, thiếu tính đồng bộ, phụ thuộc vào các công cụ thủ công hoặc nền tảng miễn phí riêng lẻ, dẫn đến nhiều hạn chế như: khó kiểm soát quyền truy cập, nguy cơ rò rỉ dữ liệu, thiếu khả năng theo dõi phiên bản tài liệu và không hỗ trợ tốt cho làm việc nhóm.

Xuất phát từ yêu cầu thực tiễn đó, đồ án **“Hệ thống Quản lý và Chia sẻ Tài liệu Trực tuyến (Web Document and File Sharing)”** được nghiên cứu và thực hiện với mục tiêu phát triển một nền tảng web tập trung, cho phép người dùng lưu trữ, quản lý, chia sẻ và cộng tác trên tài liệu một cách thuận tiện và bảo mật. Hệ thống được xây dựng dựa trên các công nghệ web hiện đại, sử dụng kiến trúc **ASP.NET Core MVC** cho phía máy chủ nhằm đảm bảo tính ổn định, khả năng mở rộng và xử lý hiệu quả nhiều người dùng đồng thời, kết hợp với **SQL Server** và **Entity Framework** để quản lý dữ liệu chặt chẽ và nhất quán.

Về phía người dùng, giao diện hệ thống được thiết kế theo hướng thân thiện và tối ưu trải nghiệm (UI/UX), hỗ trợ truy cập đa thiết bị, cho phép người dùng dễ dàng tải lên, tải xuống, phân loại tài liệu theo thư mục, cũng như chia sẻ tệp với các mức phân quyền khác nhau. Đồng thời, hệ thống cung cấp cho quản trị viên (Admin) các chức năng quản lý người dùng, kiểm soát quyền truy cập, theo dõi lịch sử thao tác và giám sát dung lượng lưu trữ thông qua các công cụ báo cáo trực quan.

Kết quả đạt được của đồ án không chỉ là một ứng dụng web hoàn chỉnh về mặt chức năng mà còn thể hiện tính khả thi trong việc ứng dụng công nghệ thông tin vào bài toán quản lý và chia sẻ tài liệu số, góp phần nâng cao hiệu quả làm việc, tăng cường bảo mật dữ liệu và hỗ trợ tốt hơn cho mô hình làm việc cộng tác trong môi trường số hiện đại.

# GIỚI THIỆU

## Bối Cảnh & Thực Trạng Vấn Đề

1. **Bối Cảnh:** Sự phát triển nhanh chóng của nội dung số và các hình thức làm việc, cộng tác trực tuyến đã làm gia tăng đáng kể nhu cầu về các hệ thống quản lý tài liệu và chia sẻ tệp tin hiệu quả. Tuy nhiên, nhiều giải pháp hiện nay chủ yếu hướng đến quy mô doanh nghiệp lớn hoặc phụ thuộc nhiều vào các dịch vụ đám mây của bên thứ ba. Điều này dẫn đến những hạn chế nhất định về khả năng tùy biến, quyền kiểm soát dữ liệu và quản lý quyền truy cập.

Nhằm đáp ứng nhu cầu về một giải pháp gọn nhẹ, dễ tùy chỉnh nhưng vẫn đảm bảo tính bảo mật, kiểm soát truy cập và dễ sử dụng, dự án này ứng dụng các công nghệ web hiện đại như **ASP.NET Core** và **Entity Framework Core** để thiết kế và xây dựng một hệ thống quản lý và chia sẻ tài liệu dựa trên nền tảng web, phù hợp cho các môi trường có yêu cầu kiểm soát người dùng và quản lý tệp tin hiệu quả.

1. **Thực Trạng Vấn Đề:** Vấn đề trọng tâm hiện nay là việc tích hợp **Google Drive** vào hệ thống đòi hỏi phải trải qua nhiều bước xác thực phức tạp. Trong khi đó, **Firebase** phát sinh thêm chi phí nhưng lại cung cấp dung lượng lưu trữ hạn chế. Do đó, các tệp tin hiện đang được lưu trữ trực tiếp trên ổ cứng máy chủ của quản trị viên, khiến hệ thống không thể tận dụng các lợi ích của điện toán đám mây như sao lưu tự động, tính sẵn sàng cao hay khả năng mở rộng linh hoạt về dung lượng lưu trữ.

Khi số lượng người dùng và tệp tin tải lên ngày càng tăng, dung lượng lưu trữ có thể trở nên hạn chế và việc quản lý khối lượng dữ liệu lớn sẽ gặp nhiều khó khăn. Bên cạnh đó, độ tin cậy của hệ thống cũng có thể bị ảnh hưởng trong trường hợp xảy ra sự cố phần cứng hoặc khi máy chủ không được duy trì hoạt động liên tục.

## Mục Tiêu Đề Tài

Mục tiêu chính yếu là thiết kế và phát triển một hệ thống lưu trữ và chia sẻ tài liệu trên nền tảng web có thể cho phép người dùng đăng tải, quản lý, và truy cập các tệp tin của mình một cách an toàn thông qua môi trường trực tuyến. Hệ thống hướng đến việc cung cấp các chức năng quản lý tệp tin cơ bản nhưng thiết yếu, đồng thời đám bảo tính đơn giảnvà dễ sử dụng.

## Phạm Vi và Hạn Chế

1. **Phạm Vi:** Dự án này tập trung vào sự phát triển của hệ thống lưu trữ và chia sẻ tài liệu trên nền tảng web dành cho môi trường quy mô nhỏ hoặc triển khai tự lưu trữ. Hệ thốn cho phép người dùng đã đăng ký tải lên, tải xuống và quản lý các tệp tin cá nhân thông qua giao diện web.
2. **Hạn Chế:**

* Trước hết, hệ thống hiện chưa được tích hợp với các dịch vụ lưu trữ đám mây; toàn bộ tệp tin được lưu trữ trên ổ cứng cục bộ của máy chủ. Điều này dẫn đến hạn chế về khả năng mở rộng dung lượng lưu trữ và không hỗ trợ các tính năng như sao lưu tự động hay cơ chế dự phòng dữ liệu.
* Kế đến, hệ thống có thể gặp các vấn đề về hiệu năng và độ tin cậy khi số lượng người dùng hoặc số lượng tệp tin tải lên tăng lên đáng kể. Do ứng dụng phụ thuộc vào một máy chủ duy nhất, các sự cố phần cứng hoặc thời gian ngừng hoạt động của máy chủ có thể ảnh hưởng trực tiếp đến khả năng truy cập tệp tin.
* **Ngoài ra**, các chức năng bảo mật hiện mới được triển khai ở mức cơ bản và có thể cần được tiếp tục cải tiến, nâng cao trong các giai đoạn phát triển tiếp theo.

## Project Significance

Hệ thống đáp ứng nhu cầu về một giải pháp đơn giản và có khả năng tùy chỉnh để quản lý và chia sẻ tệp tin trong các môi trường quy mô nhỏ hoặc triển khai tự lưu trữ. Thông qua việc lưu trữ tệp tin cục bộ trên máy chủ, dự án giúp giảm sự phụ thuộc vào các dịch vụ đám mây của bên thứ ba, đồng thời mang lại cho quản trị viên khả năng kiểm soát tốt hơn đối với việc lưu trữ và truy cập dữ liệu.

# Tổng Quan Nghiên Cứu & Yêu Cầu Hệ Thống

## Công Nghệ Đã Sử Dụng

* + - Framework: ASP.NET CORE 8.0
    - Kiến trúc: MVC combined with Razor Pages.
    - Ngôn ngữ lập trình: C#.
    - Cơ sở dữ liệu: SQL Server & EF Core Tools.
    - Công cụ phát triển:
      * Visual Studio 2022
      * NuGet Package Manager
      * ChatGPT
      * Gemini
      * Xác thực Email
      * GitHub Copilot
      * Cổng thanh toán Paypal

## Phân Tích Yêu Cầu

**Yêu cầu chức năng**

* Người dùng phải có khả năng: Đăng nhập, đăng xuất, xem, tải xuống, xóa.
* Quyền truy cập tệp bị giới hạn theo quyền sở hữu (chỉ chủ sở hữu có quyền truy cập và thao tác trên các tệp của mình).
* Cần có tài khoản quản trị viên để quán lý vai trò người dùng và giám sát việc sử dụng hệ thống.

**Yêu cầu phi chức năng**

* Đảm bảo các biện pháp bảo mật cơ bản: bảo vệ tài khoản người dùng và hạn chế quyền truy cập vào tệp.
* Áp dụng xác thực và phân quyền người dùng để ngăn chặn truy cập trái phép
* Giao diện người dùng rõ ràng, đơn giản nhằm mang lại trải nghiệm sử dụng dễ dàng.
* Hệ thống cần đảm bảo thờ gian phản hồi chấp nhận được khi xử lý các thao tác tải lên, tải xuống với số lượng người dùng hạn chế.
* Dễ bảo trì và linh hoạt, cho phép cải tiến trong tương lai như tích hợp lưu trữ đám mây hoặc tăng cường các tính năng bảo mật.

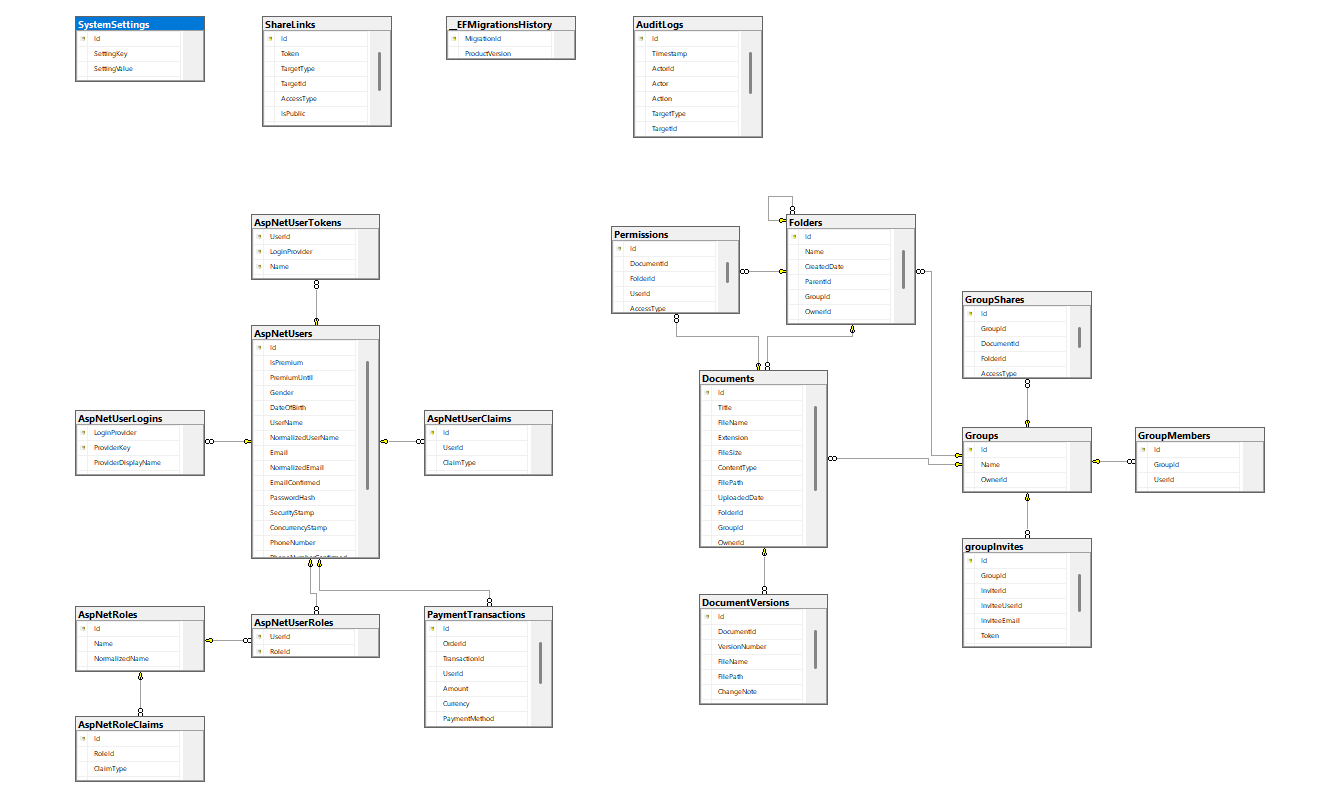
## Use-case Diagram

Hình 1: Usecase Diagram

The use case diagram illustrates the interactions between two main actors, Admin and User. The Admin is responsible for system management and monitoring, while the User can register, authenticate, manage folders and documents, share files, and access shared documents.

# Thiết Kế Hệ Thống

## Database Design



Hình 2: ERD Diagram

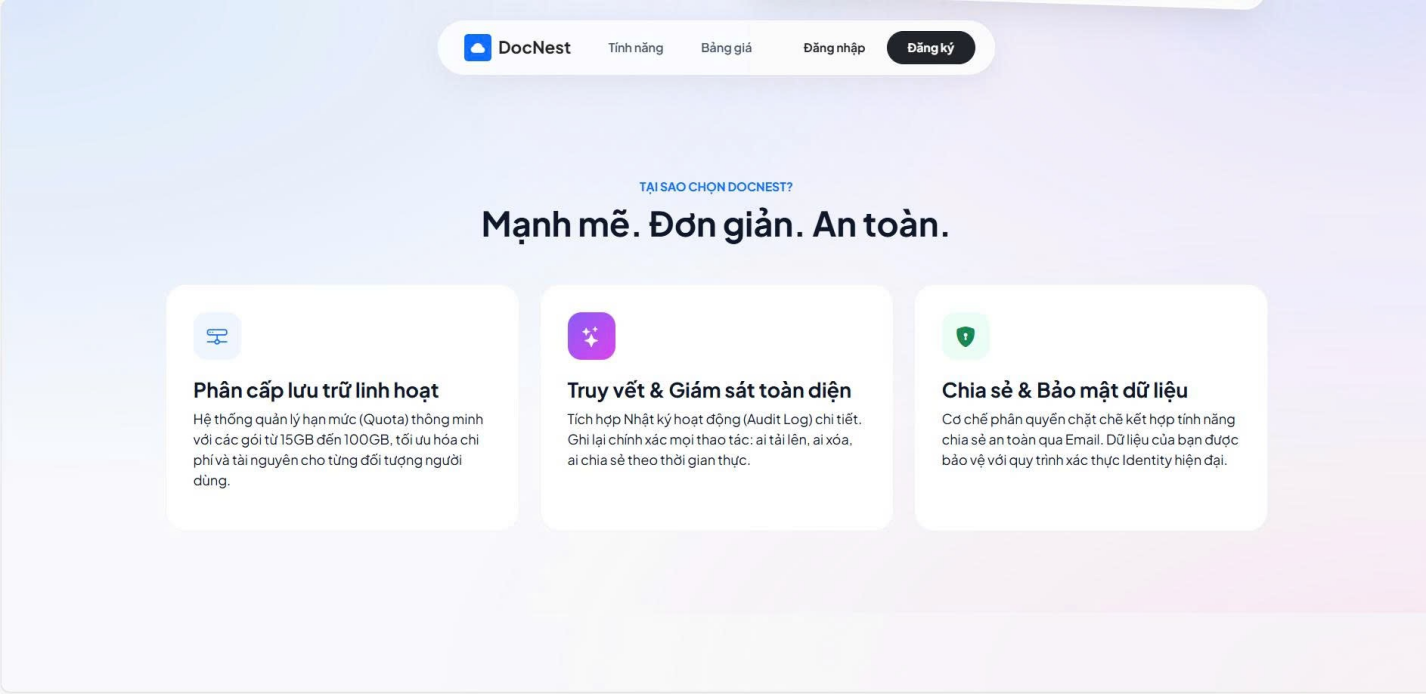
The Entity Relationship Diagram (ERD) represents the database structure of the document and file sharing system. The core entities include User**, Documents**, **Folders**, and **Permissions**, which together support file storage, organization, and access control. The **Documents** entity stores information related to uploaded files, such as file name, size, type, path, upload date, and ownership. Each document can belong to a specific folder and may have multiple versions, which are managed through the **DocumentVersions** entity.

The **Folders** entity is used to organize documents hierarchically and supports parent–child relationships, allowing users to create nested folders. Ownership information is stored to ensure that folders are associated with specific users. Access control is handled by the **Permissions** entity, which defines user access rights to documents or folders, supporting secure file sharing between users.

User authentication and authorization are managed through **ASP.NET Core Identity** tables, including **AspNetUsers**, **AspNetRoles**, and related mapping tables. These entities handle user accounts, roles, claims, and login information. In addition, the system includes supporting entities such as **AuditLogs** for tracking system activities and **SystemSettings** for storing application configuration data. Overall, the ERD illustrates how data is structured to support secure file management, user control, and system monitoring.

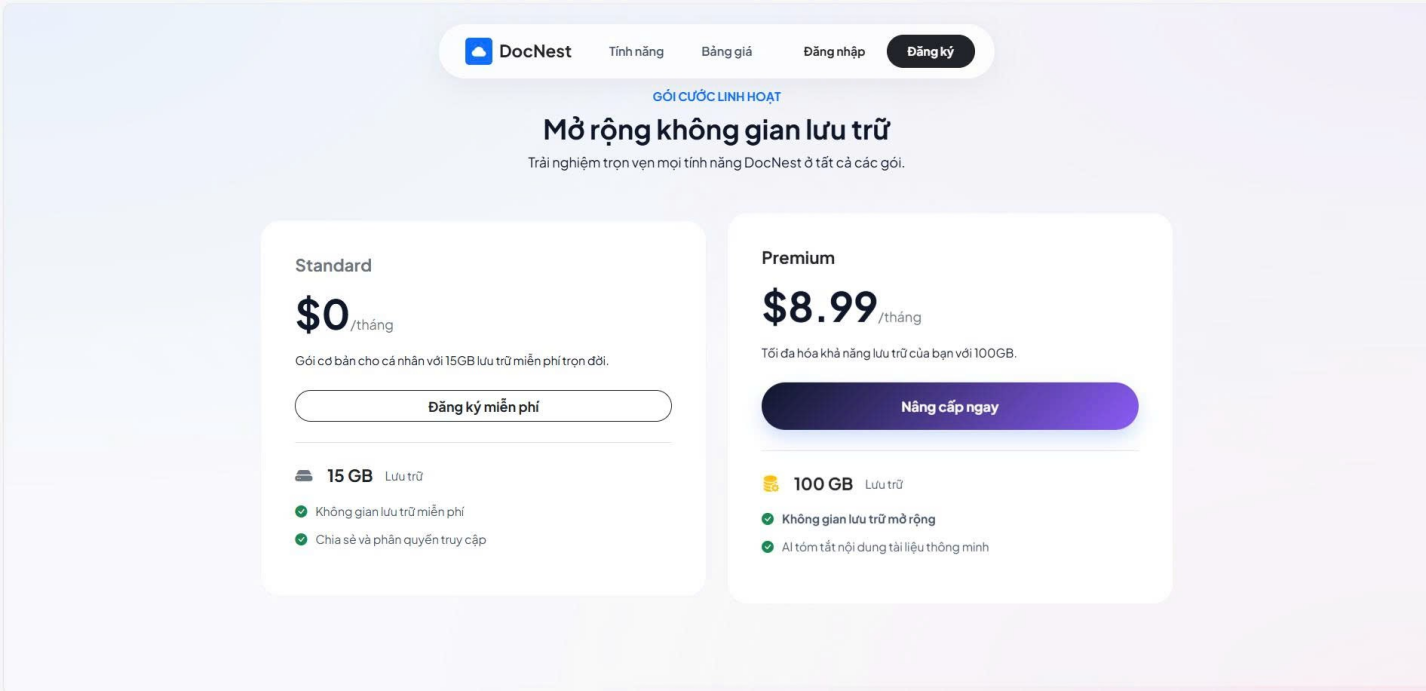
## UI / UX Design





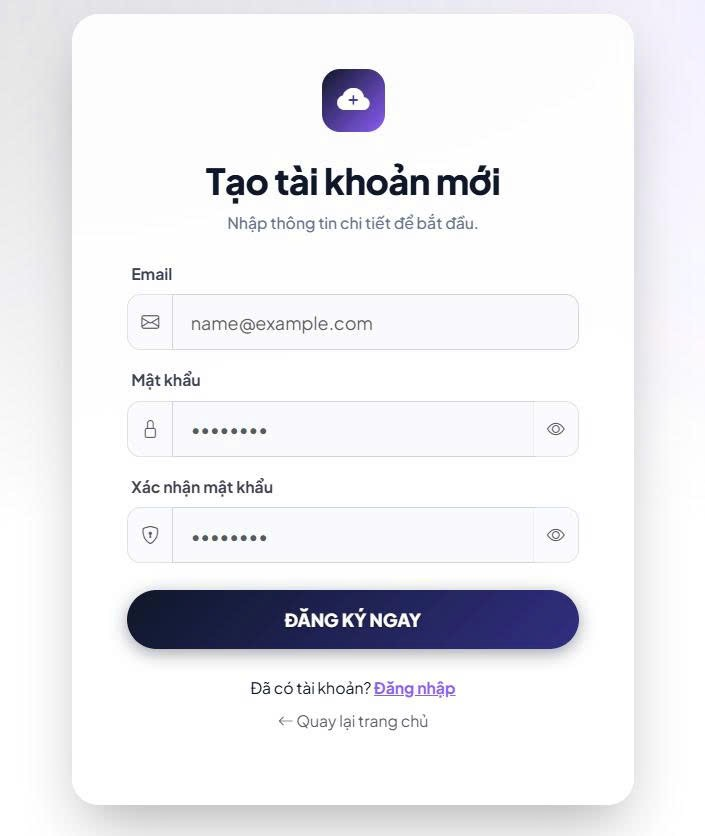
Hình 3: Landing page

Trang giao diện cho người chưa đăng nhập.



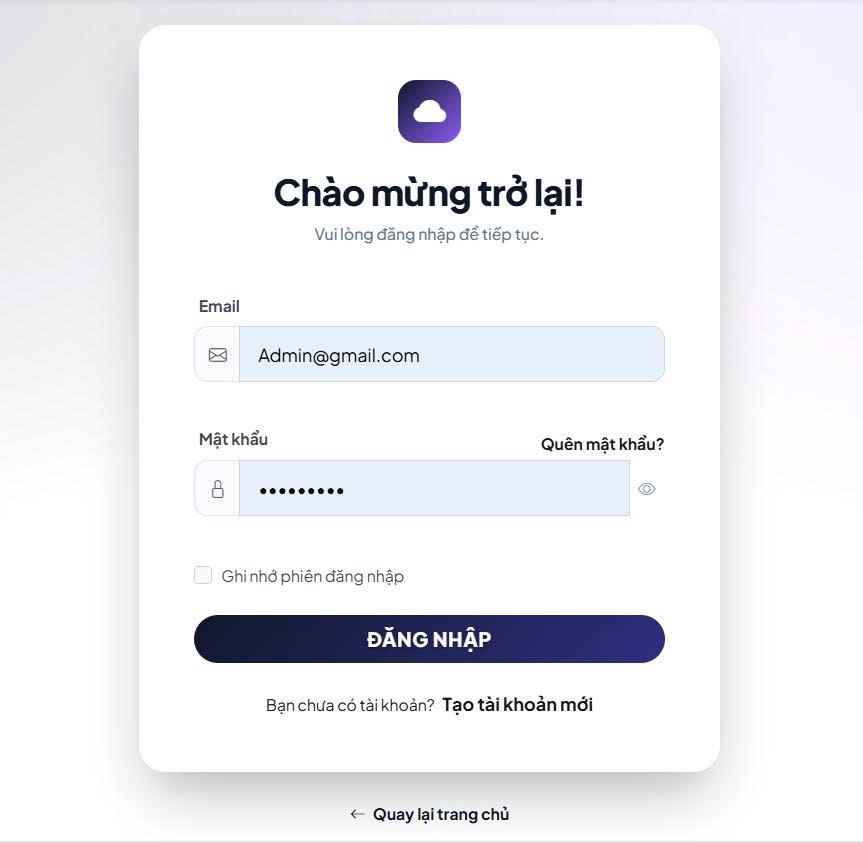
Hình 3: Nâng cấp tài khoản

Giao diện mua gói Premium.



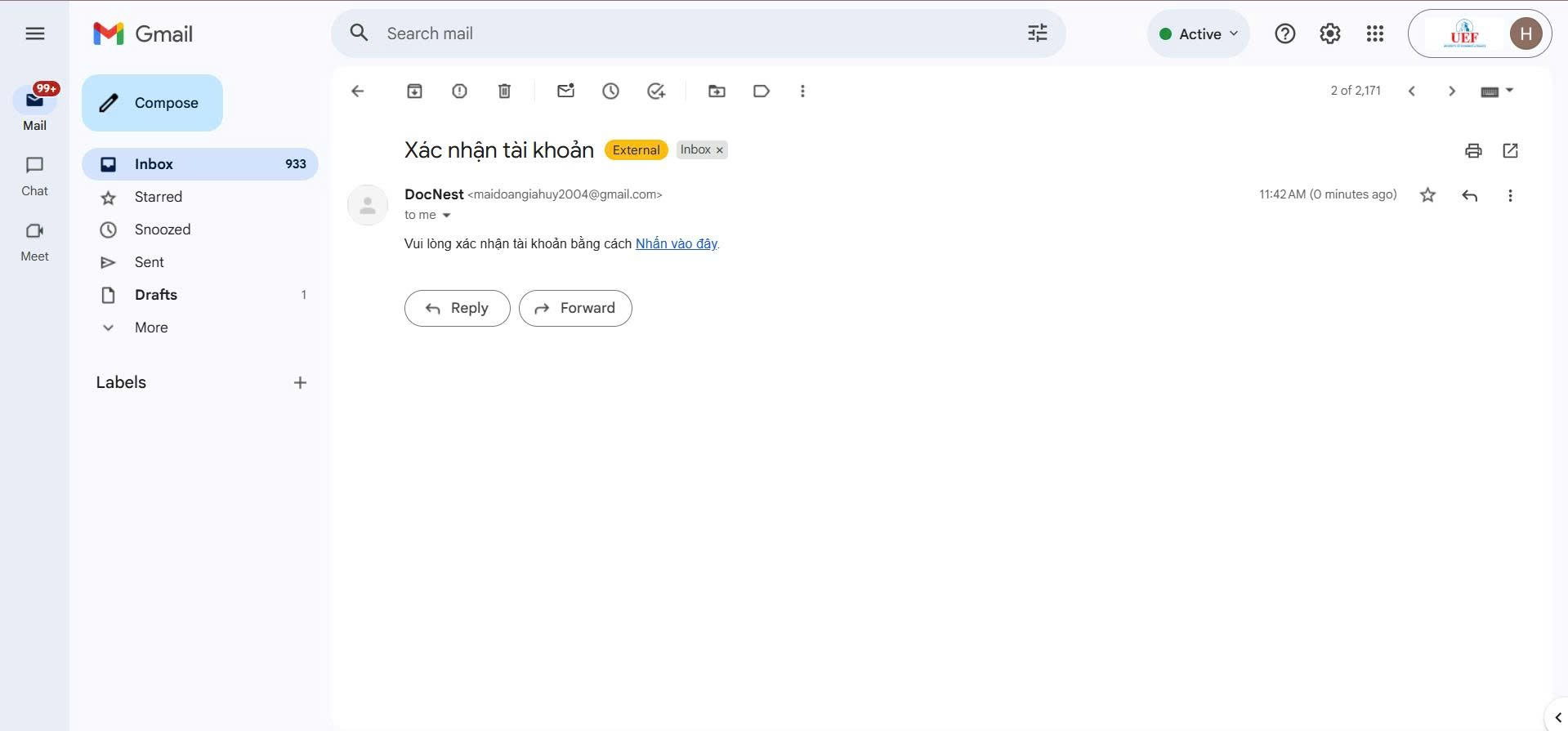
*Hình 4: Đăng ký*

Trang đăng ký tài khoản



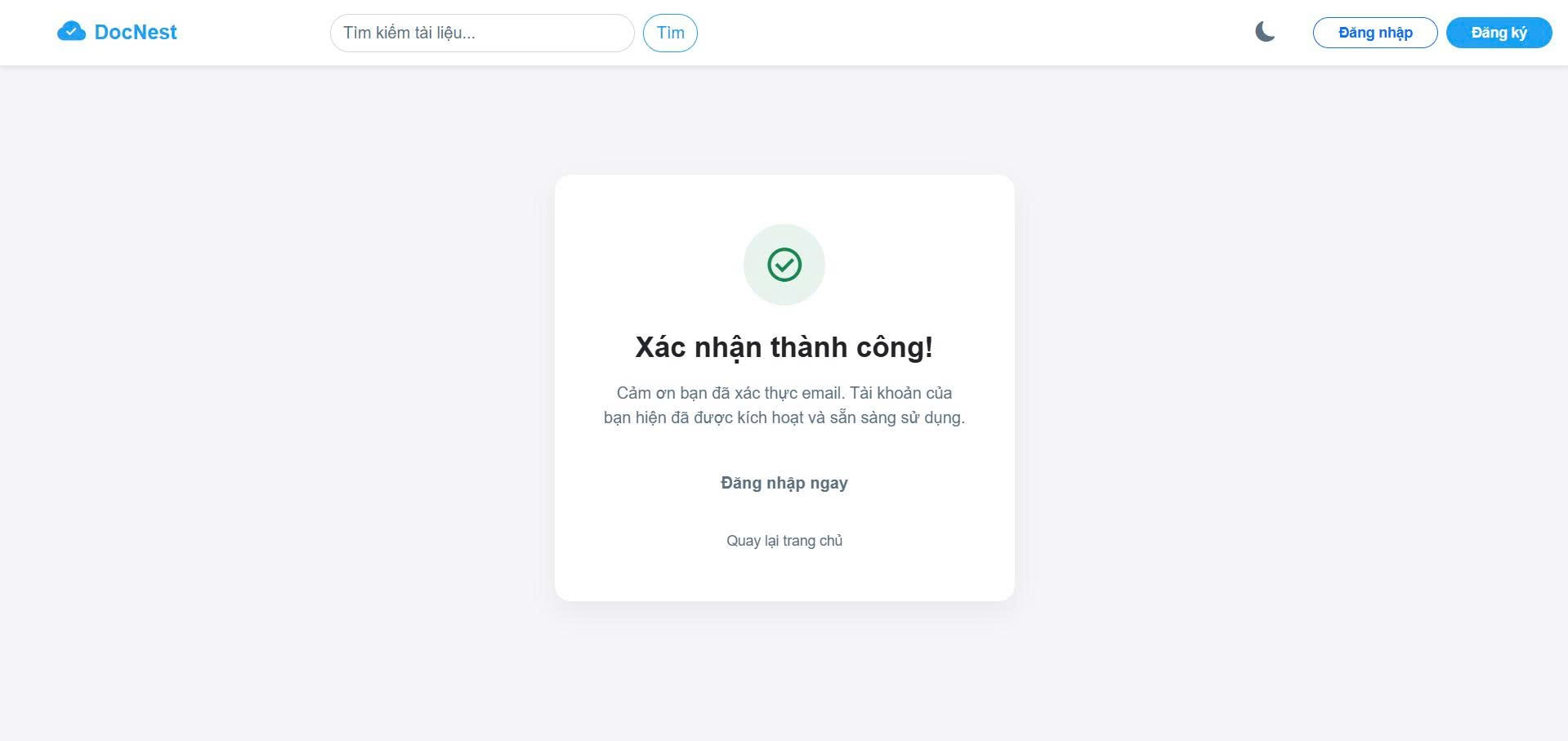
*Hình 5: Đăng nhập*

Giao diện đăng nhập tài khoản đã đăng ký



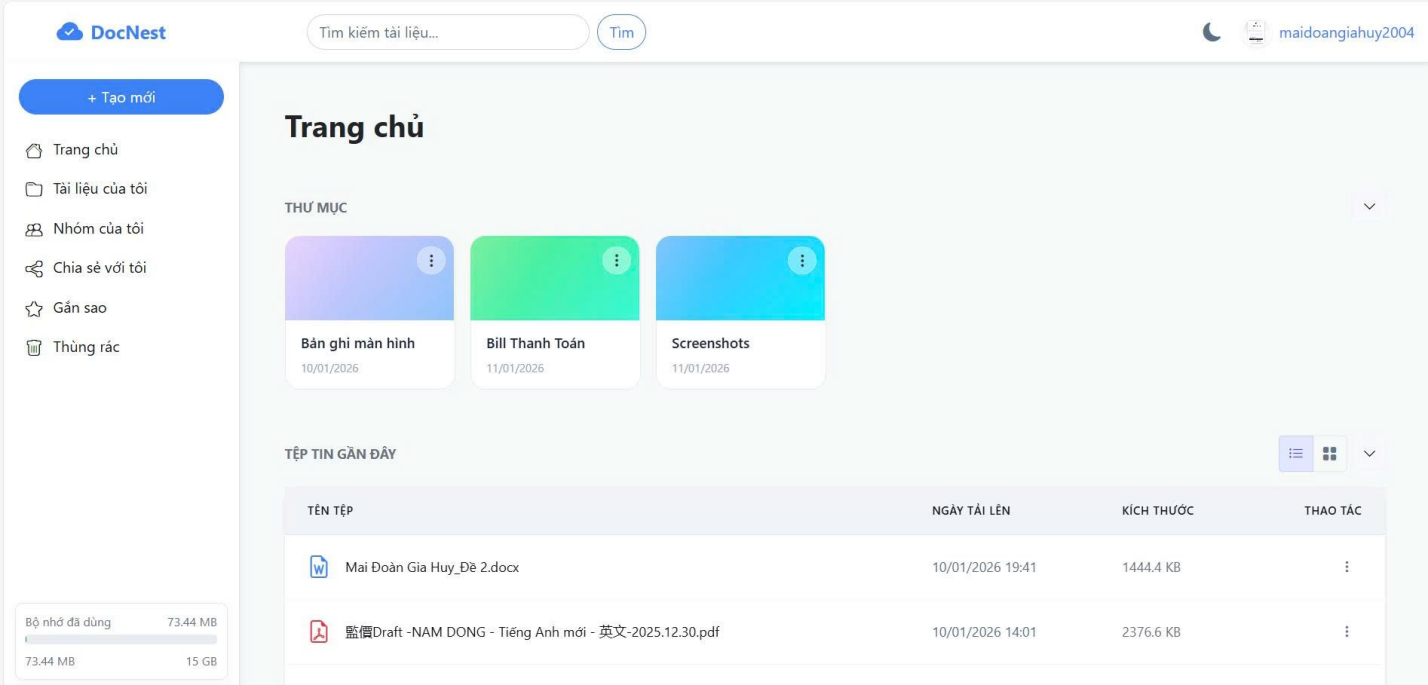
Hình N: Xác nhận tài khoản

Hệ thống sẽ gửi thư xác thực đến tài khoản email mà người dùng sử dụng để đăng ký tài khoản.



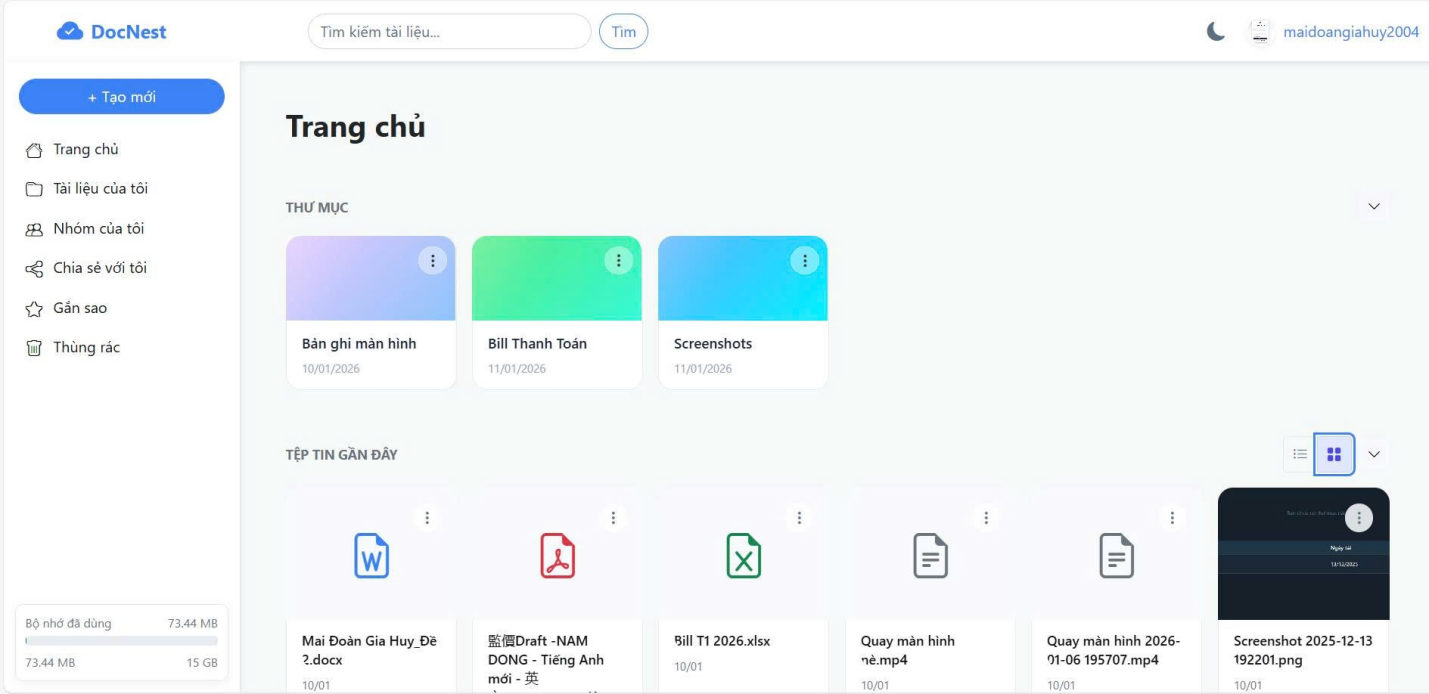
Hình N: Xác thực thành công

Người dùng click vào nút “Nhấn vào đây” để xác thực tài khoản. Hệ thống sẽ hiển thị giao diện thông báo xác nhận thành công cho người dùng.



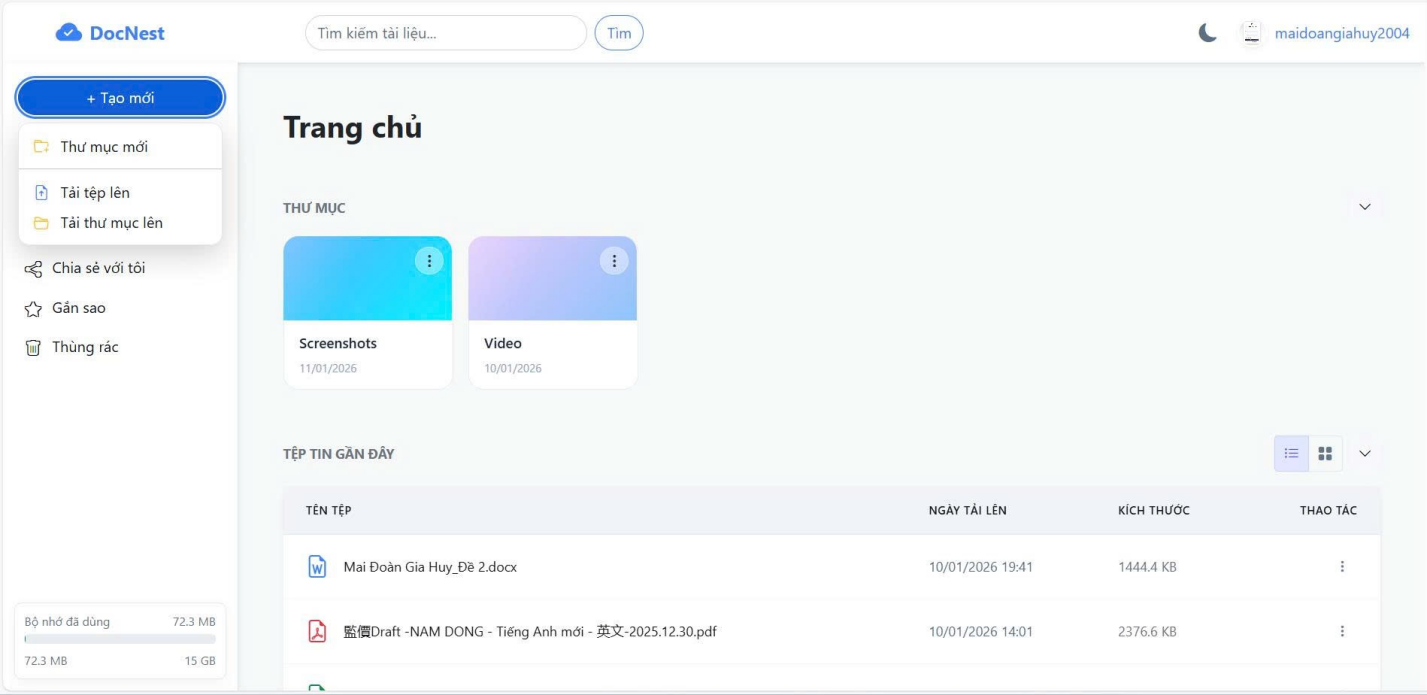
*Hình 4: User Dashboard*

Trang chủ của người dùng đã đăng nhập. Từ đây có thể xem tổng quát các file, tài liệu và các chức năng khác.



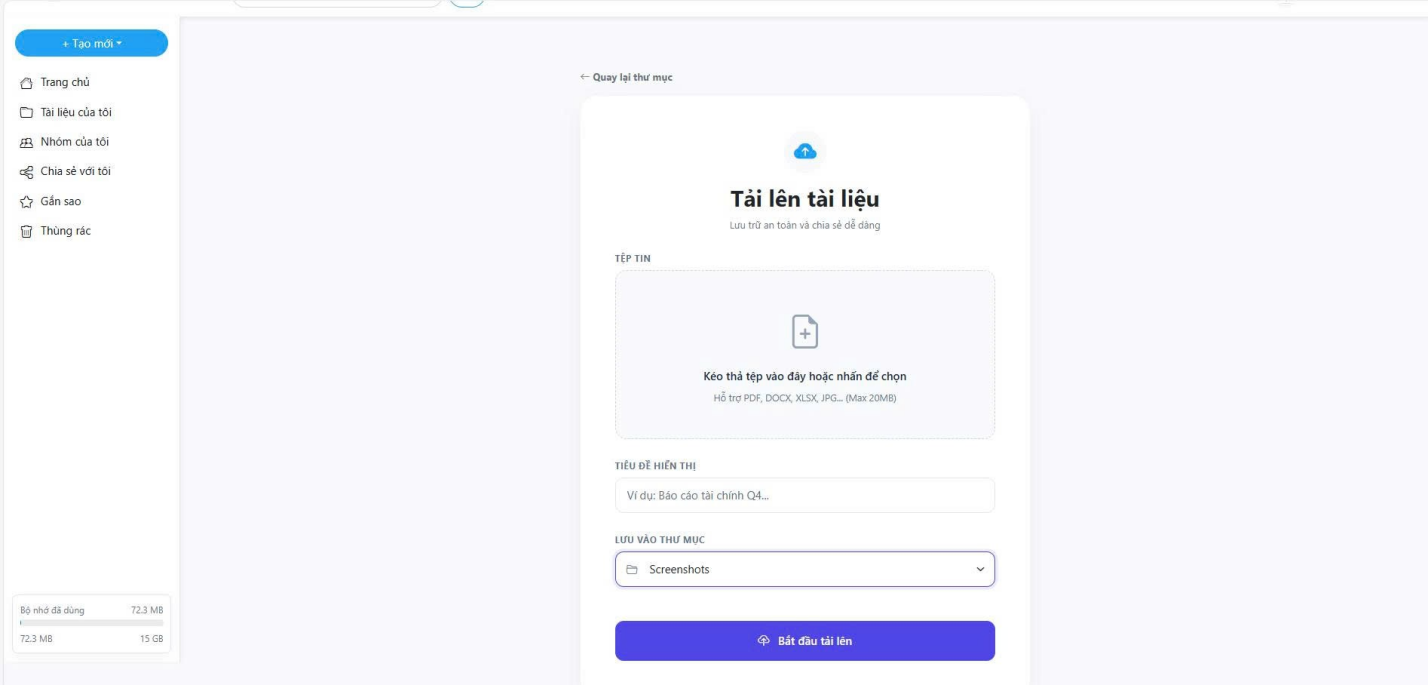
Hình 5: Giao diện người dùng (xem dạng lưới)

Người dùng được tùy chỉnh chế độ hiển thị của giao diện giữa 2 dạng: Danh Sách & Lưới



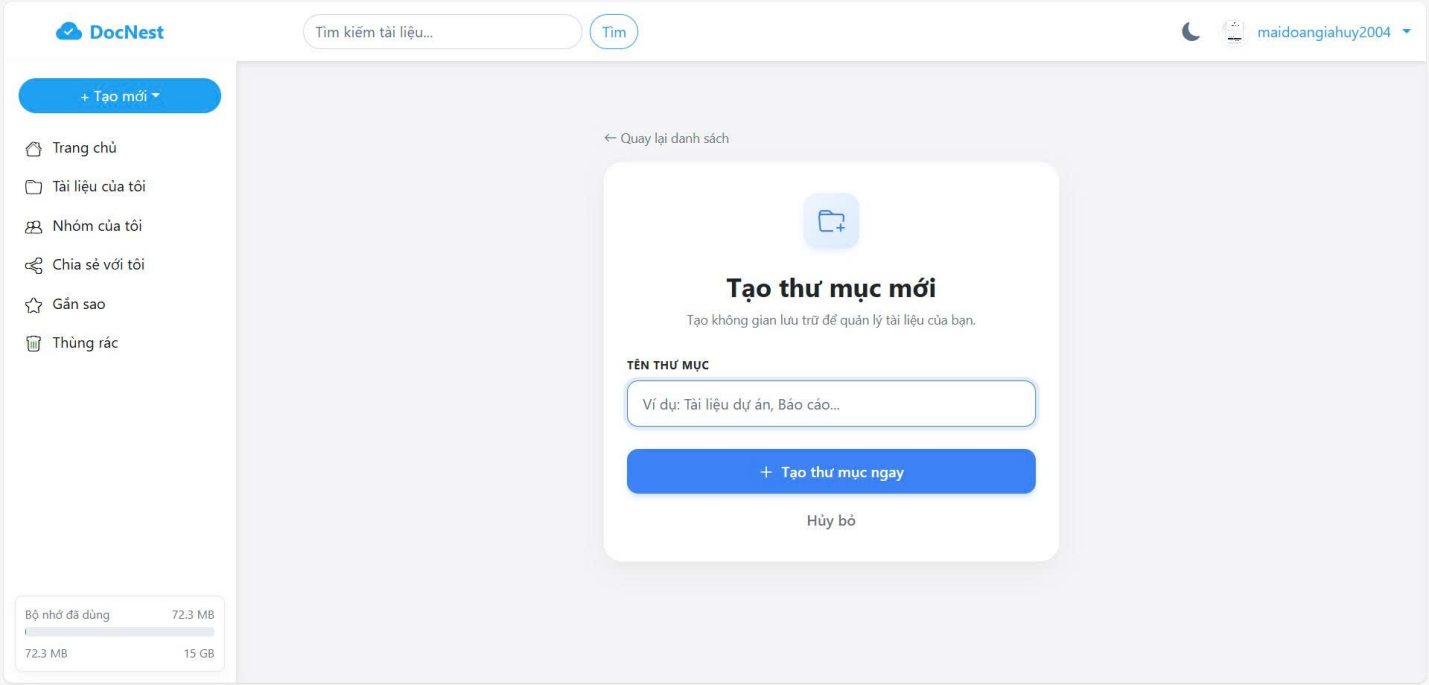
Hình N: Giao diện dropdown của nút “Tạo Mới”

Bên trong có các nút Thư Mục mới, Tải tệp lên, Tải thư mục lên. Cung cấp cho người dùng đa dạng lựa chọn để đăng tải



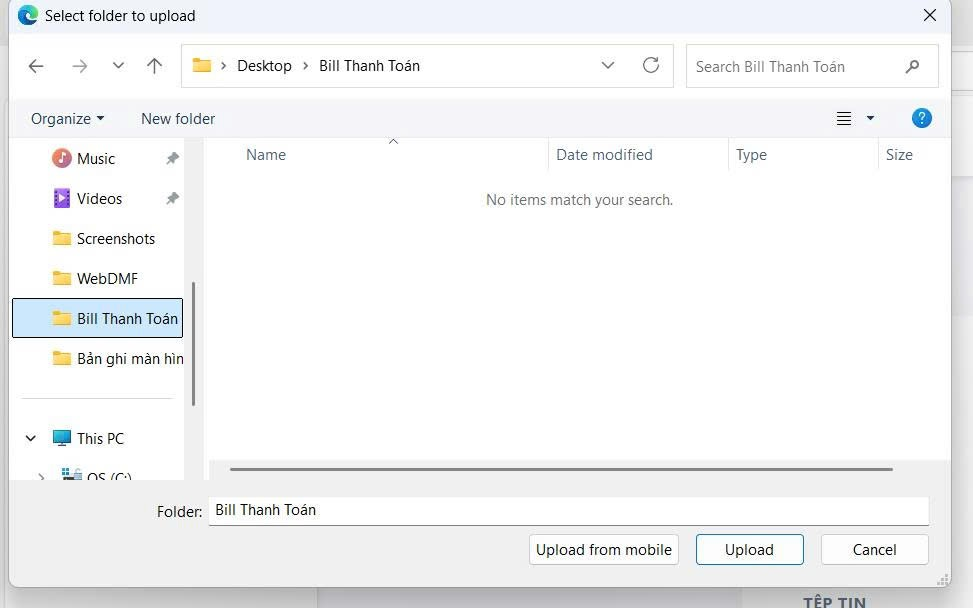
*Hình 5: Giao diện tải lên*

Giao diện đăng tải tệp tin



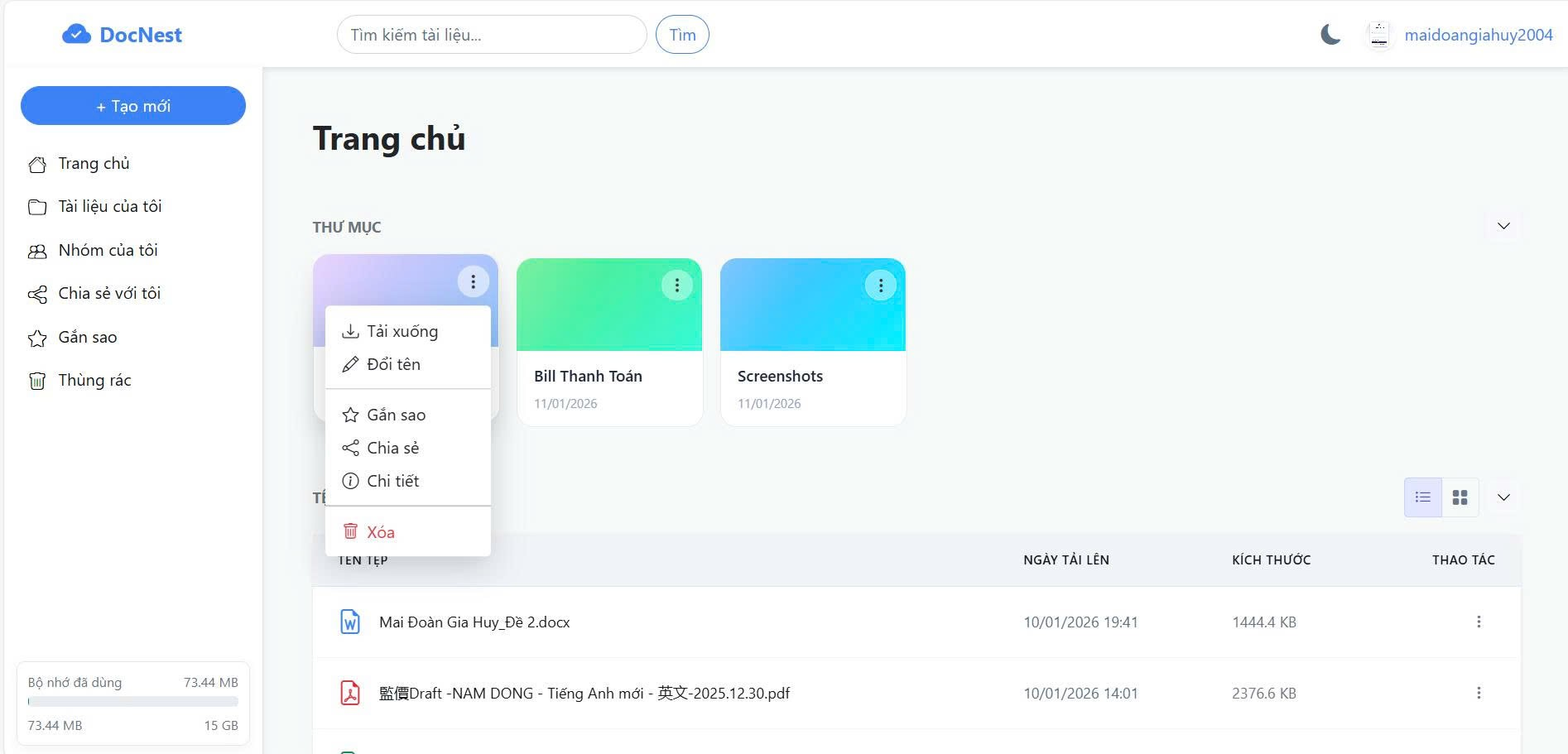
*Hình 8: Tạo Folder*

Giao diện tạo Folder mới



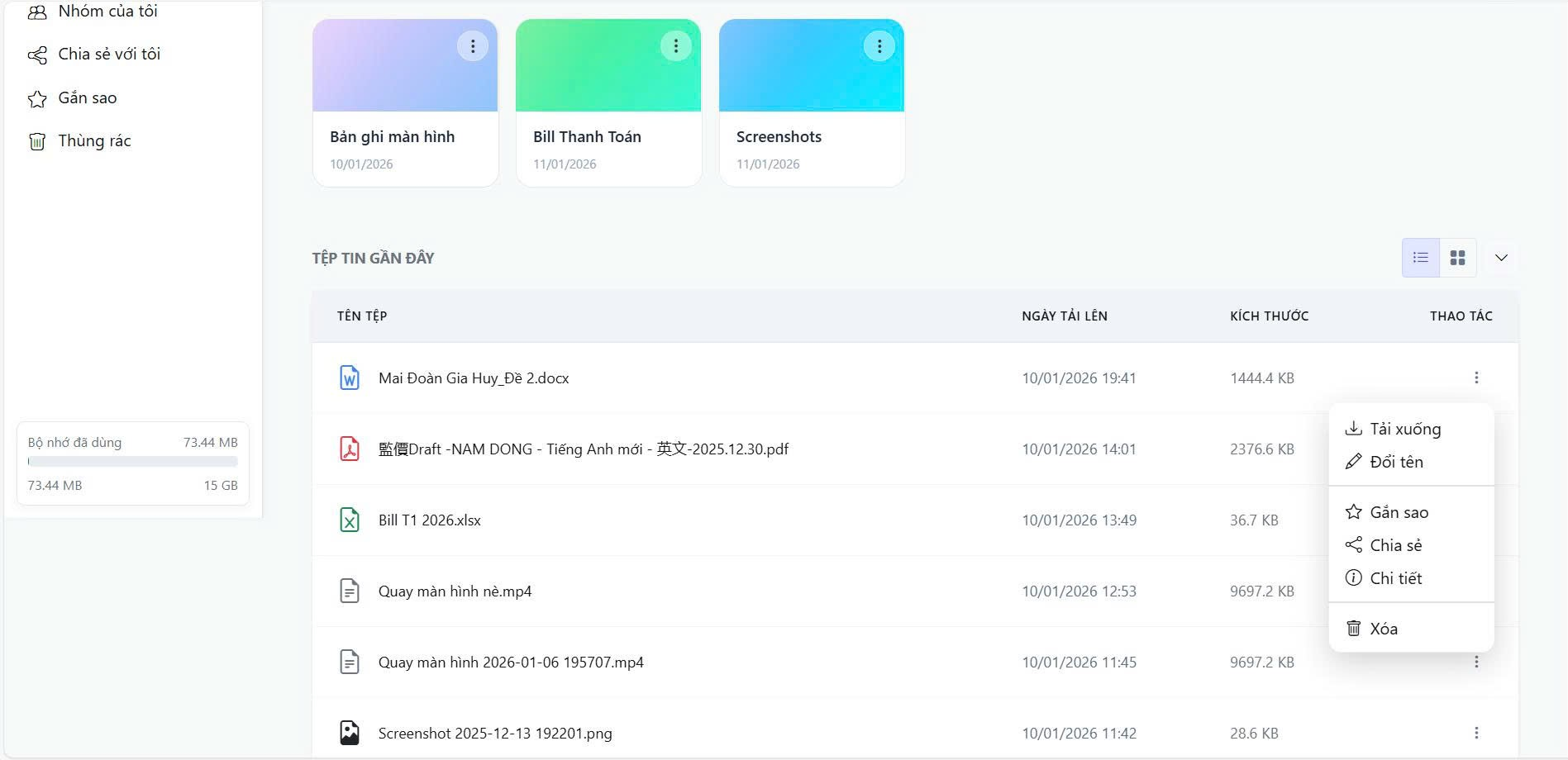
Hình N: Tải Folder

Giao diện lựa chọn thư mục để đăng tải lên hệ thống. Ngoài ra, người dùng còn có nút tùy chọn tải lên từ thiết bị di động.



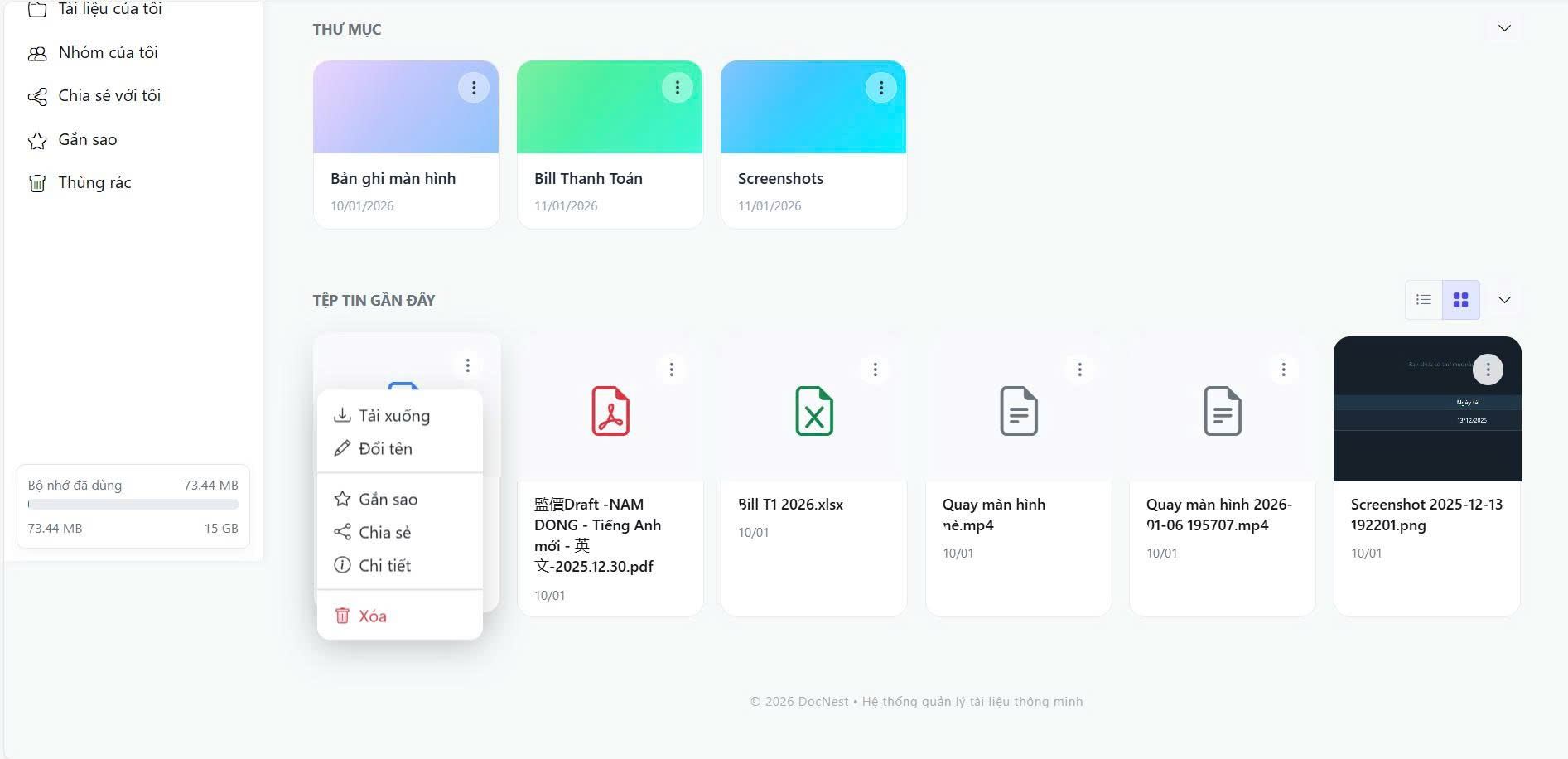
Hình N: Folder Dropdown

Nút dropdown của thư mục đã được tải lên DocNest. Cung cấp cho người dùng các tùy chọn để thao tác với thư mục.



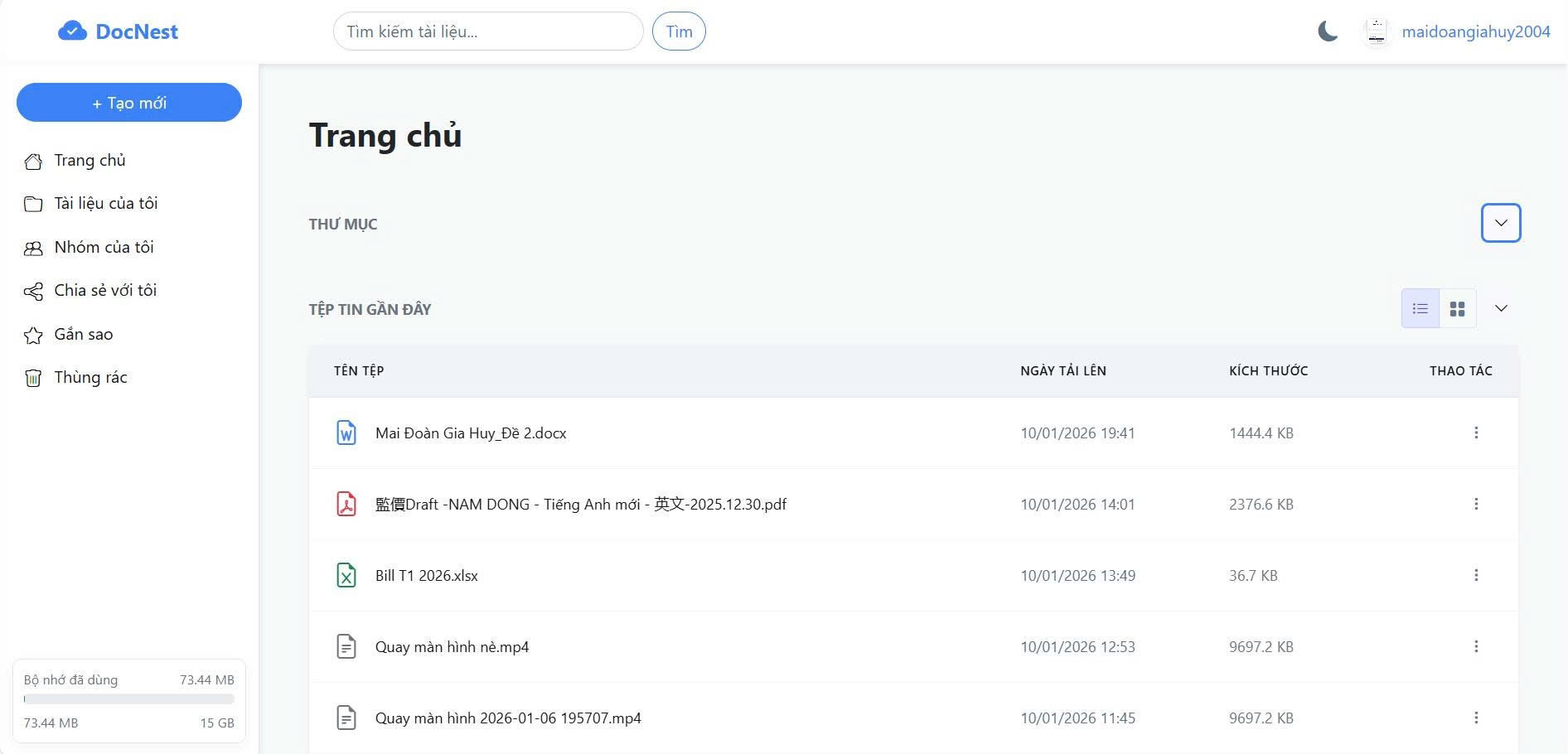
Hình N: Nút dropdown của tệp tin

Tương tự nút dropdown của thư mục. Cung cấp cho người dùng những tùy chọn, thao tác cơ bản để quản lý các tệp tin của mình.



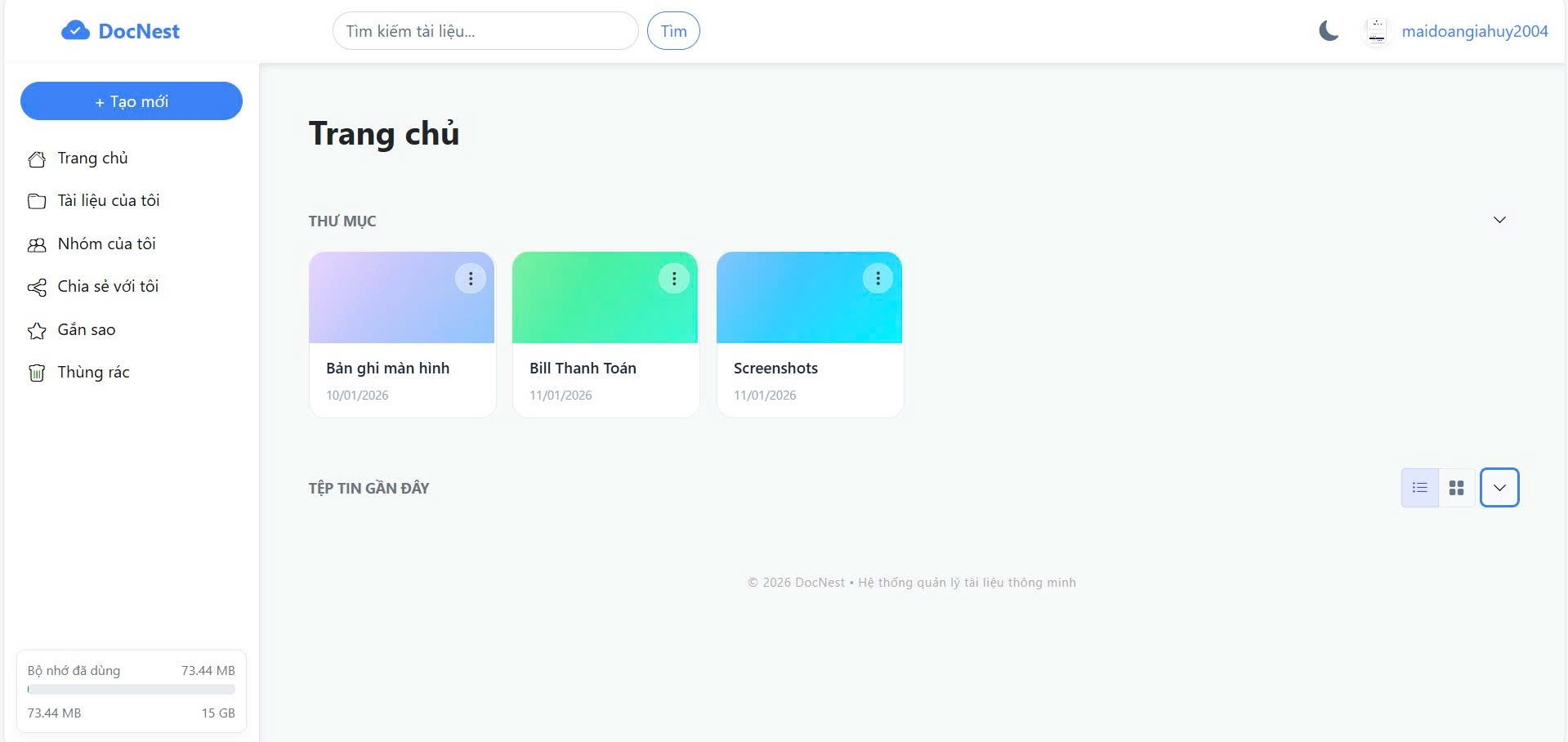
Hình N: Nút dropdown tệp tin (dạng lưới)

Giao diện nút dropdown của dạng lưới.

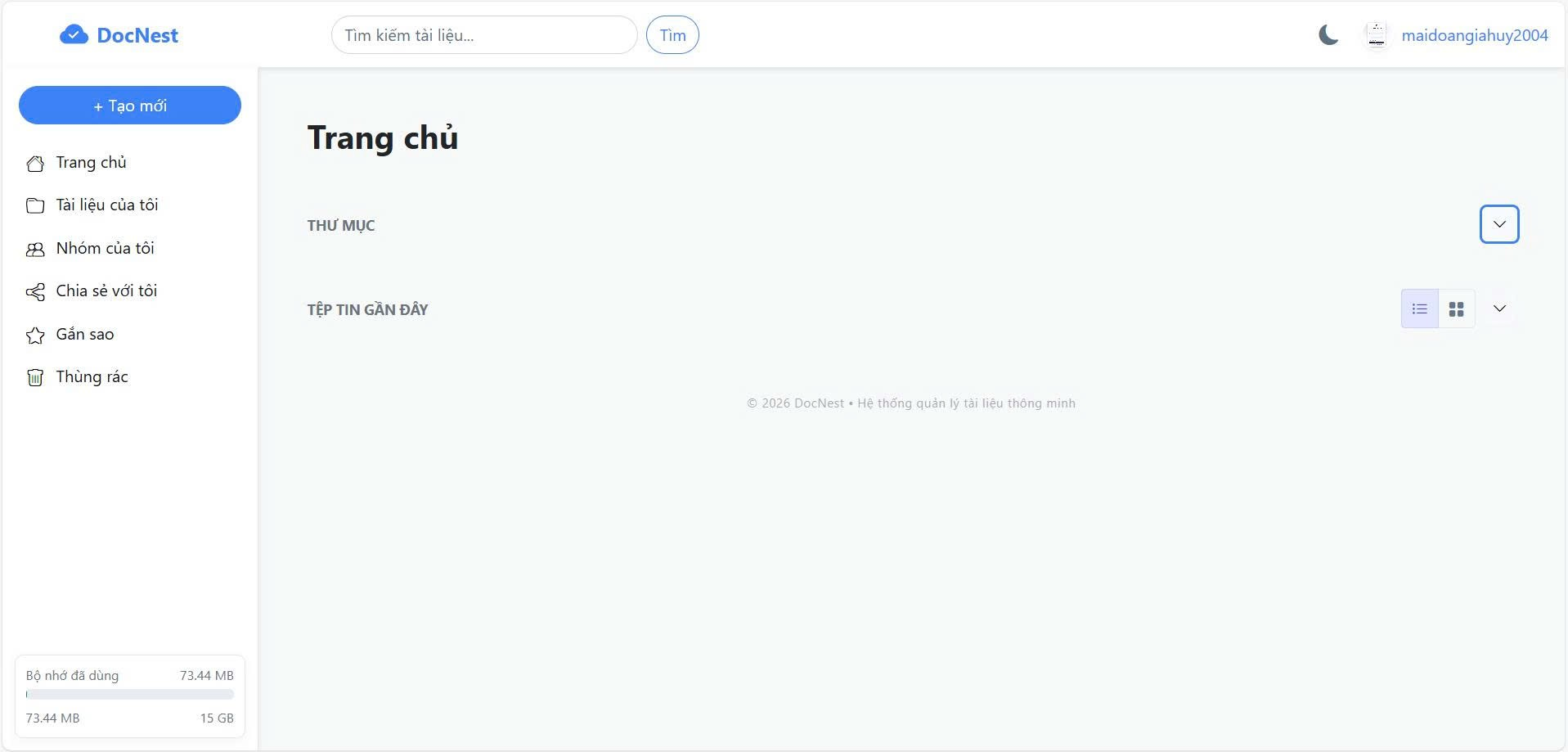


Hình N: Nút ấn Folder

Giao diện khi dùng nút mũi tên ẩn đi Folder.

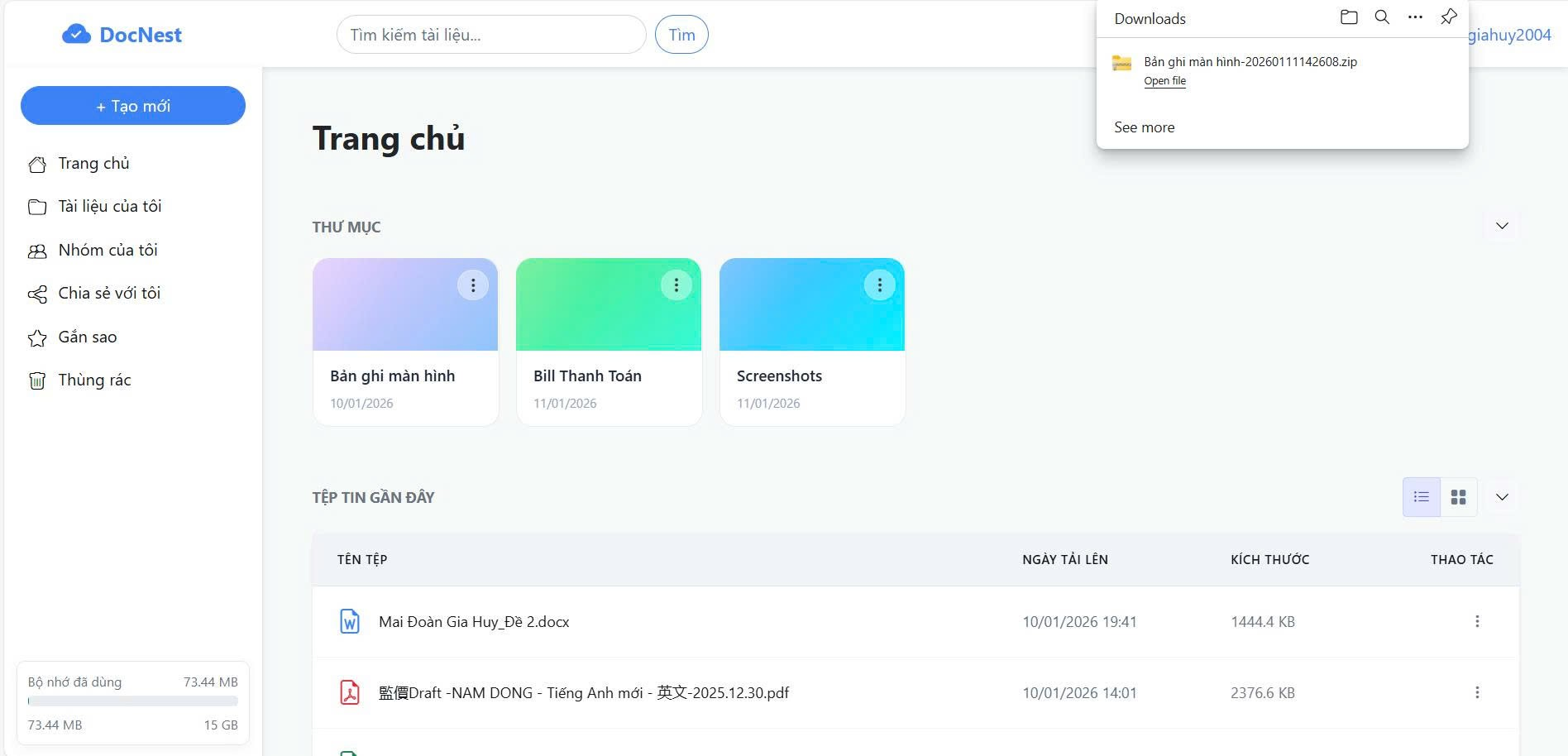
  
Hình N: Ẩn danh sách tệp tin

Nút mũi tên ẩn đi danh sách các tệp tin.

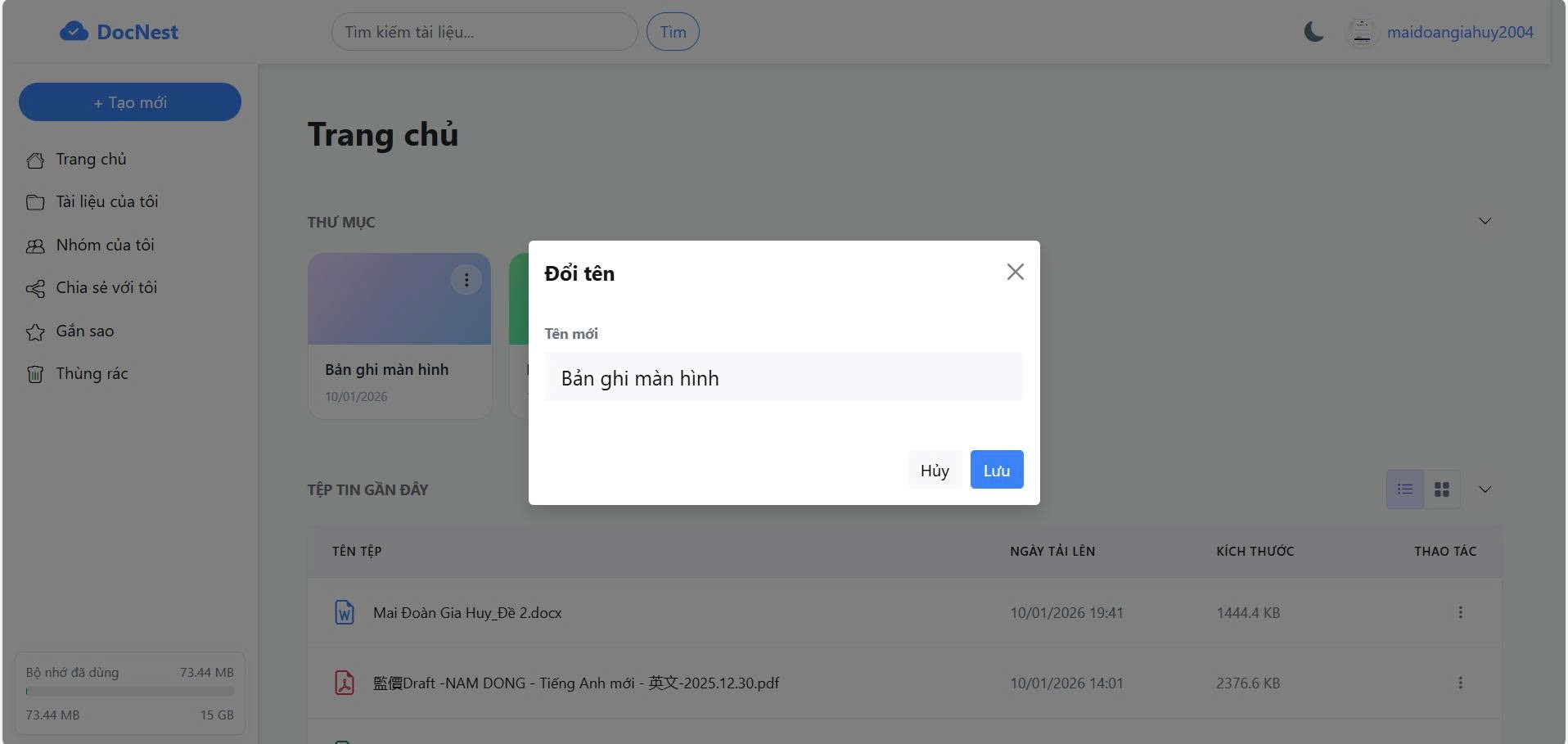


Hình N: Ấn cả hai

Giao diện ấn cả tệp tin và thư mục bằng nút mũi tên.

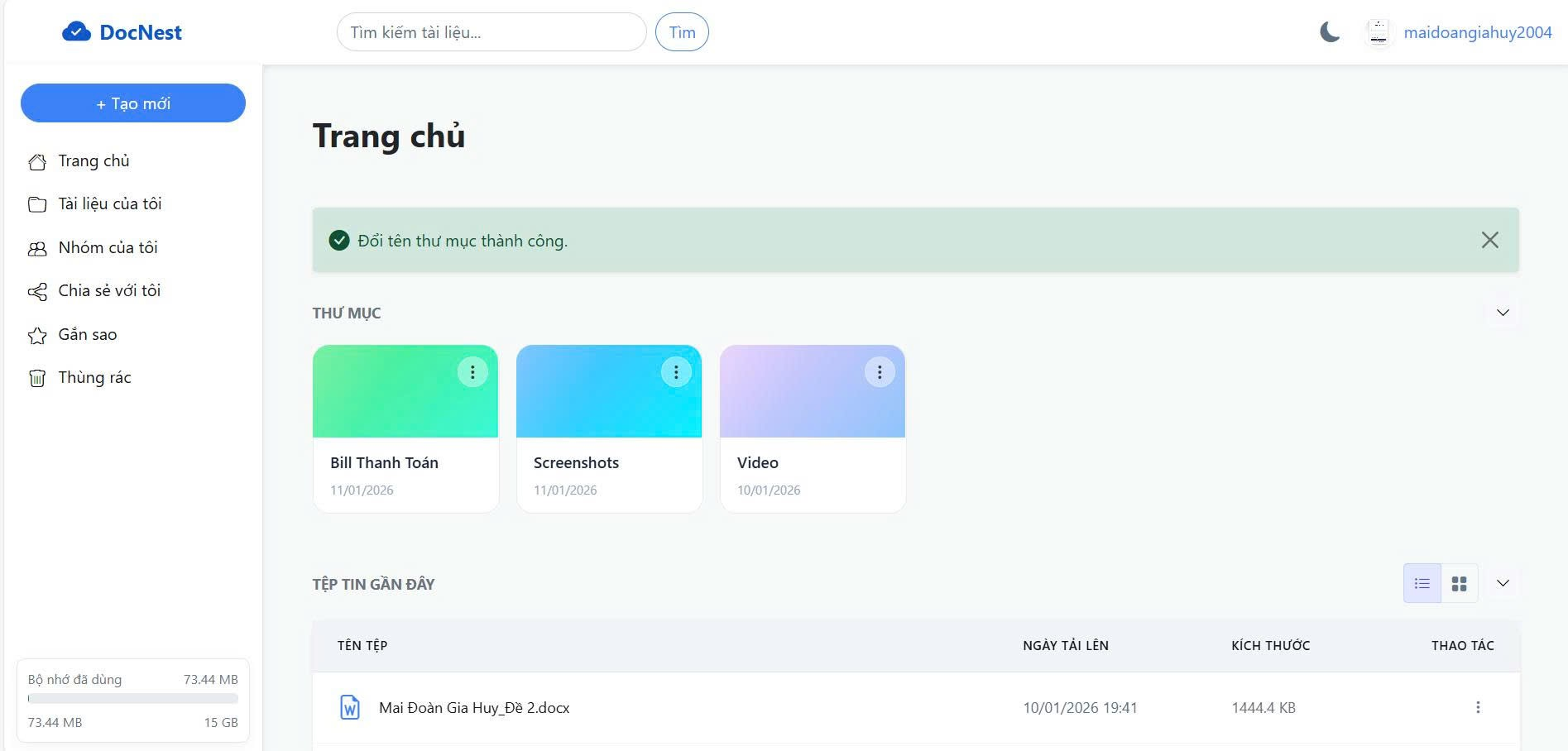
  
Hình N: Tải xuống thư mục

Người dùng nhấn vào nút ba chấm bên cạnh tệp để mở menu tùy chọn (dropdown), sau đó chọn Tải xuống để lưu tệp về máy.



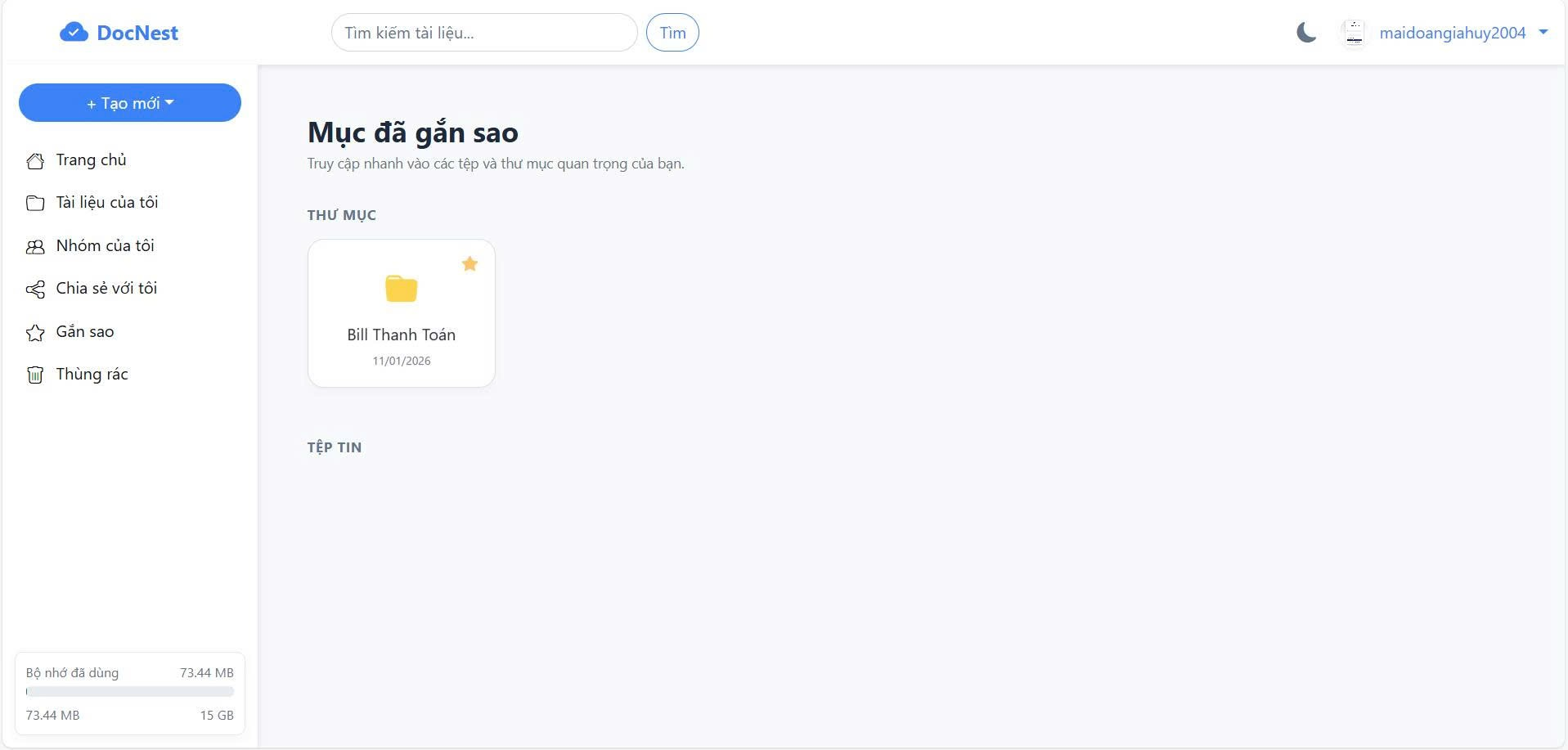
Hình N: Đổi tê thư mục

Người dùng nhấn vào nút ba chấm bên cạnh để mở menu tùy chọn (dropdown), sau đó chọn Đổi tên. Hệ thống sẽ hiện ra hộp textbox để nhập tên mới cho thư mục.



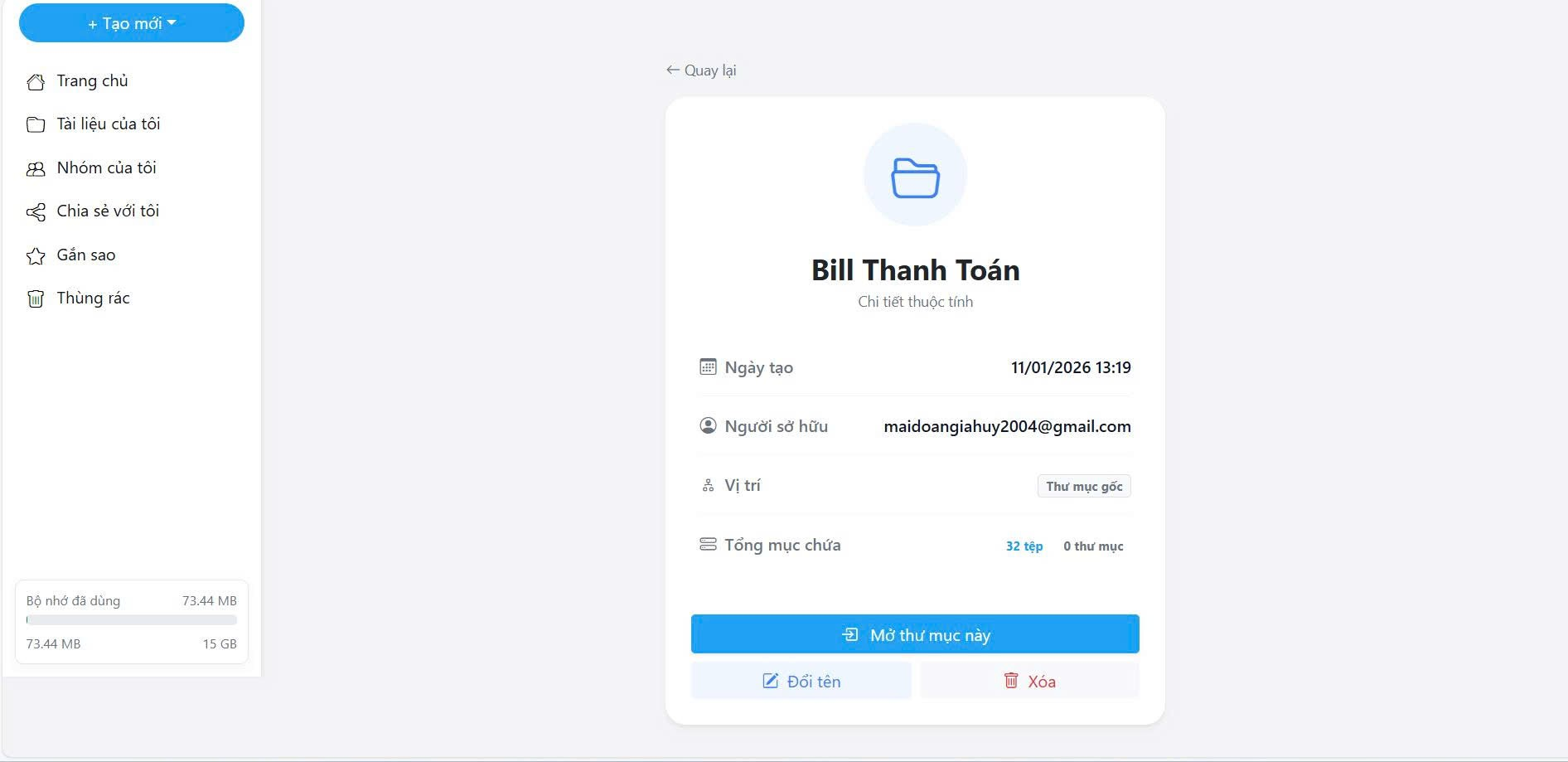
Hình N: Đổi tên thành công

Giao diện thông báo người dùng đã đổi tên thư mục thành công.



Hình N: Mục đã gắn sao

Người dùng bấm vào Folder nút gắn sao, hệ thống sẽ chuyển qua giao diện các mục được gắn sao.



Hình N: Xem chi tiết thư mục

Người dùng bấm vào nút ba chấm bên cạnh để mở menu tùy chọn (dropdown), chọn nút Chi tiết. Hệ thống sẽ hiển thị chi tiết ngày tạo, người sở hữu, vị trí và tổng mục chứa của tài liệu

## System Architecture

The system is designed using a layered web application architecture based on ASP.NET Core. The architecture follows the **Model–View–Controller (MVC)** pattern, combined with **Razor Pages** for handling authentication and user-related features. This approach helps separate responsibilities between data processing, application logic, and user interface, making the system easier to maintain and extend.

At the presentation layer, the system uses Razor views to display user interfaces and handle user interactions through a web browser. Bootstrap and JavaScript are applied to improve usability and responsiveness. The application layer contains controllers and services responsible for handling user requests, processing business logic, and enforcing access control rules, such as verifying file ownership before allowing access.

The data access layer is implemented using **Entity Framework Core**, which communicates with a **SQL Server** database. This layer manages data operations related to users, documents, folders, permissions, and system logs. The Code First approach is used to define database structures through models and manage schema changes using migrations.

For file storage, uploaded documents are saved directly on the server’s local file system within the wwwroot directory. File metadata is stored in the database, while the actual file content is managed at the server level. User authentication and authorization are handled by **ASP.NET Core Identity**, ensuring secure access to system features. Overall, the system architecture supports secure file management, controlled access, and efficient interaction between components.