**VINH UNIVERSITY**

**SCHOOL OF ENGINEERING AND TECHNOLOGY**

****

**PROJECT REPORT ON**

**WEB PROGRAMMING**

**BUILDING A WEBSITE FOR**

**A DIGITAL LIBRARY AND RESEARCH CENTER**

**GROUP: 13**

|  |  |
| --- | --- |
| **Instructor**: | Cao Thanh Son, PhD. |
| **Students**: | Tran Ngoc Nguyen, 215748020110090 |
|  | Tran Tien Dat, 225748020110094(Leader) |

**Nghe An, 6/202****5**

# ACKNOWLEDGMENTS

First and foremost, on behalf of Group 13, Class CNTT\_CLC, I would like to extend our deepest gratitude to Mr. Cao Thanh Son — the instructor who has wholeheartedly guided, supported, and accompanied us throughout the process of researching and developing this project. His dedicated instructions and valuable feedback have been a great source of motivation for our team to overcome challenges and gradually complete the project.

We would also like to express our sincere thanks to Mr. Tran Ngoc Nguyen and Mr. Tran Tien Dat for our enthusiastic assistance and collaboration during the implementation of this project. Our contributions in terms of ideas, technical support, and teamwork have been an important part of the project’s success.

In addition, we would like to send our heartfelt appreciation to our families, friends, and classmates, who have always encouraged, motivated, and shared their knowledge and experiences, helping us overcome difficulties throughout the entire working process.

Finally, we would like to thank all organizations and individuals who have shown interest and provided both moral and material support, directly or indirectly contributing to the successful completion of this project.

Once again, we are truly grateful and would like to express our sincerest thanks.

# PREAMBLE

## Reasons for choosing the topic

The digital transformation has significantly changed the way information is stored, accessed, and shared. Digital libraries provide a convenient platform for users to access a vast collection of books, journals, and research papers from anywhere in the world. Unlike traditional libraries, digital libraries remove physical barriers, making knowledge more accessible to a broader audience.

With the rise of online learning, research institutions, and digital education platforms, the need for reliable and efficient online research resources has grown rapidly. Students, researchers, and professionals increasingly prefer digital platforms for quick and easy access to academic content.

## Research objectives

The primary goal of this project is to develop a website that serves as a digital library and research center, offering users access to a variety of academic materials. The platform aims to be intuitive and easy to navigate, ensuring that users can efficiently locate and access the resources they need. The system will feature an advanced search function with filters based on categories such as author, publication year, and document type to help users find relevant materials quickly.

## Purpose and scope of the study

This project focuses on designing and developing a functional digital library system that allows users to search, access, and borrow digital books and research materials. The system is intended for students, researchers, educators, and professionals seeking academic materials for study and research purposes.

## Structure of the thesis

The thesis is structured into several chapters, covering the overview, analysis, design, development, and evaluation of the digital library system.

**TABLE OF CONTENTS**

[ACKNOWLEDGMENTS 1](#_Toc199080739)

[PREAMBLE 2](#_Toc199080740)

[1. Reasons for choosing the topic 2](#_Toc199080741)

[2. Research objectives 2](#_Toc199080742)

[3. Purpose and scope of the study 2](#_Toc199080743)

[4. Structure of the thesis 2](#_Toc199080744)

[LIST OF FIGURES 5](#_Toc199080745)

[CHAPTER 1. OVERVIEW OF DIGITAL LIBRARY AND RESEARCH CENTER 7](#_Toc199080746)

[1.1. Introduction to digital library and research center 7](#_Toc199080747)

[1.2. Opportunities in developing an online library system 8](#_Toc199080748)

[1.3. Related systems 10](#_Toc199080749)

[CHAPTER 2. ANALYSIS AND DESIGN OF THE DIGITAL LIBRARY AND RESEARCH CENTER SYSTEM 12](#_Toc199080750)

[2.1. Overview of the system 12](#_Toc199080751)

[2.2. Requirements analysis 12](#_Toc199080752)

[2.3. Database design 14](#_Toc199080753)

[2.3. Use case diagram 22](#_Toc199080754)

[CHAPTER 3. DEVELOPMENT OF THE DIGITAL LIBRARY AND RESEARCH CENTER WEBSITE 29](#_Toc199080755)

[3.1. Choosing programming languages 29](#_Toc199080756)

[3.2. Project management on Trello 32](#_Toc199080757)

[3.3. Interface design 34](#_Toc199080758)

[CONCLUSTIONS 60](#_Toc199080759)

[1. Achievement 60](#_Toc199080760)

[2. Limitations 60](#_Toc199080761)

[3. Future direction 60](#_Toc199080762)

[REFERENCES 61](#_Toc199080763)

*(Righ click on above area, choose* ***Update Field****, and then* ***Update entire table*** *to update the Table of contents - delete this line when printing)*.

LIST OF FIGURES

[Fig 1.1. Digital library 7](#_Toc199080630)

[Fig 2.1. Database schema 14](#_Toc199080631)

[Fig 2.2. Use case diagram 22](#_Toc199080632)

[Fig 2.3. Function forgot passwordsequence diagram 23](#_Toc199080633)

[Fig 2.4. Function registration sequence diagram 24](#_Toc199080634)

[Fig 2.5. Function post a book diagram 25](#_Toc199080635)

[Fig 2.6. Function post scientific research diagram 26](#_Toc199080636)

[Fig 2.7. Function post a comment diagram 27](#_Toc199080637)

[Fig 2.8. Function rating a book diagram 28](#_Toc199080638)

[Fig 3.1. HTML & CSS 30](#_Toc199080639)

[Fig 3.2. ASP.NET 31](#_Toc199080640)

[Fig 3.3. SQL Server 31](#_Toc199080641)

[Fig 3.4. Trello project management tool 32](#_Toc199080642)

[Fig 3.5. Login Interface 35](#_Toc199080643)

[Fig 3.6. Forgot Password Interface 36](#_Toc199080644)

[Fig 3.7. Register Interface 37](#_Toc199080645)

[Fig 3.8. User Page Interface 38](#_Toc199080646)

[Fig 3.9. User Account Details Interface 39](#_Toc199080647)

[Fig 3.10. User Book Registration Interface 40](#_Toc199080648)

[Fig 3.11. User Book Management Interface 41](#_Toc199080649)

[Fig 3.12. User Research Management Interface 42](#_Toc199080650)

[Fig 3.13. Admin Interface 43](#_Toc199080651)

[Fig 3.14. Admin Profile Interface 44](#_Toc199080652)

[Fig 3.15. Admin Edit Profile Interface 45](#_Toc199080653)

[Fig 3.16. Admin Need Help Interface 46](#_Toc199080654)

[Fig 3.17. Admin Document Management Interface 47](#_Toc199080655)

[Fig 3.18. Admin Document Details Interface 48](#_Toc199080656)

[Fig 3.19. Admin Review The Document Interface 49](#_Toc199080657)

[Fig 3.20. Admin Document Classification Interface 50](#_Toc199080658)

[Fig 3.21. Admin Label The Document Interface 51](#_Toc199080659)

[Fig 3.22. Admin Research Project Management Interface 52](#_Toc199080660)

[Fig 3.23. Admin Research Project Details Interface 53](#_Toc199080661)

[Fig 3.24. Admin Comment Management Interface 54](#_Toc199080662)

[Fig 3.25. Admin Rating Management Interface 55](#_Toc199080663)

[Fig 3.26. Admin Comment Statistics Interface 56](#_Toc199080664)

[Fig 3.27. Admin User Management Interface 57](#_Toc199080665)

[Fig 3.28. Admin Account Upgrade Approval Interface 58](#_Toc199080666)

[Fig 3.29. Admin Account Upgrade Details Interface 59](#_Toc199080667)

# CHAPTER 1. OVERVIEW OF DIGITAL LIBRARY AND RESEARCH CENTER

## 1.1. Introduction to digital library and research center

### 1.1.1. Definition of a digital library

A digital library is an online platform that provides access to a collection of digital content, including books, journals, and research papers. It allows users to search, read, and borrow materials without physical constraints.



Fig 1.1. Digital library

### 1.1.2. Key components of a digital library system

An effective digital library system is built on many functional components that work closely together to ensure the ability to store, search, access and manage academic resources conveniently and securely. The main components include:

Database Management System:

This is the platform for storing and organizing all data of the system, including e-books, research articles, user accounts, reviews, comments and interactive activities. The database needs to be designed to be standardized, support scalability and optimize queries to ensure system performance.

Allows registration, login, account authorization (e.g. regular user, researcher, administrator), profile management, and processing of account upgrade requests. The system also supports security through two-factor authentication (2FA), OTP codes, and security alert options.

A core function that allows users to easily access scholarly resources. The system should support searching by keyword, author, year of publication, genre, as well as suggesting results based on relevance or popularity. Integrating advanced search engines (such as Elasticsearch) helps improve the query experience.

As the bridge between the user and the system, the UI needs to be designed to be intuitive, easy to use, and responsive across multiple devices (computers, tablets, phones). The interface includes pages such as: login, registration, personal dashboard, book/research management page, and community area (comments, reviews).

The system needs to integrate security layers such as data encryption, role-based access control (RBAC), and user authentication. In addition, limiting access to unapproved documents or copyrighted content is essential to ensure legality and data protection.

### 1.1.3. Current trends in online research

Digital libraries are evolving with emerging technologies such as Artificial Intelligence (AI) and blockchain for content verification, enhancing user experience and data security.

## 1.2. Opportunities in developing an online library system

Developing an online library system offers numerous opportunities such as providing 24/7 access to knowledge, especially for remote users, and enhancing learning through personalized recommendations, search tools, and user interaction features like ratings and comments. It supports content management, allowing authors or educators to upload and share materials easily. The system can integrate with educational platforms, track user activity for analytics, and offer monetization options like subscriptions or paid content. Additionally, it helps build a learning community, supports scalability through cloud infrastructure, ensures security with role-based access, and opens doors to global reach with multilingual support.

### 1.2.1. Advantages of digital over traditional libraries

The development of digital technology has brought many outstanding advantages to the electronic library model compared to traditional libraries. These advantages not only contribute to improving the efficiency of learning and research but also expand the scope of knowledge access for the global user community. Specifically:

24/7 Access from Anywhere: Digital libraries allow users to access resources anytime and from anywhere with just an internet connection, completely eliminating the time and space limitations of traditional libraries.

Quick search and retrieval: With the smart search system, users can quickly find the desired document through keywords, authors, topics or year of publication, instead of having to spend time manually searching on the bookshelf.

Saving costs and resources: Digitizing documents helps reduce costs of printing, physical preservation, space rental, on-duty personnel, etc., contributing to optimizing library operating budgets.

Unlimited storage: Digital library systems can store millions of documents on servers or cloud platforms without the physical space limitations of traditional libraries.

High integration and interactivity: Materials in the digital library can be linked to online learning systems (LMS), allowing them to be embedded into lectures, assessed, or shared on other platforms. In addition, users can comment, rate, and discuss directly on the platform.

### 1.2.2. Potential challenges and solutions

**Although digital libraries bring many outstanding benefits, the process of building and operating the system still has many challenges. Early identification of potential problems and proposing reasonable solutions is necessary to ensure the system operates stably, effectively and sustainably.**

Challenge: Protecting Copyright and Intellectual Property

- Problem: Digitizing and distributing academic materials is vulnerable to copyright infringement or unauthorized distribution.

- Solution: Integrate access control mechanisms such as user authentication, authorization, and use of Digital Rights Management (DRM) systems to limit unauthorized copying, downloading, or sharing of materials.

Challenge: Ensuring content quality

- Problem: Users may post poor quality, inappropriate or duplicate content.

- Solution: Implement a two-step moderation process, including self-review by the poster and review by the administrator, combined with the use of a plagiarism checker if necessary.

Challenge: Suboptimal user experience

- Problem: Difficult, unfriendly, or unresponsive interfaces on mobile devices will prevent users from accessing resources.

- Solution: Design a modern, intuitive user interface (UI), support responsive design, and conduct regular user surveys to improve features based on actual needs.

## 1.3. Related systems

### 1.3.1. Comparison with similar online library platforms

An online library system can be compared to popular platforms like Google Books, OverDrive, or Open Library. While Google Books focuses on digitized book previews and purchase options, and OverDrive caters mainly to public libraries and paid lending, a custom online library system offers greater flexibility in features such as personalized user dashboards, custom roles (admin, student, teacher), real-time analytics, and direct content uploads from users or educators. Unlike Open Library, which aims to archive all published books, a custom system can be tailored to specific needs such as educational institutions, private collections, or community learning. This gives it an edge in targeted functionality, integration with learning systems, and control over content curation and user engagement.

### 1.3.2. Strengths and weaknesses of current solutions

In recent years, many digital library platforms such as Google Books, OverDrive, Open Library, or internal university systems have been deployed to digitize knowledge and serve online academic needs. However, besides the obvious benefits, current systems still have some notable limitations.

**- Limited Customization:** Existing platforms offer limited flexibility for institutions to adapt to specific workflows, roles, or branding.

**- Restricted Access to Full Content:** Many books are only partially viewable (Google Books) or require library affiliation (OverDrive).

**- Monetization Focus:** Heavily commercialized models limit free access or open educational use.

**- Lack of Educational Integration:** Few platforms support integration with LMS, assignments, or real-time academic tracking.

**- User Engagement Gaps:** Limited features for personalized recommendations, feedback, gamification, or community interaction compared to what modern systems can offer.

# CHAPTER 2. ANALYSIS AND DESIGN OF THE DIGITAL LIBRARY AND RESEARCH CENTER SYSTEM

## 2.1. Overview of the system

The Digital Library and Research Center System is designed to provide a centralized platform for storing, accessing, and interacting with a wide range of digital learning materials and academic resources. The system serves multiple user roles, including administrators, students, and researchers, allowing them to upload, search, read, rate, and comment on digital content such as e-books, research papers, and academic documents.

The platform aims to bridge the gap between traditional libraries and modern digital needs by offering 24/7 access, intelligent content organization, and user-driven interaction. It includes essential modules for user management, book and document uploads, rating and review functionalities, comment threads, and administrative tools for content moderation and system monitoring.

Additionally, the system supports data analytics to track reading trends and user engagement, which can help institutions evaluate academic performance and content popularity. With a focus on scalability, security, and usability, the system is well-suited for educational institutions, research centers, and digital knowledge-sharing communities.

## 2.2. Requirements analysis

In order to build an digital librrary and research center website, the process of thoroughly analyzing and defining requirements plays an essential role. During this phase, all requirements are classified into two main categories: functional requirements and non-functional requirements. This classification helps ensure that the design and development process is organized, and that the system will not only meet user needs but also operate stably, flexibly, and securely.

### 2.2.1. Functional requirements

Functional requirements describe the core features that a digital library system needs to provide to meet the needs of end users (students, lecturers, researchers) as well as administrators. These functions are designed to ensure the system operates effectively, supporting the process of searching, posting, managing and accessing digital documents.

- The system must allow users to register new accounts and log in securely, ensuring that only authorized individuals have access to the platform.

- Administrators are responsible for content management on the system, so they must be able to create new book, edit book details, delete book, and approve book before they are publicly available to users.

- Subsequent to each reading experience, the system should allow users to submit feedback and reviews about the event’s quality, which can help organizers improve future events.

### 2.2.2. Non-functional requirements

Non-functional requirements are important characteristics that are not directly related to specific functions, but strongly influence the performance, user experience, stability, and security of the digital library system. These requirements play a role in ensuring that the system operates efficiently, reliably, and is suitable for the actual deployment environment.

- Scalability: The system must be designed to be flexible, allowing new features or adjustments to be added easily whenever new requirements arise or when the scale of events expands in the future.

- Performance: The system must ensure fast response times, smooth page loading, and efficient data processing, even when handling a large number of concurrent users, especially during peak times of major events.

- Security: The system must protect user personal information and event data through secure encryption techniques, user authentication processes, and strict role-based access control.

- Availability: The system should maintain high availability, operating stably and continuously 24/7, and support flexible access across a variety of devices, including personal computers, tablets, and mobile phones.

- User-friendly: The system interface should be designed to be intuitive, easy to understand, and user-friendly, with a focus on providing a smooth user experience, allowing both students and organizers to interact with the platform comfortably from their first use.

## 2.3. Database design

The database of the digital library management system is designed with a clear and well-structured schema to efficiently organize data, ensure scalability, and support future development.

### 2.3.1. Database schema

A computer screen shot of a computer

AI-generated content may be incorrect.

Fig 2.1. Database schema

### 2.3.2. Database table design

a) AdminUser table

The system revolves around several core entities, each playing a specific role in managing users, digital books, user interactions, and content moderation. At the heart of the system is the AdminUser table, which stores essential information about all users of the system, including administrators and general users. Each record contains fields such as username, email, password, role, account status (IsActive), and personal information like birthdate, gender, address, and phone number. The table also includes advanced security features such as two-factor authentication (2FA), password reset tokens, and security alert preferences, ensuring secure access and role-based authorization.

|  |  |  |  |
| --- | --- | --- | --- |
|  | Column Name | Data Type | Constraint |
| 1 | UserName | int | Primary key |
| 2 | Email | nvarchar(255) |  |
| 3 | Password | nvarchar(255) |  |
| 4 | IsActive | bit |  |
| 5 | PasswordResetToken | nvarchar(255) |  |
| 6 | Role | int |  |
| 7 | Avatar | nvarchar(500) |  |
| 8 | BirthDate | datetime |  |
| 9 | Gender | nvarchar(10) |  |
| 10 | Address | nvarchar(255) |  |
| 11 | PhoneNumber | nvarchar(20) |  |
| 12 | IsTwoFactorEnabled | bit |  |
| 13 | ReceiveSecurityAlerts | bit |  |
| 14 | TwoFactorToken | nvarchar(100) |  |
| 15 | TwoFactorTokenExpiry | datetime |  |

b) Book table

The Book table manages all uploaded books in the system. Each book includes metadata such as title, category name, author, published year, cover image, file path, and a detailed description. It also tracks the number of views and whether the book has been approved for public visibility (IsApproved). Every book is associated with a user via a foreign key (UserID), which allows the system to trace the origin of the uploaded content.

|  |  |  |  |
| --- | --- | --- | --- |
|  | Column Name | Data Type | Constraint |
| 1 | BookID | int | Primary key |
| 2 | UserID | int | Foreign key |
| 3 | Title | nvarchar(255) |  |
| 4 | CategoryName | nvarchar(255) |  |
| 5 | AuthorName | nvarchar(255) |  |
| 6 | PublishedYear | int |  |
| 7 | CoverImage | nvarchar(500) |  |
| 8 | FilePath | nvarchar(500) |  |
| 9 | TotalPages | int |  |
| 10 | Description | text |  |
| 11 | IsApproved | bit |  |
| 12 | Views | int |  |
| 13 | CreatedAt | datetime |  |
| 14 | RejectReason | nvarchar(500) |  |

c) Rating table

The Rating table handles book evaluations submitted by users. Each rating is linked to a user and a specific book and includes a star rating and the timestamp of creation. These records enable the system to measure user satisfaction and provide aggregated feedback for each book, which can be used to guide quality improvements and content moderation.

|  |  |  |  |
| --- | --- | --- | --- |
|  | Column Name | Data Type | Constraint |
| 1 | RatingID | int | Primary key |
| 2 | UserID | int | Foreign key |
| 3 | BookID | int | Foreign key |
| 4 | Rating | int |  |
| 5 | CreatedAt | datetime |  |

d) Comment table

The Comment table allows users to leave comments on books, fostering interaction and discussion among readers. Each comment record includes the user who posted it, the associated book, the comment content, creation time, like count, and dislike count. This structure supports a dynamic and interactive reading community within the system.

|  |  |  |  |
| --- | --- | --- | --- |
|  | Column Name | Data Type | Constraint |
| 1 | CommentID | int | Primary key |
| 2 | UserID | int | Foreign key |
| 3 | BookID | int | Foreign key |
| 4 | [Content] | text |  |
| 5 | CreatedAt | datetime |  |
| 6 | LikeCount | int |  |
| 7 | DislikeCount | int |  |

e) Bookmark table

The Bookmark table helps users save their current reading positions in books. Each bookmark entry contains the user ID, the book, the page number bookmarked, an optional note, and the creation timestamp.

|  |  |  |  |
| --- | --- | --- | --- |
|  | Column Name | Data Type | Constraint |
| 1 | BookmarkID | int | Primary key |
| 2 | UserID | int | Foreign key |
| 3 | BookID | int | Foreign key |
| 4 | PageNumber | int |  |
| 5 | Note | nvarchar(500) |  |
| 6 | CreatedAt | datetime |  |

f) BookLabel table

The BookLabel table allows tagging books with labels for better classification and searchability. Each record links a book to a label name and includes the time the label was assigned.

|  |  |  |  |
| --- | --- | --- | --- |
|  | Column Name | Data Type | Constraint |
| 1 | LabelID | int | Primary key |
| 2 | BookID | int | Foreign key |
| 3 | Label | int | nvarchar(50) |
| 4 | CreatedAt | text | datetime |

g) Research table

The Research table manages academic research submissions by users. Each research item contains metadata such as title, field, abstract, file path, status, published year, and confirmation status.

|  |  |  |  |
| --- | --- | --- | --- |
|  | Column Name | Data Type | Constraint |
| 1 | ResearchID | int | Primary key |
| 2 | UserID | int | Foreign key |
| 3 | Title | nvarchar(255) |  |
| 4 | Field | nvarchar(255) |  |
| 5 | Abstract | text |  |
| 6 | FilePath | nvarchar(500) |  |
| 7 | Status | nvarchar(50) |  |
| 8 | PublishedYear | int |  |
| 9 | CreatedAt | datetime |  |
| 10 | IsConfirmed | bit |  |

h) ResearchLabel table

The ResearchLabel table functions similarly to BookLabel, enabling tags for research documents to enhance discoverability.

|  |  |  |  |
| --- | --- | --- | --- |
|  | Column Name | Data Type | Constraint |
| 1 | LabelID | int | Primary key |
| 2 | ResearchID | int | Foreign key |
| 3 | Label | nvarchar(50) |  |
| 4 | CreatedAt | datetime |  |

i) ResearchComment table

The ResearchComment table allows users to provide feedback on research entries, with each record storing the commenter, the research ID, content, timestamps, and reaction counts.

|  |  |  |  |
| --- | --- | --- | --- |
|  | Column Name | Data Type | Constraint |
| 1 | CommentID | int | Primary key |
| 2 | UserID | int | Foreign key |
| 3 | [Content] | int | Foreign key |
| 4 | CreatedAt | datetime |  |
| 5 | LikeCount | int |  |
| 6 | DislikeCount | int |  |

j) UpgradeRequest table

The UpgradeRequest table supports user upgrade applications, enabling users to submit personal and academic documents—such as portraits, academic degrees, transcripts, and essays—for administrator review. It maintains comprehensive metadata including request status (e.g., pending, approved, rejected), submission timestamps, administrator review notes, and final decision outcomes. The schema ensures integrity and traceability by linking each request to a specific user account and logging all administrative actions. This well-structured design not only supports current upgrade workflows within the digital library system but also lays a robust foundation for future enhancements, such as automated document verification, multi-stage review processes, audit trails, and integration with external academic credential services. Its modular structure also supports scalability and adaptability for broader roles or privileges beyond academic upgrades.

|  |  |  |  |
| --- | --- | --- | --- |
|  | Column Name | Data Type | Constraint |
| 1 | RequestID | int | Primary key |
| 2 | UserID | int | Foreign key |
| 3 | FullName | nvarchar(255) |  |
| 4 | Portrait | nvarchar(500) |  |
| 5 | DegreeFile | nvarchar(500) |  |
| 6 | Essay | nvarchar(500) |  |
| 7 | Status | nvarchar(50) |  |
| 8 | SubmittedAt | datetime |  |
| 9 | ReviewedAt | datetime |  |
| 10 | ReviewNote | nvarchar(500) |  |

## 2.3. Use case diagram

A diagram of a diagram

AI-generated content may be incorrect.

Fig 2.2. Use case diagram

### 2.3.1. General functions

a) Forgot password

|  |  |
| --- | --- |
| **Purpose** | Forgot password |
| **Preconditions** | Already have an account |
| **implementation phase** | 1. User selects forgot password function to retrieve password  2. User fill personal information in the form  3. Interface sends data to the system for processing and notifies the user of success |
| **Additional phases** | In step 2, if the data is invalid, the system will notify you to re-enter the data |
| **Exceptions** | The system reports an error if the user entered data does not exist |
| **References** | No |

A diagram of a project

AI-generated content may be incorrect.

Fig 2.3. Function forgot passwordsequence diagram

b) Edit personal profile

|  |  |
| --- | --- |
| **Purpose** | Edit personal profile |
| **Preconditions** | Already have an account |
| **implementation phase** | 1. User selects the function to request to change personal profile information  2. User edits personal information in the form  3. Interface sends data to the system for processing and notifies the user of success |
| **Additional phases** | In step 2, if the data is invalid, the system will notify you to re-enter the data |
| **Exceptions** | The system reports an error if the user entered data is duplicated |
| **References** | No |

A diagram of a project

AI-generated content may be incorrect.

Fig 2.4. Function registration sequence diagram

c) Post a book

|  |  |
| --- | --- |
| **Purpose** | Post a book |
| **Preconditions** | No |
| **implementation phase** | 1. User selects the function to request to post a book  2. User fills in the required information in the form  3. Interface sends data to the system for processing and notifies the user of success |
| **Additional phases** | In step 2, if the data is invalid, the system will notify you to re-enter the data |
| **Exceptions** | The system reports an error if the user entered data is duplicated |
| **References** | No |

A diagram of a diagram

AI-generated content may be incorrect.

Fig 2.5. Function post a book diagram

d) Post scientific research

|  |  |
| --- | --- |
| **Purpose** | Post scientific research |
| **Preconditions** | No |
| **implementation phase** | 1. User selects the function to request to post scientific research  2. User fills in the required information in the form  3. Interface sends data to the system for processing and notifies the user of success |
| **Additional phases** | In step 2, if the data is invalid, the system will notify you to re-enter the data |
| **Exceptions** | The system reports an error if the user entered data is duplicated |
| **References** | No |

A diagram of a diagram

AI-generated content may be incorrect.

Fig 2.6. Function post scientific research diagram

e) Post a comment

|  |  |
| --- | --- |
| **Purpose** | Post a comment |
| **Preconditions** | No |
| **implementation phase** | 1. User selects the function to request to post a comment  2. User fills in the required information in the form  3. Interface sends data to the system for processing and notifies the user of success |
| **Additional phases** | In step 2, if the data is invalid, the system will notify you to re-enter the data |
| **Exceptions** | The system reports an error if the user input data is blank. |
| **References** | No |

A diagram of a diagram

AI-generated content may be incorrect.

Fig 2.7. Function post a comment diagram

f) Rating

|  |  |
| --- | --- |
| **Purpose** | Rating a book |
| **Preconditions** | No |
| **implementation phase** | 1. User selects the function to request to rating a book  2. Interface sends data to the system for processing and notifies the user of success |
| **Additional phases** | No |
| **Exceptions** | No |
| **References** | No |

A diagram of a diagram of a diagram

AI-generated content may be incorrect.

Fig 2.8. Function rating a book diagram

# CHAPTER 3. DEVELOPMENT OF THE DIGITAL LIBRARY AND RESEARCH CENTER WEBSITE

## 3.1. Choosing programming languages

In this project, we have decided to select and combine multiple programming languages to ensure the system is both flexible and efficient. Each part of the system is designed using a combination of suitable languages tailored to its specific role.

For the backend, we chose C# as the primary programming language, along with the integration of other languages such as Python and Java to handle particular tasks that require their respective strengths.

For the frontend, we utilized standard web development languages and frameworks, including HTML, Bootstrap, CSS, and JavaScript. Each language brings its own unique advantages to the development process — HTML defines the structure of the pages, Bootstrap ensures responsive design, CSS handles the visual styling, while JavaScript adds dynamic and interactive features.

The combination of these technologies allows us to maximize the performance, usability, and scalability of the system, ultimately aiming to deliver a well-structured and efficient product.

### 3.1.1. Frontend technologies

HTML (HyperText Markup Language) is the fundamental building block of web development, responsible for defining the structure and content of web pages. As the standard markup language for creating websites, HTML provides a clear and organized way to arrange elements such as text, images, links, forms, and other multimedia components. Its simplicity and compatibility with all web browsers make HTML an essential part of any web-based project.

CSS (Cascading Style Sheets), on the other hand, is used to control the visual presentation and layout of HTML elements. CSS allows developers to design attractive and responsive user interfaces by defining styles for fonts, colors, spacing, positioning, and animations. By separating the content structure (HTML) from the visual design (CSS), web applications can maintain a clean architecture and improve both flexibility and maintainability.

In this project, the combination of HTML and CSS plays a crucial role in building a clear, intuitive, and user-friendly interface, which enhances the overall user experience and supports modern design principles.



Fig 3.1. HTML & CSS

### 3.1.2. Backend technologies

In order to ensure the stability, scalability, and maintainability of the Event System Website, selecting an appropriate programming language and development framework is a critical step in the development process. After carefully evaluating various available technologies, the project team decided to use C# combined with ASP.NET Core MVC as the primary programming language and framework.

C# is a modern, object-oriented, and versatile programming language developed by Microsoft, which is widely used for building secure and high-performance web applications. Alongside C#, the ASP.NET Core MVC framework offers a powerful and flexible architecture that supports the separation of concerns, promotes code reusability, and simplifies the process of maintaining and expanding the system over time.

Moreover, ASP.NET Core MVC allows the project to leverage modern development practices, including cross-platform compatibility, cloud readiness, strong security mechanisms, and excellent integration with databases and third-party services. These advantages make it an ideal choice for developing an enterprise-level web application like the a digital librrary and research center website.



Fig 3.2. ASP.NET

### 3.1.3. Database management

Microsoft SQL Server is a proprietary [relational database management system](https://en.wikipedia.org/wiki/Relational_database_management_system) developed by [Microsoft](https://en.wikipedia.org/wiki/Microsoft) using [Structured Query Language](https://en.wikipedia.org/wiki/SQL) (SQL, often pronounced "sequel"). As a [database server](https://en.wikipedia.org/wiki/Database_server), it is a [software product](https://en.wikipedia.org/wiki/Software_product) with the primary function of storing and retrieving data as requested by other [software applications](https://en.wikipedia.org/wiki/Software_application)—which may run either on the same computer or on another computer across a network (including the Internet). Microsoft markets at least a dozen different editions of Microsoft SQL Server, aimed at different audiences and for workloads ranging from small single-machine applications to large Internet-facing applications with many [concurrent users](https://en.wikipedia.org/wiki/Concurrent_user).



Fig 3.3. SQL Server

## 3.2. Project management on Trello

Trello is a visual project management tool that helps organize work using boards, lists, and cards. The team created a Trello for the project with “To Do,” “Doing,” and “Done” boards for each sprint. Each task card contains specific information about the task name, responsible person, due date, and detailed description.

Work progress is marked with colors: green (completed), yellow (in progress), orange (to be done). Members attach evidence and work results to the comment section to check and evaluate effectiveness. The system also supports notification one day in advance when the deadline is approaching, helping to ensure timely completion.

Ảnh có chứa văn bản, phần mềm, Biểu tượng máy tính, Trang web

Nội dung do AI tạo ra có thể không chính xác.

Fig 3.4. Trello project management tool

### 3.2.1. Sprint 1: Login and register form

- Time: 12/03/2025 – 20/03/2025

- Objective: Build the registration, login, and user profile setup functions.

- Tasks:

+ Develop the interface and handle logic for login, registration, and password recovery.

+ Add security CAPTCHA.

+ Create a form to display and edit personal profile information.

+ Test basic functions and deploy to the testing environment.

### 3.2.2. Sprint 2: Admin form

- Time: 20/03/2025 – 28/03/2025  
- Objective: Implement a management system for senior administrators.  
- Tasks:

+ User management.

+ Digital document management (add, edit, delete).

+ Review and manage scientific research, comments, and evaluations.

+ Data statistics, management of books & research resources.

+ Publisher management.

### 3.2.3. Sprint 3: User form

- Time: 28/03/2025 – 04/04/2025  
- Objective: Build key functionalities for general users.  
- Tasks:

+ View personal account information.

+ Manage post submissions, book uploads, and research papers.

+ Submit and view posts.

+ Submit and track research paper status.

### 3.2.4. Sprint 4: Advanced Features

- Time: 04/04/2025 – 12/04/2025  
- Objective: Integrate advanced features using modern technologies.  
- Tasks:

+ Integrate a chatbot to assist users.

+ Implement Google Sign-In.

+ Manage account upgrades (e.g., researchers, lecturers).

+ Summarize document content using AI.

### 3.2.5. Sprint 5: Testing and deployment

- Time: 12/04/2025 – 20/04/2025  
- Objective: Ensure system stability and readiness for deployment.  
- Tasks:

+ Write and execute test cases (unit and integration testing).

+ Conduct performance and security testing.

+ Prepare and deploy the system in a test environment.

### 3.2.6. Sprint 6: Maintenance and Optimization

- Time: 20/04/2025 – 30/04/2025  
- Objective: Improve performance and extend system capabilities.  
- Tasks:

+ Fix bugs and update UI/UX components.

+ Optimize overall system performance.

+ Analyze user feedback from testing and make adjustments.

### 3.2.7. Sprint 7: Reporting and Project Wrap-up

- Time: 30/04/2025 – 26/05/2025  
- Objective: Finalize reports, present results, and deliver the project.  
- Tasks:

+ Prepare and write the final project report.

+ Create presentation slides.

+ Deliver the final product and related documentation.

+ Review, evaluate outcomes, and close the project.

## 3.3. Interface design

### 3.3.1. User Interface

One of the most essential components that enables users to interact effectively with a system is the User Interface (UI). The UI serves as a visual bridge between the user and the underlying functionalities of the system, providing users with a clear and intuitive overview of how the system operates. A well-designed user interface not only enhances the aesthetic appeal of the application but also plays a critical role in guiding users through their tasks in a smooth and efficient manner. By offering an accessible and user-friendly environment, the interface helps minimize the learning curve, reduces errors, and ultimately contributes to a more satisfying user experience.

### 3.3.1.1. Login User

The Login function allows users who already have an account to access the digital library system and research center. When accessing the login page, users will enter their login name (or email) and password into the provided form. The login interface is designed to be simple and easy to use, including necessary information fields and support links such as "Forgot password?", helping users to easily operate right from the first use.

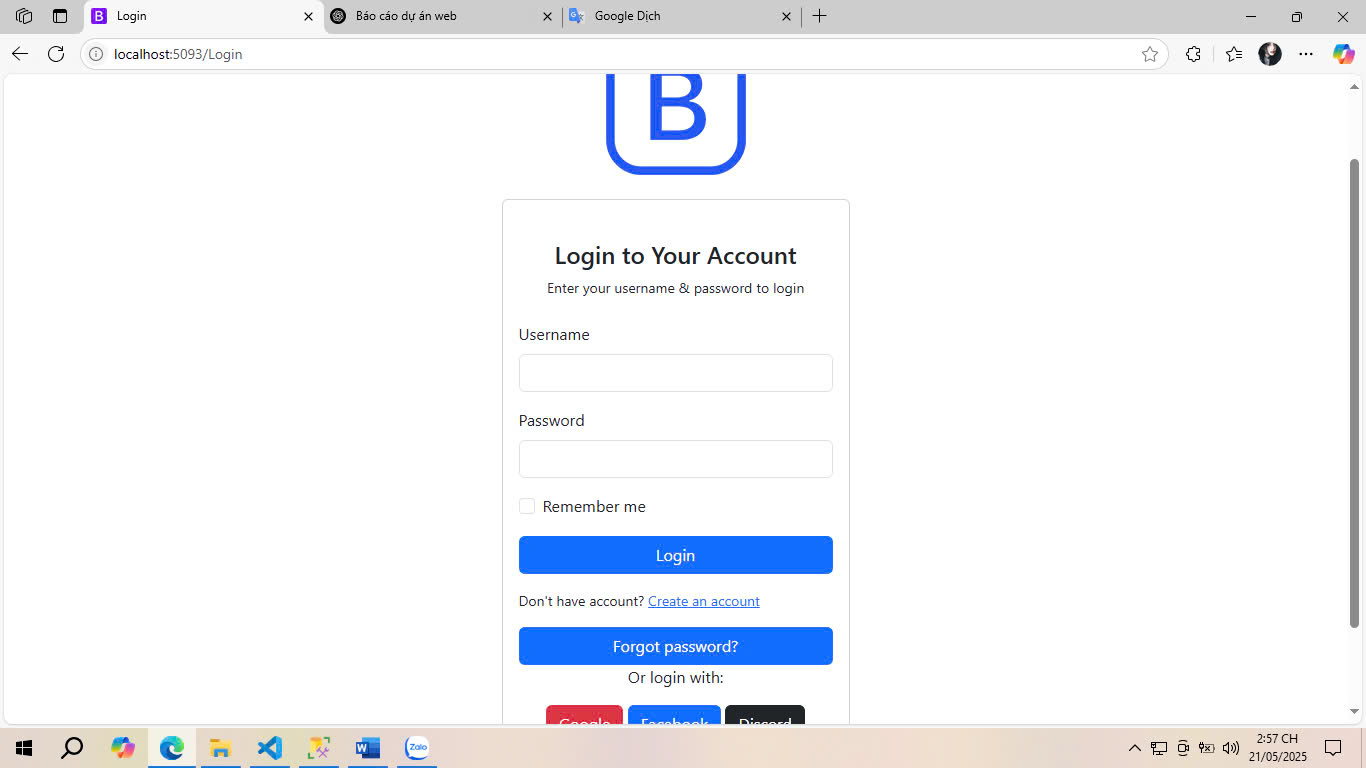


Fig 3.5. Login Interface

### 3.3.1.2. Forgot Password User

The Forgot Password function helps users recover their passwords in case they cannot log in to the system. When selecting this function on the login interface, users will be asked to enter personal information such as email or previously registered username. The system will check the data and, if valid, send password reset instructions via email or display a success message. If the information is incorrect or does not exist in the system, the user will receive an error message and be asked to re-enter. The Forgot Password function interface is designed to be simple and easy to operate, helping to ensure a convenient and safe user experience during the account recovery process.

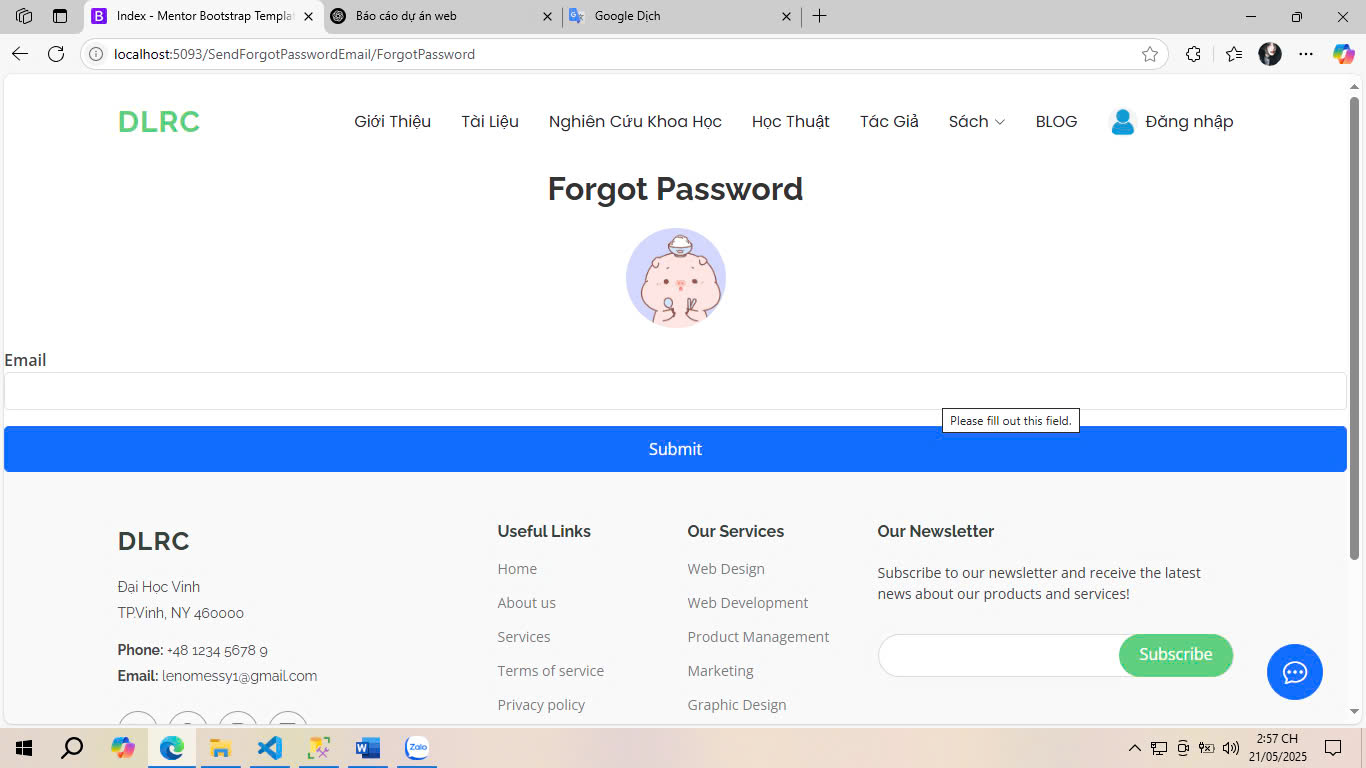


Fig 3.6. Forgot Password Interface

### 3.3.1.3. Register Interface

The registration function enables users to create a new account in order to access and utilize the digital library and research center system. Upon navigating to the registration page, users are required to complete a form with essential information such as full name, email address, password, and other personal details.

Once the user submits the form, the system will validate the entered information. If all data is valid and no duplication is found (e.g., existing email), the system will proceed to create a new account and notify the user of successful registration.

In cases where required fields are missing or the provided data already exists in the system (such as a duplicate email), an error message will be displayed. The system will prompt the user to review and update the form with the correct or missing information before resubmitting.

This ensures that only complete and unique user profiles are created in the system, maintaining data integrity and a smooth onboarding experience.

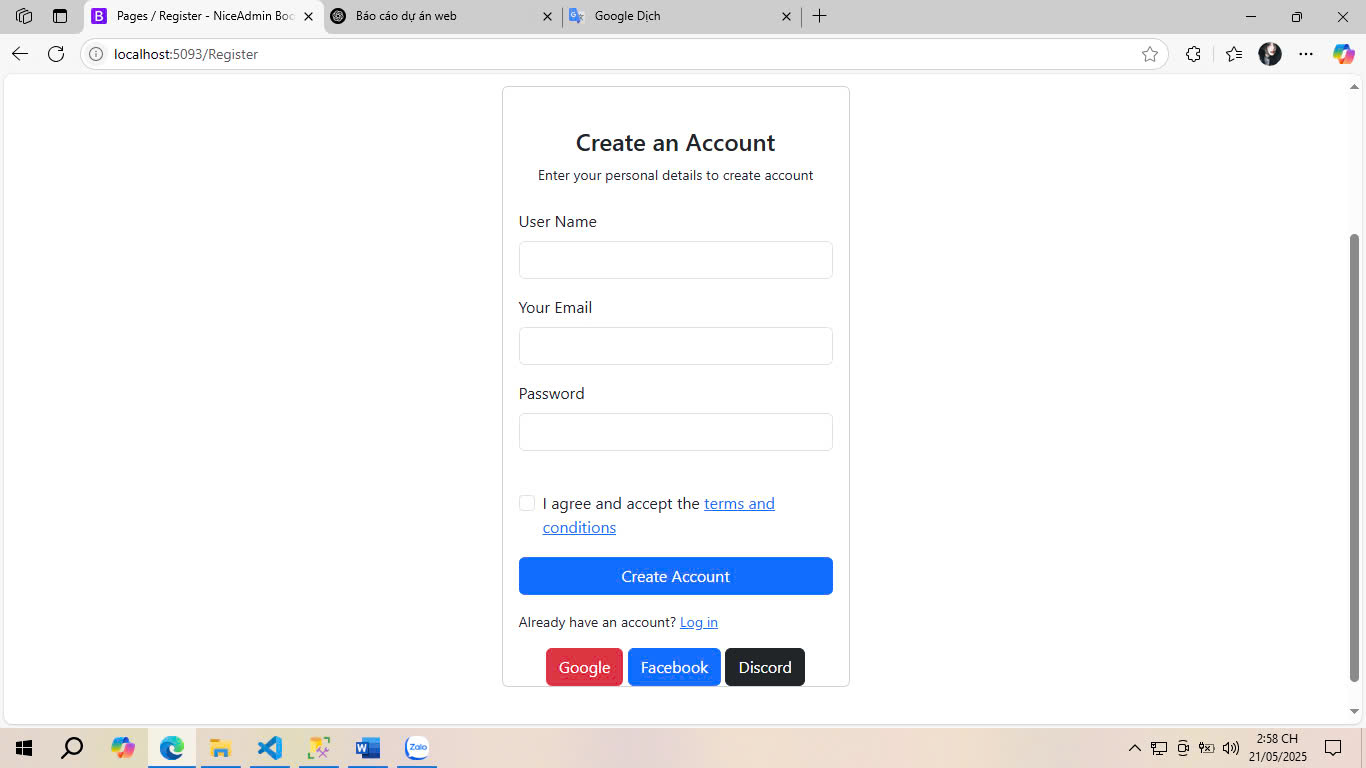


Fig 3.7. Register Interface

### 3.3.1.4. User Page Interface

The user interface serves as the main navigation center for system users, especially students, lecturers and researchers. This is where users can access key functions such as viewing personal information, posting books or research papers, managing posted papers, rating, commenting and saving favorite papers.

The interface is designed with an intuitive, friendly and easy-to-use layout, helping users easily perform necessary operations right from the first access. The navigation bar is placed in an easy-to-see position, allowing quick access to subsystems such as "My Documents", "My Research", "Post Books", "Update Profile", or "Log Out". In addition, the system also integrates an advanced search function, allowing filtering by criteria such as author name, year of publication, document type, etc., helping users easily find suitable resources.

In addition, the interface also displays a list of documents that users have posted or marked, along with information such as approval status, number of views, and average rating. From here, users can edit, update, or delete documents if necessary. This interface also allows users to track feedback from the community through comments and rating systems.

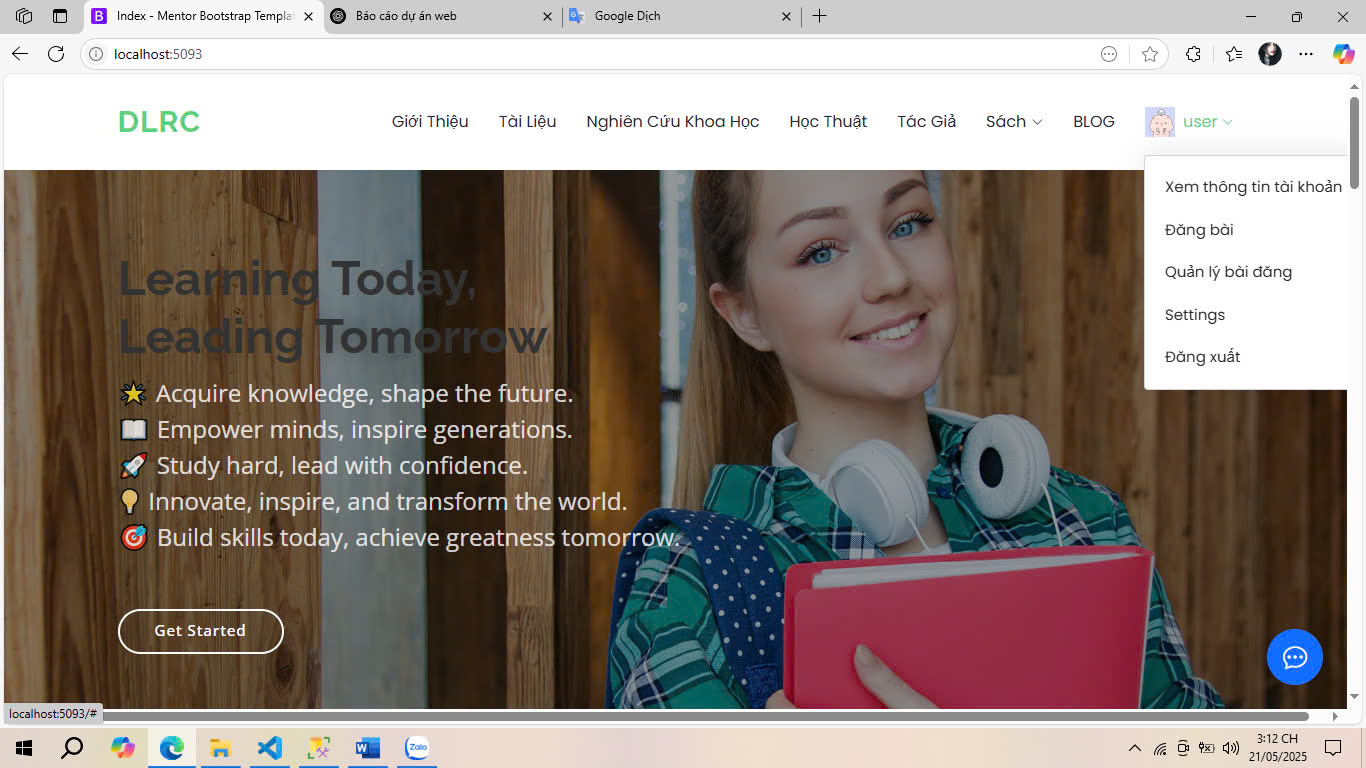


Fig 3.8. User Page Interface

### 3.3.1.5. User Account Details Interface

The user account details interface is where all the personal information provided by the user is displayed when registering an account or during the use of the system. This interface allows users to check and update information such as full name, date of birth, gender, address, phone number, avatar, and email address. All data fields are presented clearly and easily, making it easy for users to verify and edit when necessary.

The interface is designed in a friendly and simple way, using forms with corresponding input boxes, along with function buttons such as "Save changes", "Cancel", or "Change password". In addition, the system also integrates advanced security features such as two-factor authentication (2FA), the option to turn on/off security alerts, and display the account's active status (active or locked).

Editing personal information is processed through layers of data validation to ensure validity and security, such as checking email format, password strength, or input length limits. In case the user enters incorrect or duplicate information, the system will issue a warning and instructions for adjustment.

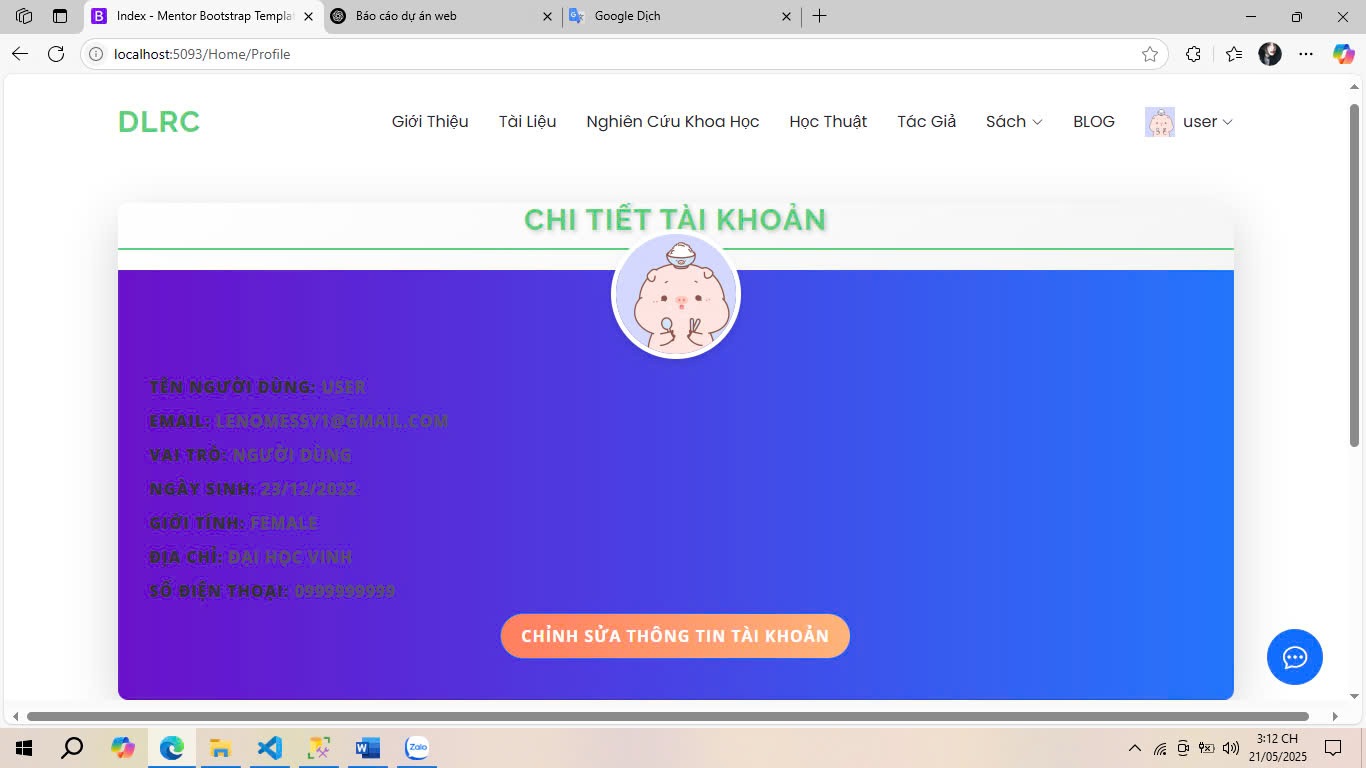


Fig 3.9. User Account Details Interface

### 3.3.1.6. User Book Registration Interface

The user's book posting interface is where system users can upload documents or e-books to the digital library. This interface is designed as an information filling form, in which users need to provide all necessary data fields such as: book title, author name, year of publication, genre, number of pages, cover photo, content description, and attached file (PDF or supported format).

The interface is optimized for user experience, with a clear layout, marked required fields, and specific instructions to support users in entering data accurately. Tools to check file format, maximum size, and description content are also integrated to ensure consistency and quality of posted documents.

After completing the form, users can click the "Submit book" button for the system to record and transfer the document to the administrator for review. In case of incomplete information or invalid file, the system will give specific error message, helping users to easily adjust.

In addition, this interface also has a draft saving feature that allows users to temporarily save the content they are entering and complete it later. This helps to minimize the risk of data loss due to technical problems or sudden exit of the page.

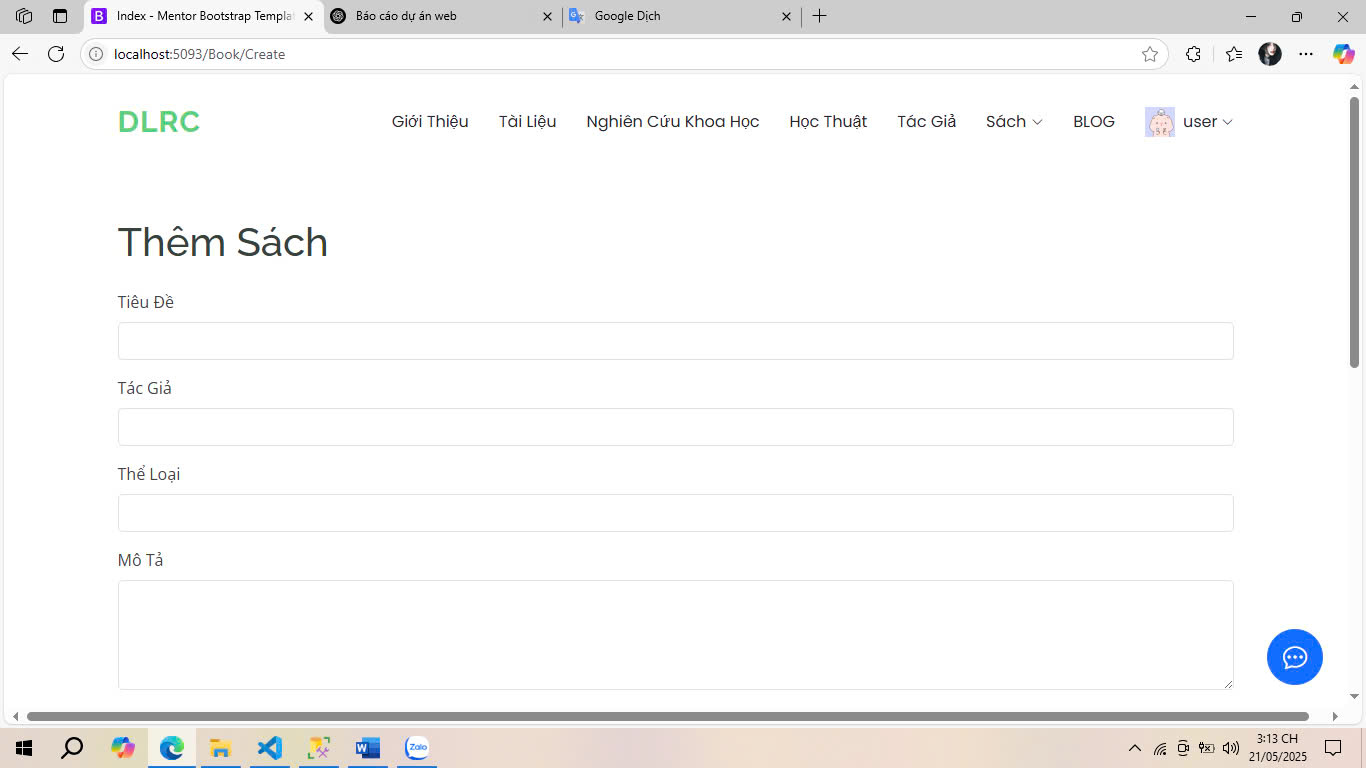


Fig 3.10. User Book Registration Interface

### 3.3.1.7. User Book Management Interface

The user's book management interface is where all documents, books or textbooks that users have uploaded to the digital library system are centralized. This interface provides a comprehensive overview of each uploaded book, including information such as title, approval status (approved, pending approval, rejected), views, average rating, and posting time.

At this interface, users can perform many management operations such as: editing book information, updating files, changing cover images, deleting books from the system, or viewing rejection reasons if the document is not approved by the administrator. The functions are displayed through clear icons or action buttons such as "Edit", "Delete", "View details", making it easy for users to operate.

The system also supports filtering and searching by criteria such as title, genre, or approval status, to help users quickly find documents that need to be edited or checked. In addition, in case the document is rejected, the system will display the specific reason provided by the administrator, thereby helping users know the content that needs to be adjusted to be re-approved.

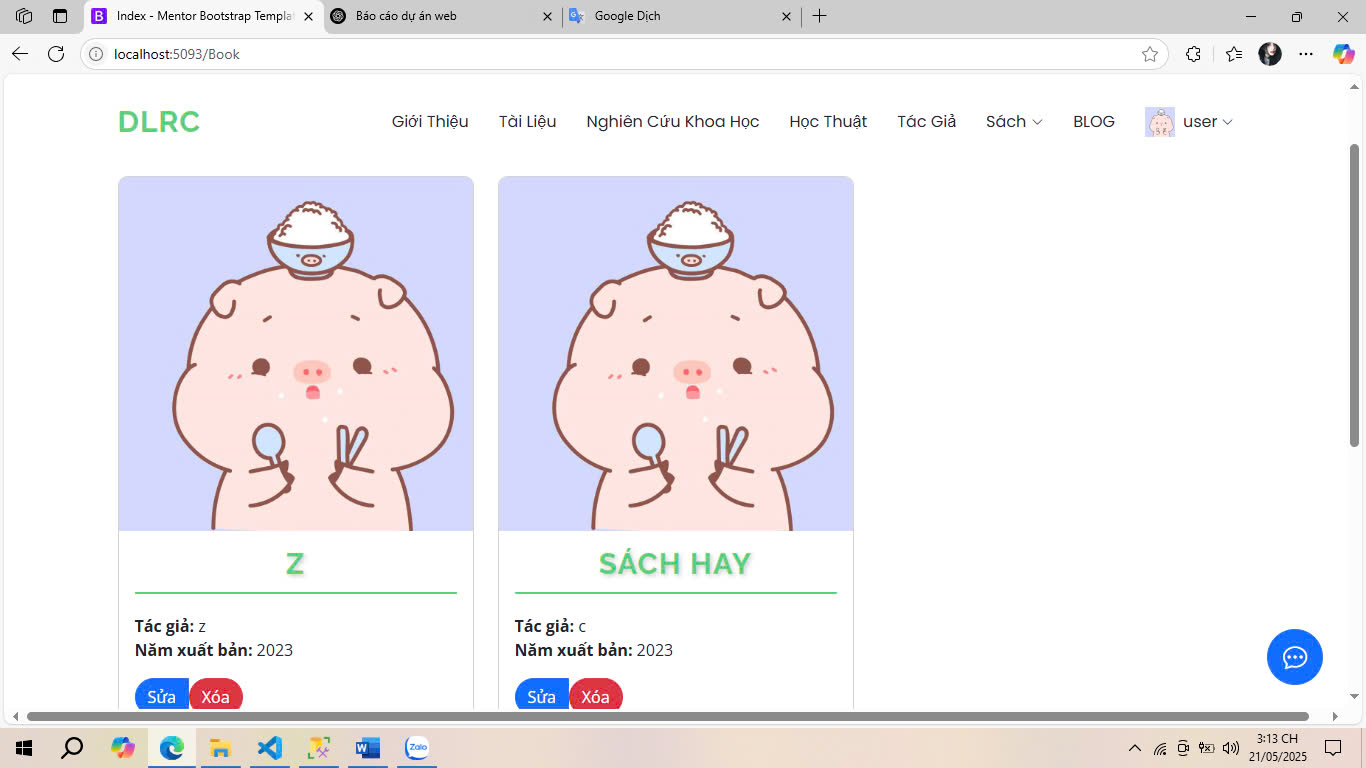


Fig 3.11. User Book Management Interface

### 3.3.1.8. User Research Management Interface

The user research management interface is a tool that supports users in tracking, editing, and controlling academic research articles posted to the system. Similar to the book management interface, this interface displays a list of research articles that users have submitted, along with important information such as article title, research field, year of publication, confirmation status (confirmed, pending approval, rejected), and posting time.

Users can easily perform operations such as editing the content of the research article, updating attached files, deleting articles, or viewing detailed feedback from the administrator. If the research article is rejected, the interface also displays the reason for rejection and instructions for adjustment so that users can resubmit more accurately and completely.

The interface also supports filtering and searching by field, year of publication, or article status, helping users easily manage a large number of documents. Action buttons are arranged logically next to each article, with intuitive icons such as “Edit”, “View details”, “Delete”, to enhance the user experience.

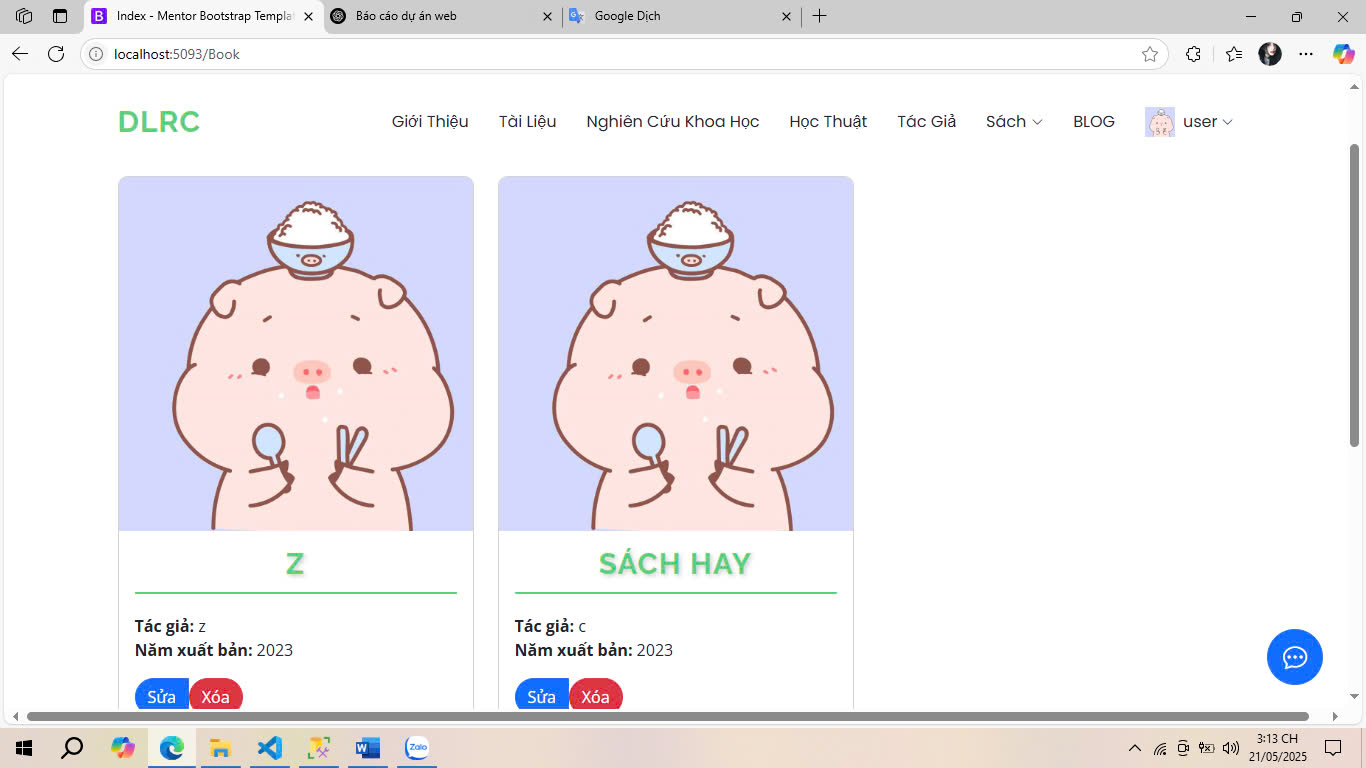


Fig 3.12. User Research Management Interface

### 3.3.2. Admin interface

Similarly to the user interface, the admin interface plays a vital role in the overall architecture of the system, particularly from the standpoint of system administrators and managers. It is specifically designed to provide administrators with comprehensive tools and controls to efficiently manage, monitor, and configure various components of the platform.

A well-structured admin interface significantly enhances operational efficiency by streamlining administrative workflows. It enables administrators to perform critical tasks such as managing user accounts, handling digital document submissions, moderating content and user feedback, assigning roles and permissions, and overseeing system performance and security.

Moreover, the admin interface serves as the central hub for content management, including the approval or rejection of research papers, book submissions, and comments. It also allows administrators to generate reports, track usage statistics, and make data-driven decisions that align with organizational goals.

In addition to functionality, the interface emphasizes usability and accessibility, ensuring that system managers can easily navigate and utilize its features without requiring deep technical knowledge. This reduces the likelihood of errors and improves response time in addressing system issues.

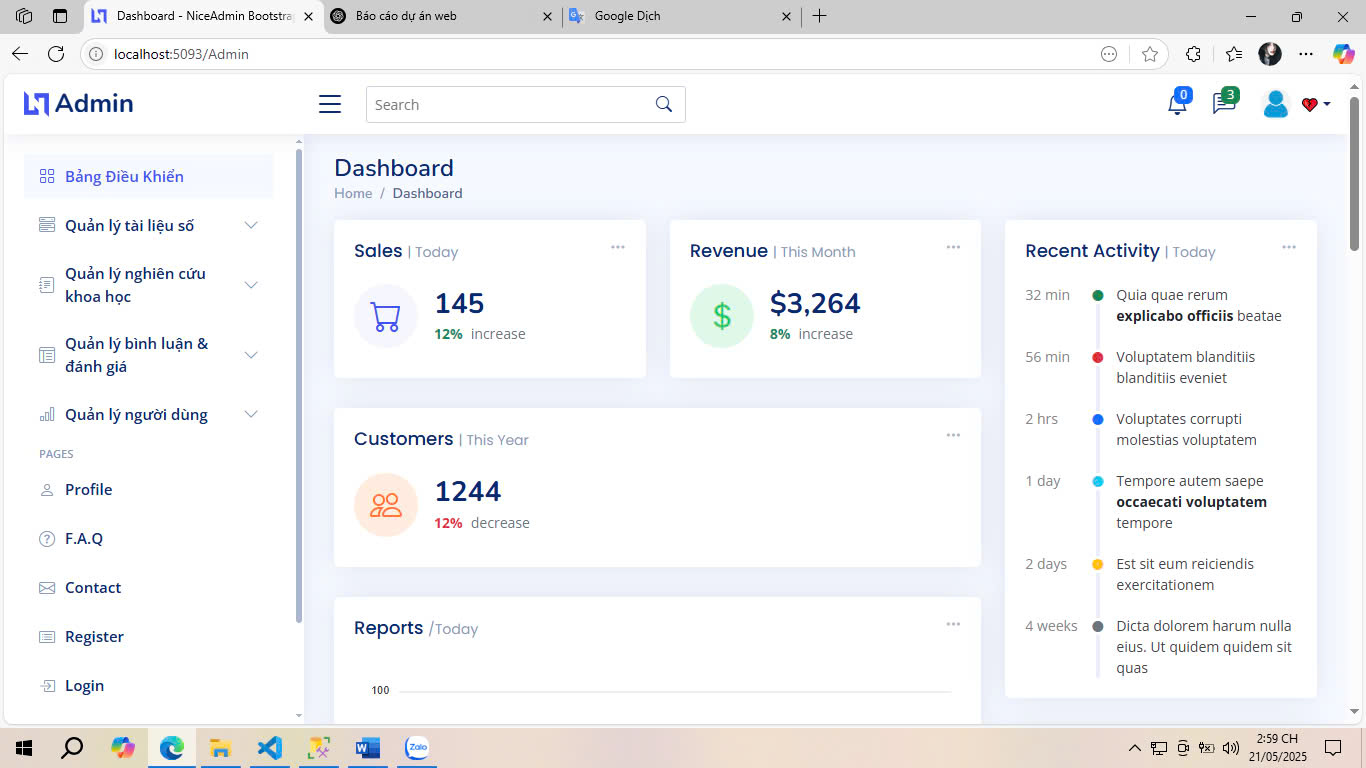


Fig 3.13. Admin Interface

### 3.3.2.1. Admin Profile Interface

The administrator profile interface is where the personal information of users belonging to the system administrator group is displayed and updated. This is an important part of the account management module, helping to ensure that the administrator's information is always fully updated, accurate and secure.

This interface displays information fields such as full name, date of birth, gender, email, phone number, address, and avatar. In addition, the system also displays the account role (admin), activity status, and security settings such as two-factor authentication (2FA) and security alert options. The information is displayed in a clear, organized layout, making it easy for administrators to monitor and edit when necessary.

The editing function is integrated directly on the interface through the "Edit profile" or "Update information" button, allowing administrators to change information related to themselves. All updates are accompanied by input validation checks to avoid data errors and ensure system consistency.

The administrator profile interface not only serves the purpose of updating personal information but also acts as an additional layer of security, contributing to ensuring the stable and safe operation of the entire digital library management system.

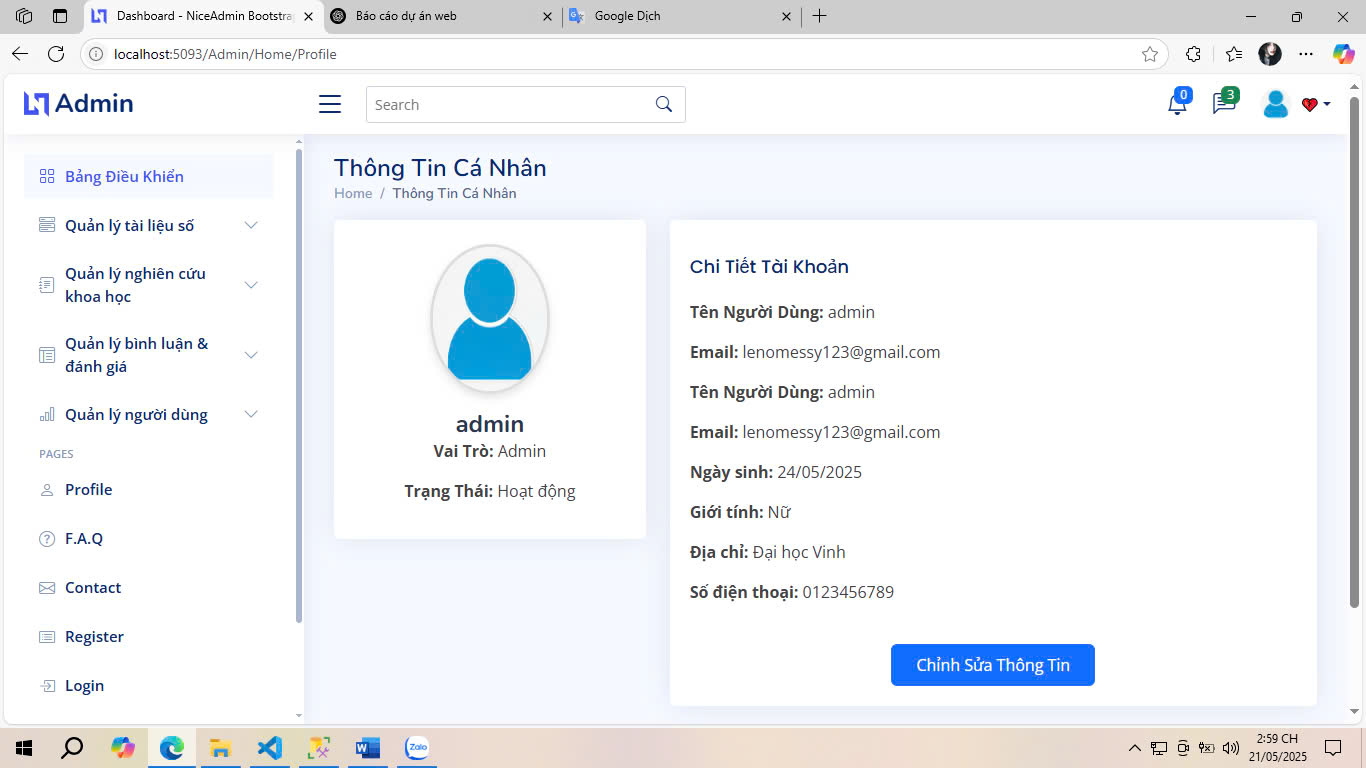


Fig 3.14. Admin Profile Interface

### 3.3.2.2. Admin Edit Profile Interface

The administrator profile editing interface allows administrators to proactively and securely update personal information and security settings. The interface is designed to be intuitive, clean, and user-friendly, including input forms corresponding to information fields such as full name, date of birth, address, phone number, gender, and profile picture.

In addition to editing basic information, the interface also allows administrators to change the login email address and update the current password through a secure verification process. Users can also optionally enable or disable advanced security features such as two-factor authentication (2FA), receive unusual login alerts, and manage recent verification tokens.

The system integrates input data checks to ensure that updated information is in a valid format and is not duplicated. For example, the email must be in the correct format, the phone number must be valid, and the new password must be strong enough. If an error occurs, the system will provide a clear notification to help users easily adjust.

The “Save Changes” button is prominently placed, allowing administrators to confirm the update after completing the editing. A “Cancel” button is also provided to return to the profile interface without saving the changes.

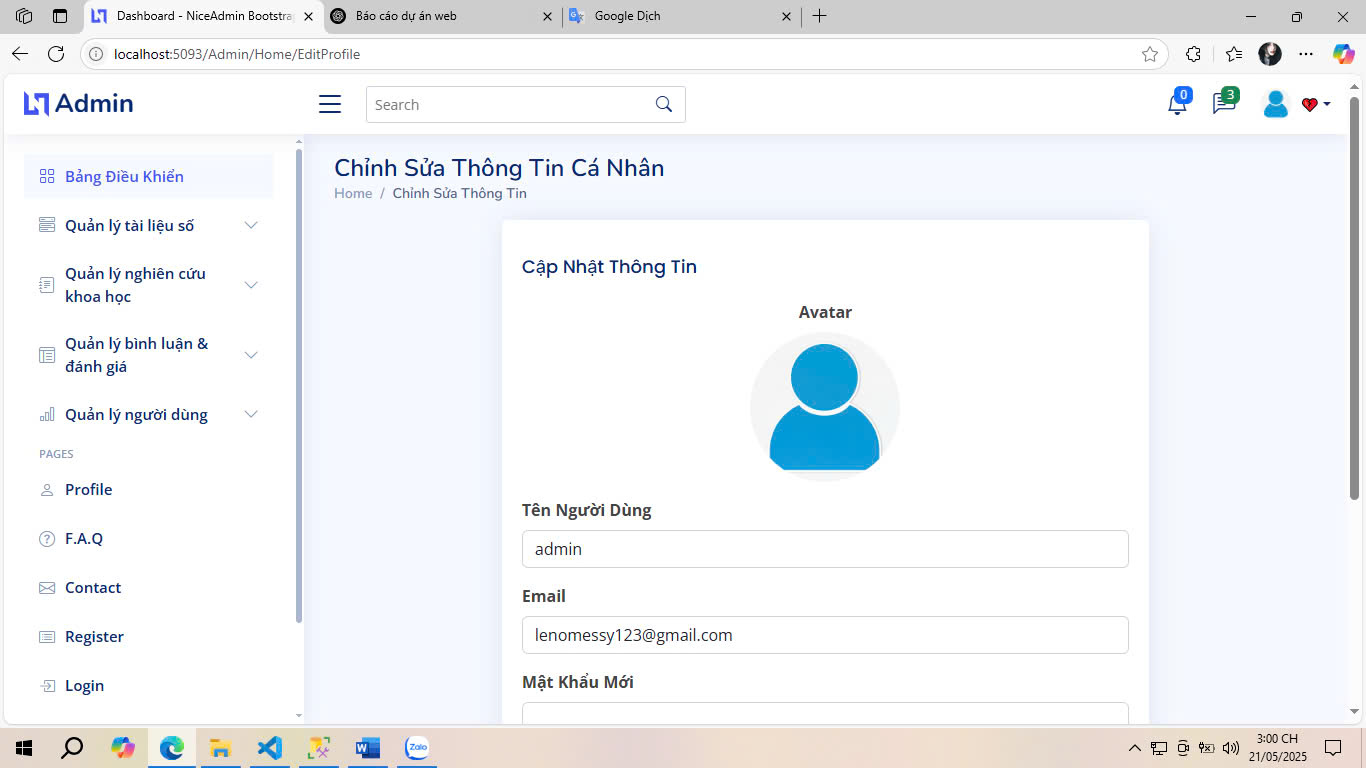


Fig 3.15. Admin Edit Profile Interface

### 3.3.2.3. Admin Need Help Interface

The Admin Need Help Interface is an important support component in the system, providing a quick communication channel between the administrator and the technical support department or system manual. This interface is designed to support the administrator when encountering technical problems, needing feature advice, or needing to report errors during the operation of the digital library system.

The interface includes an information filling form with fields such as request title, detailed content, priority level (low, medium, high), and the option to attach images or files illustrating the problem. The interface also allows users to choose the type of support needed, such as “Technical issues”, “Account & access”, or “System feedback”.

After submitting a request, the system will generate a support ticket code and display the processing status (pending, processing, responded). Administrators can track the progress of responses, send additional information, or mark requests as resolved. Additionally, if the system has a chatbot or a knowledge base (FAQ), the interface will automatically suggest common answers based on the content being entered.

With a friendly design, easy to use, and quick response, this interface helps improve troubleshooting efficiency and improve the administrator's experience in operating the system. At the same time, it acts as a bridge between administrators and the development team, contributing to maintaining the stability and quality of the digital library platform.

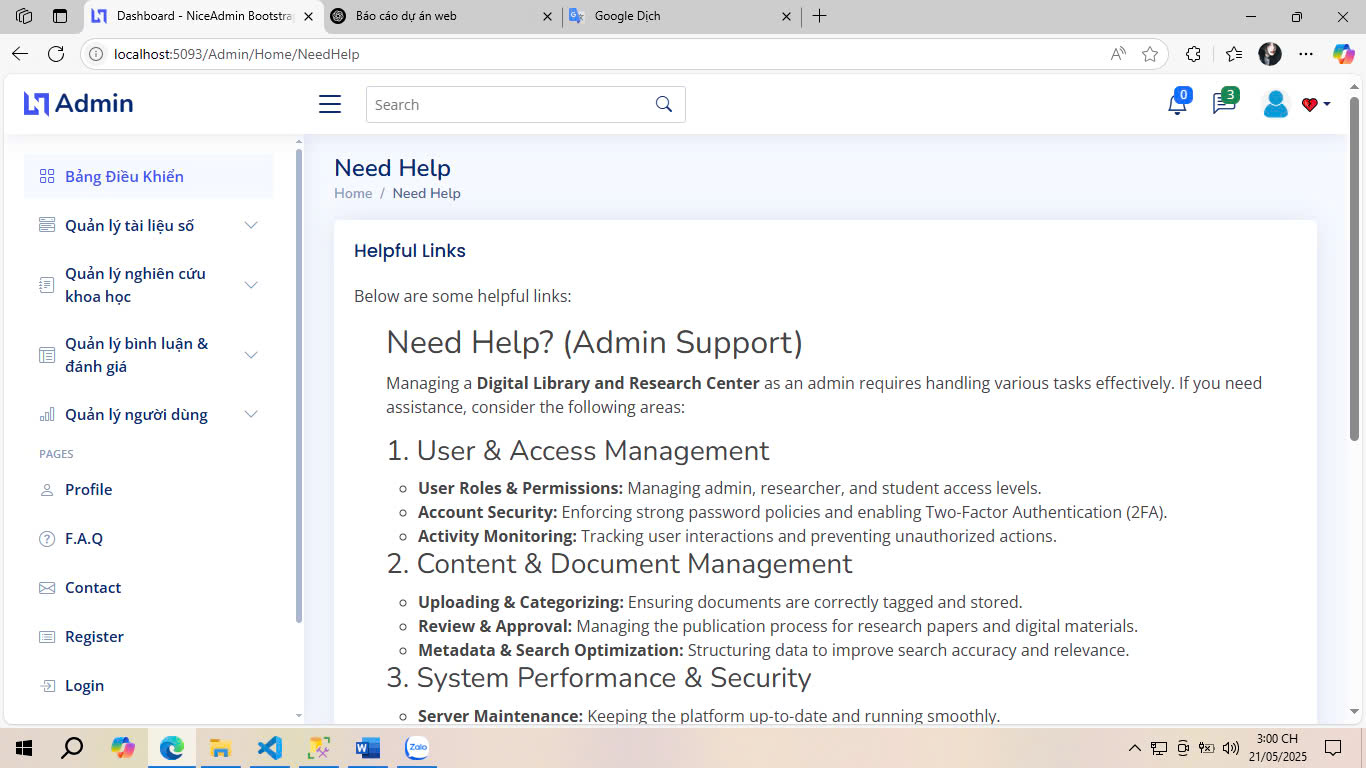


Fig 3.16. Admin Need Help Interface

### 3.3.2.4. Admin Document Management Interface

The Administrator Document Management Interface is an important subsystem in the digital library system, allowing administrators to control, review, and process all documents (books, textbooks, research) uploaded by users to the platform. This interface provides a comprehensive view of all documents pending review, approved, or rejected, along with detailed information such as document name, author, poster, submission date, status, number of views, and ratings.

Documents in the interface are presented in a paginated list format, with the ability to filter and search by criteria such as: title, poster name, year of publication, or approval status. Each document item comes with function buttons such as "View details", "Approve", "Reject", or "Delete", helping administrators operate quickly and accurately.

When viewing a document in detail, administrators can review the full content, view attachments, evaluate quality, and make a decision to approve or reject. In case of rejection, the system requires administrators to enter a reason to respond to the poster, ensuring transparency and building a fair review process.

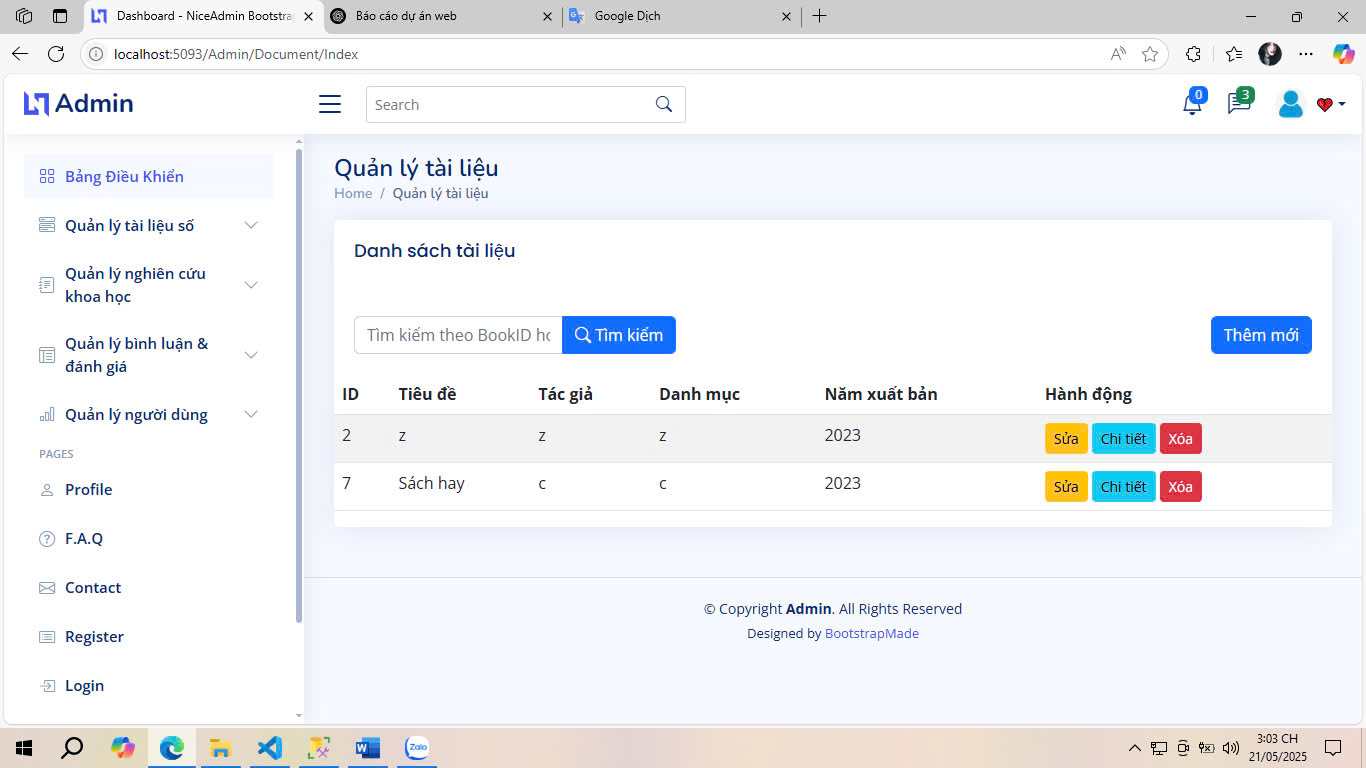


Fig 3.17. Admin Document Management Interface

### 3.3.2.5. Admin Document Details Interface

The Administrator Document Details interface is an extension of the document management system, allowing administrators to access and review specific information of each document uploaded by users. This interface plays a supporting role in making decisions to approve or reject documents based on a comprehensive assessment of content, quality, and appropriateness.

When accessing a specific document from the document list in the system, the interface will display full data fields including: title, author, poster, year of publication, genre, description, number of pages, views, attachments, cover image, and current status (approved, pending approval, rejected). In addition, if the document has been rejected before, the reason for rejection will also be displayed to ensure transparency in the review process.

This interface integrates action tools such as “Approve Document”, “Reject”, or “Send Feedback”. When rejecting, the administrator is required to enter the reason for rejection in a text box, which helps the poster understand the problem and edit the content if necessary.

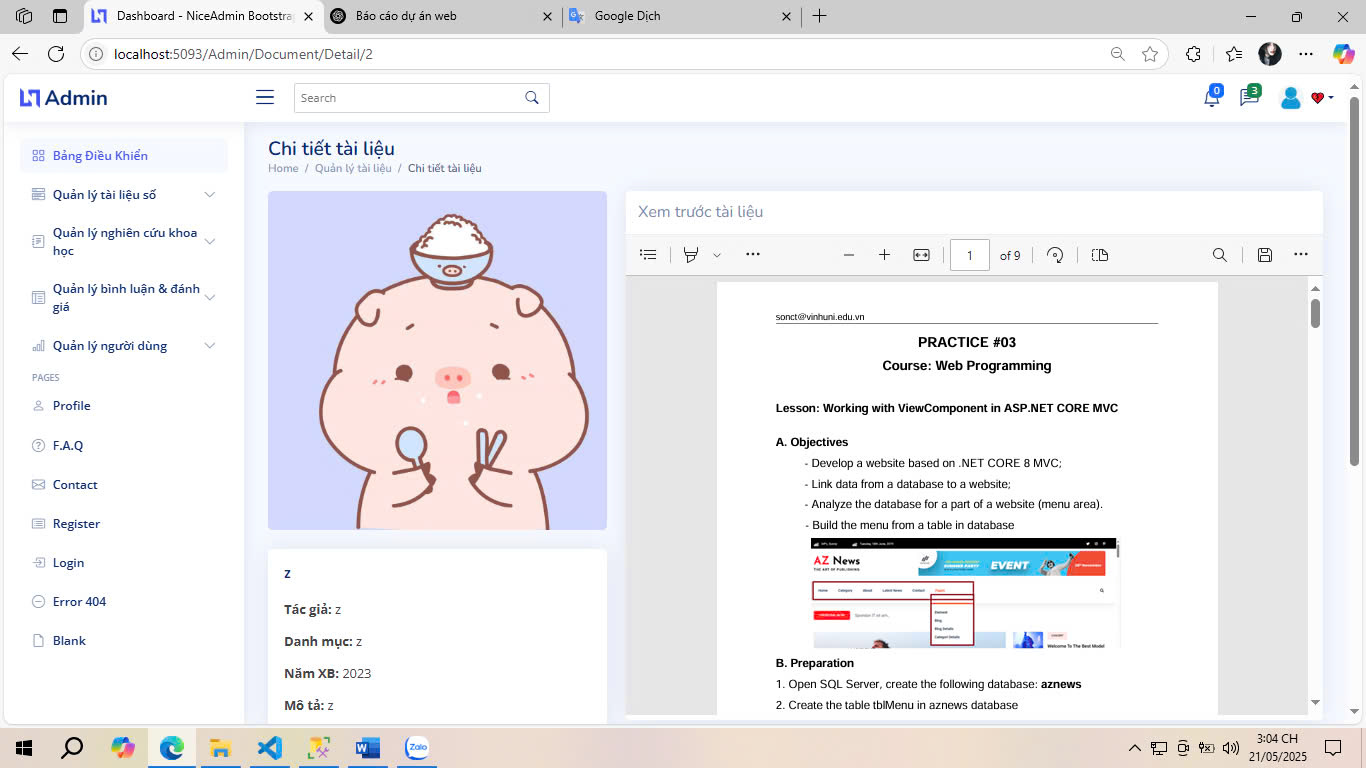


Fig 3.18. Admin Document Details Interface

### 3.3.2.6. Admin Review The Document Interface

The Administrator Document Review Interface is a place where tools and functions are centralized to serve the purpose of checking, evaluating and deciding to approve or reject an academic document posted by a user. This is an important intermediate step to ensure that all content published on the system meets the standards of quality, content and legality.

This interface provides a specialized working environment for administrators with a clear layout, including a document information display section (title, description, poster, attachments, genre, etc.) and a content review section. Administrators can preview document files directly (PDF, DOCX, etc.) through the built-in reader, supporting scrolling, zooming in/out and flexible page turning, making it convenient to read the full text of the document without downloading.

During the review process, administrators can leave internal comments, marking the status as “Waiting for more information”, “Needs editing”, or “Reviewed”, thereby supporting collaboration between multiple administrators if necessary. In addition, action buttons such as “Approve”, “Reject”, or “Request editing” are prominently placed, making the review process quick and efficient.

If the document is rejected, the interface requires detailed reasons to be entered, providing clear feedback to the poster, and storing the review data for future monitoring or re-evaluation purposes.

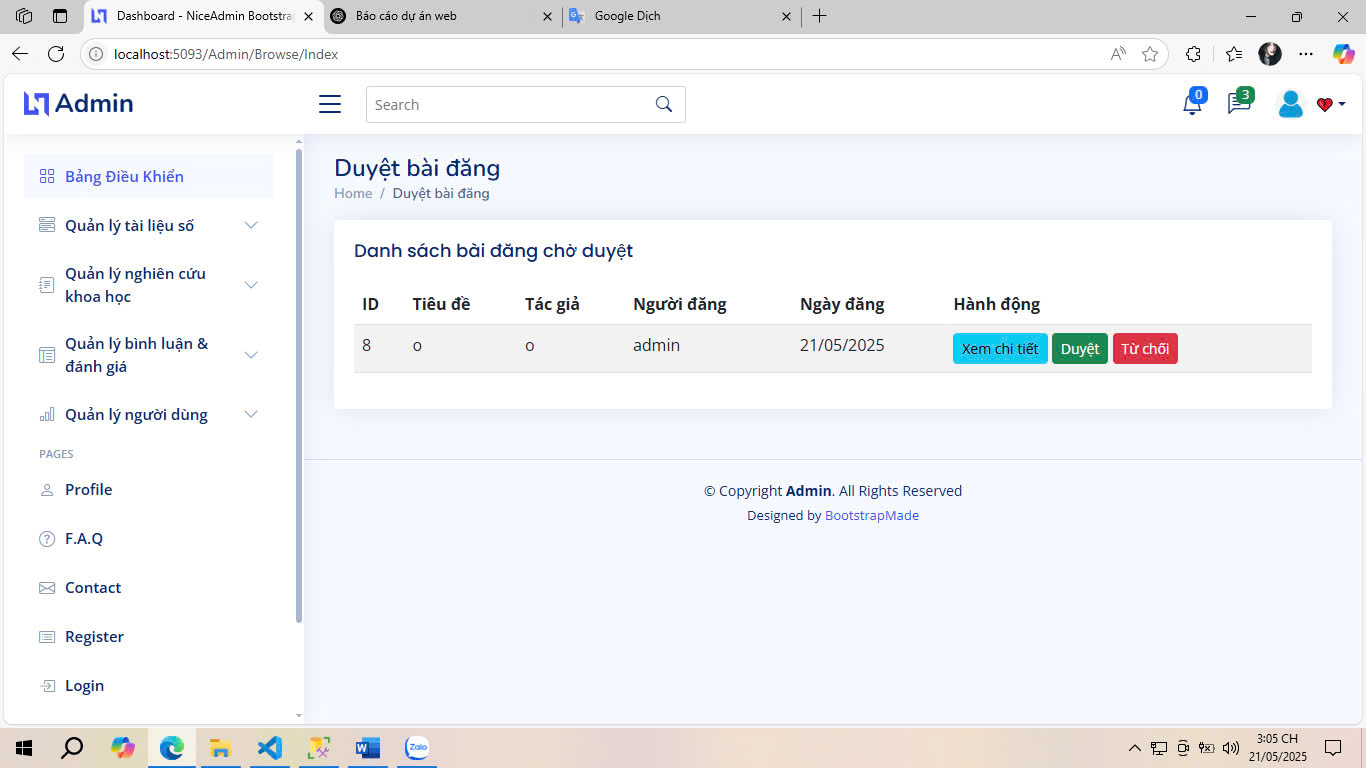


Fig 3.19. Admin Review The Document Interface

### 3.3.2.7. Admin Document Classification Interface

The Administrator Document Classification Interface is a function that supports the process of organizing and arranging documents scientifically in the digital library system. This interface allows administrators to assign documents to specific categories, topics, or fields to serve the end users in searching, filtering, and managing content more effectively.

On the interface, each document is displayed with basic information such as title, poster, date of submission, and brief description. Administrators can select one or more documents, then assign a main category (e.g. Information Technology, Social Sciences, Medicine, etc.) and a sub-category if any (e.g. Artificial Intelligence, Information Security, Database, etc.). The categories and classification labels are taken from the central classification management system to ensure uniformity.

The interface supports searching and filtering of documents by name, poster or upload date to help administrators work faster when working with large amounts of data. Each classification operation is saved in the system, allowing access to editing history when needed.

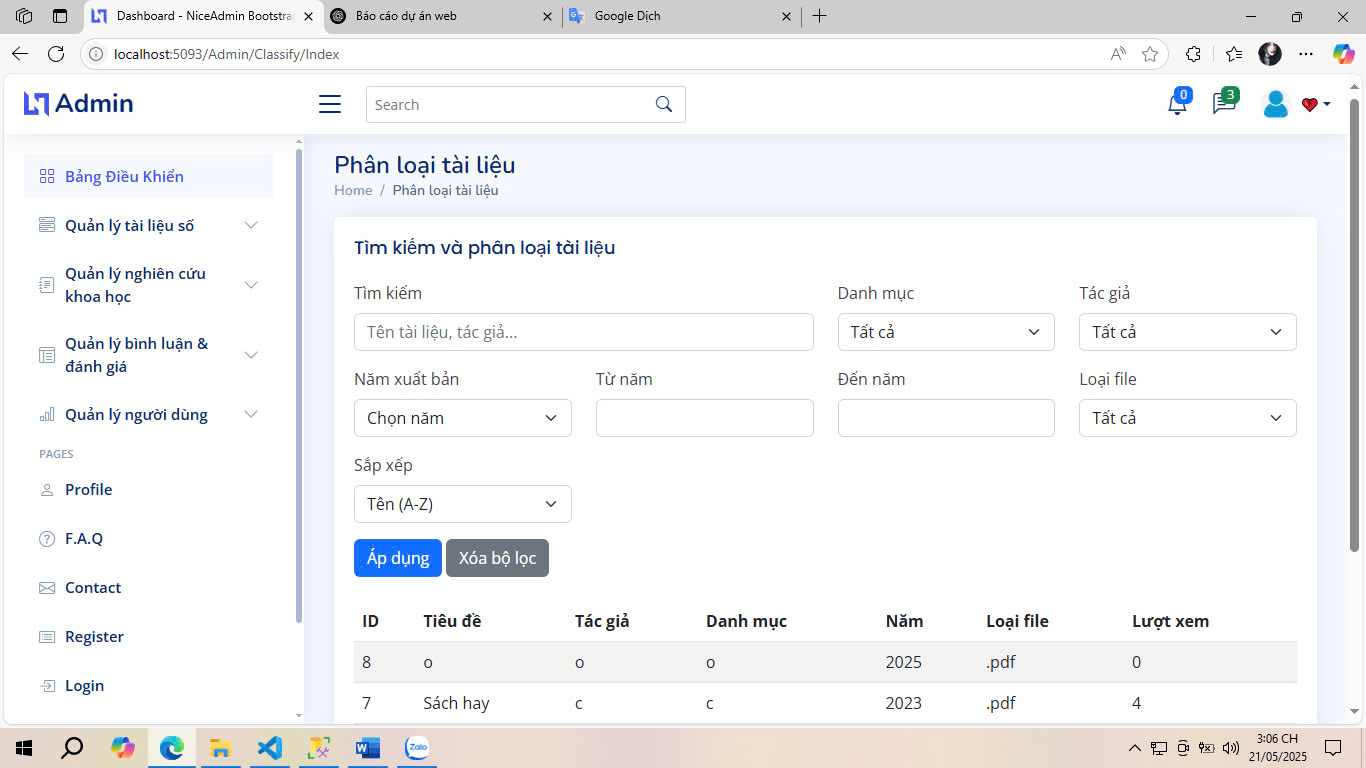


Fig 3.20. Admin Document Classification Interface

### 3.3.2.8. Admin Label The Document Interface

The Administrator Document Labeling interface is an additional support tool in the content management process, allowing administrators to assign custom labels to each document for more detailed classification purposes, serving advanced search features and document recommendation systems.

Unlike classifying documents by main categories, labeling allows for flexible use of keywords such as “AI”, “Big Data”, “Open Access”, “Thesis”, “Peer-reviewed”, etc. Each document can be assigned multiple labels at the same time, increasing the ability to retrieve and display when users search or filter by keywords.

The interface displays a list of documents with basic information such as title, poster, status, and assigned labels (if any). Administrators can add new labels by entering keywords in the search box, selecting suggested words, or creating new labels if they do not exist in the system. All labeling and de-labeling operations are processed in real time, ensuring efficiency and convenience in the management process.

In addition, the system can record the labeling history with operator information and update time for later checking and adjustment if necessary.

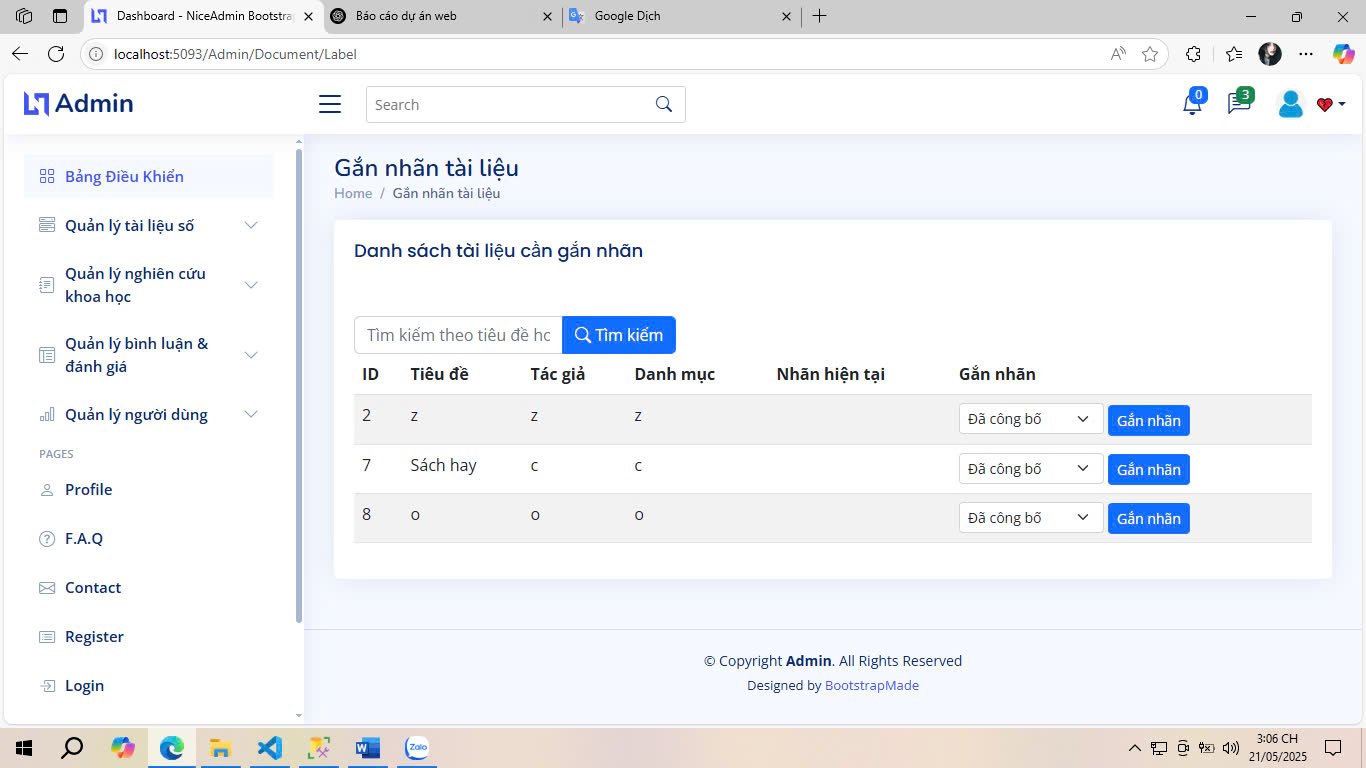


Fig 3.21. Admin Label The Document Interface

### 3.3.2.9. Admin Research Project Management Interface

The Administrator's Research Topic Management interface is the main component of the academic management system, allowing administrators to track, check, and process all research topics submitted by users to the system. This interface acts as a centralized dashboard to monitor the quality, processing progress, and validation status of academic research documents.

The interface displays a list of research topics with important information such as title, poster, research field, publication year, validation status (pending approval, validated, rejected), submission date, and engagement level (number of views, comments, ratings). Filtering tools by field, status, or time help administrators quickly process when there are many topics at the same time.

From this interface, administrators can perform operations such as viewing topic details, approving content, rejecting with specific reasons, or requesting users to edit and resubmit. In addition, the system also supports labeling and classifying research fields, serving the purpose of searching and organizing data effectively.

The interface also provides real-time quick reports, displaying the number of pending topics, confirmed topics, and rejected topics, helping administrators grasp the overall situation intuitively.

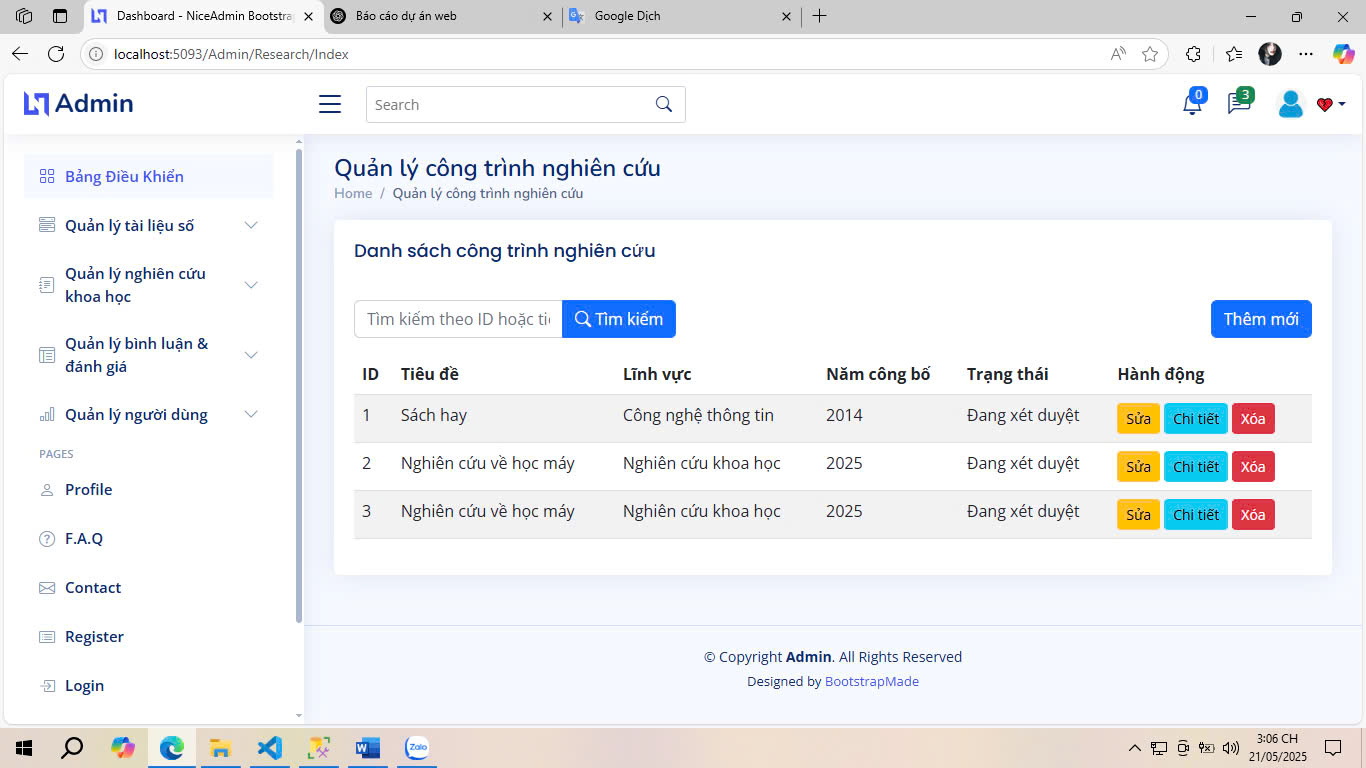


Fig 3.22. Admin Research Project Management Interface

### 3.3.2.10. Admin Research Project Details Interface

The Administrator's Research Topic Details interface provides the most complete and detailed information about a specific research topic posted by a user. This is a tool to support administrators in comprehensively evaluating academic content, making decisions to confirm, reject, or request editing of the topic.

When accessing a specific research topic, the interface will display key information fields such as topic title, research field, abstract, year of publication, submission date, current status, uploader, and attached document file. In addition, if the topic has been previously confirmed or rejected, the system also displays the review date, reviewer, and feedback notes (if any).

Administrators can view the entire content of the research file directly on the interface through the integrated preview function, saving processing time and eliminating the need to download the file. Action tools include approving a topic, rejecting a topic with a specific reason, or moving it to edit status, which are clearly laid out to support a quick and consistent review process.

The interface also links to additional functions such as viewing comments, ratings from other users, and research classification labels, giving administrators a more comprehensive view of the level of engagement and quality of content.

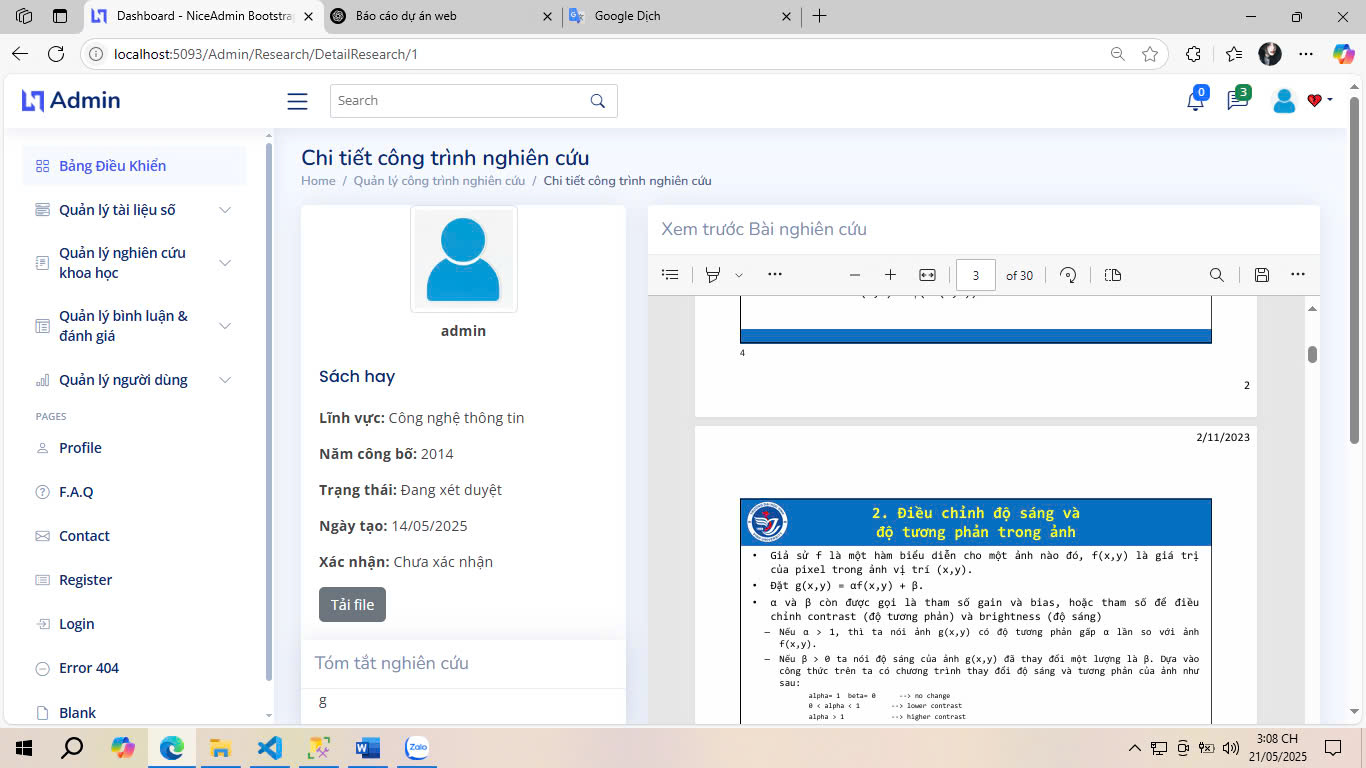


Fig 3.23. Admin Research Project Details Interface

### 3.3.2.11. Admin Comment Management Interface

The Administrator's Comment Management Interface is an important component that helps control user interactions on the digital library system, specifically comments sent about books and research topics. This interface allows administrators to monitor, evaluate, and process comments to ensure a healthy, objective, and moderated academic environment.

On the interface, a list of comments is displayed along with information such as: comment content, commenter name, posting time, related objects (book name or topic), number of likes/dislikes, and comment status (visible, hidden, reported). The interface provides filtering and searching functions by user, keyword, or posting date to facilitate processing of large volumes of comments.

Administrators can perform operations such as:

- Delete comments that violate or have inappropriate content.

- Temporarily hide comments from the system for further review.

- Send warnings or reminders to users if they violate the rules.

- View detailed interaction history such as likes, violation reports, or other responses related to that comment.

The system also supports aggregate statistics, helping administrators grasp the number of comments per day, the rate of hidden or reported comments, and identify users with inappropriate behavior in community interactions.

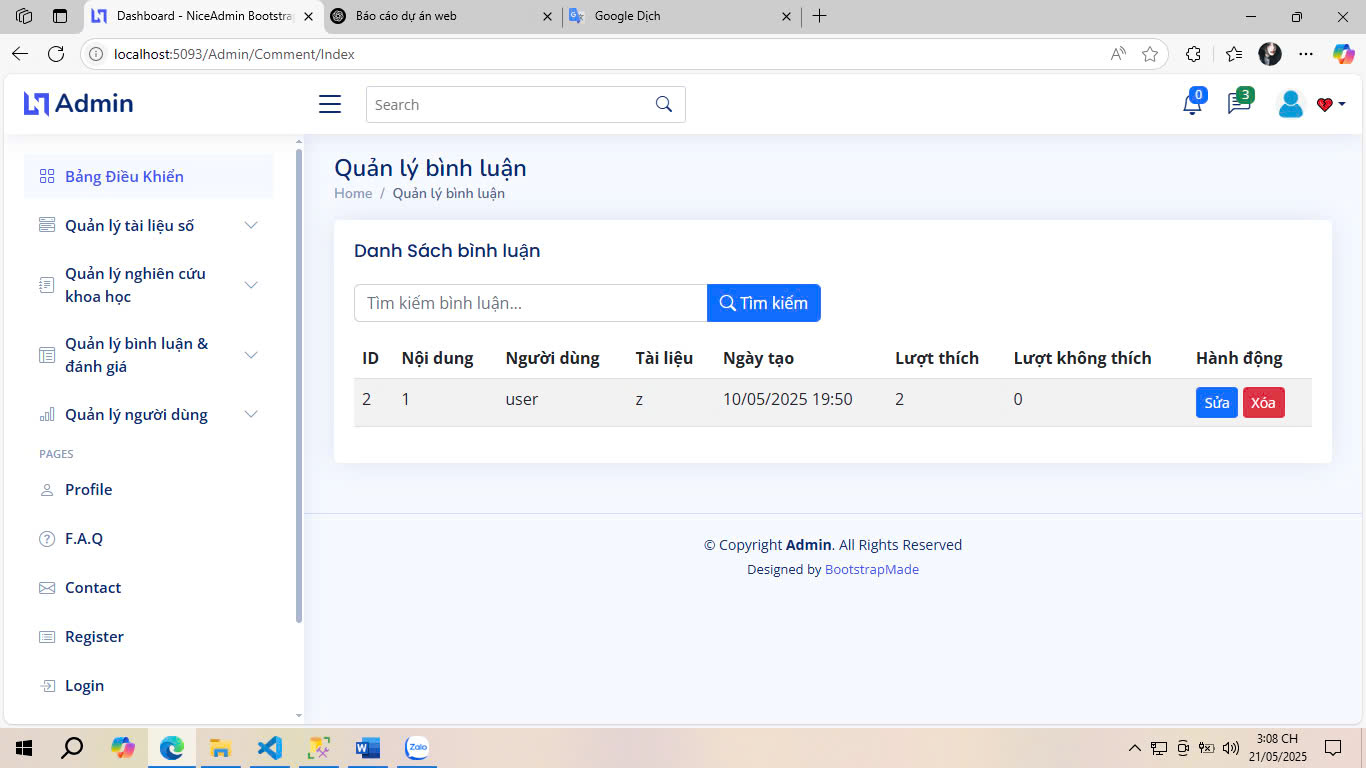


Fig 3.24. Admin Comment Management Interface

### 3.3.2.12. Admin Rating Management Interface

The Administrator's Review Management Interface is a tool that allows you to control and monitor all review activities (ratings) that users perform on books or research topics on the digital library system. This is an important part to ensure fairness, objectivity and transparency in the community feedback system.

On the interface, each review item displays information including: reviewer name, rating level (number of stars), reviewed object (book or research), review time, and processing status. The system supports filtering and searching by user, time or review value (from 1 to 5 stars), helping administrators easily analyze and monitor review activities according to each criterion.

Administrators can perform a number of management operations such as:

- Delete or disable reviews that are invalid, show signs of spam or violate the rules.

- Flag reviews reported by other users for manual checking.

- Check the editing history (if the user is allowed to update the rating).

- Statistics on the frequency and level of ratings, supporting the analysis of feedback trends from the community.

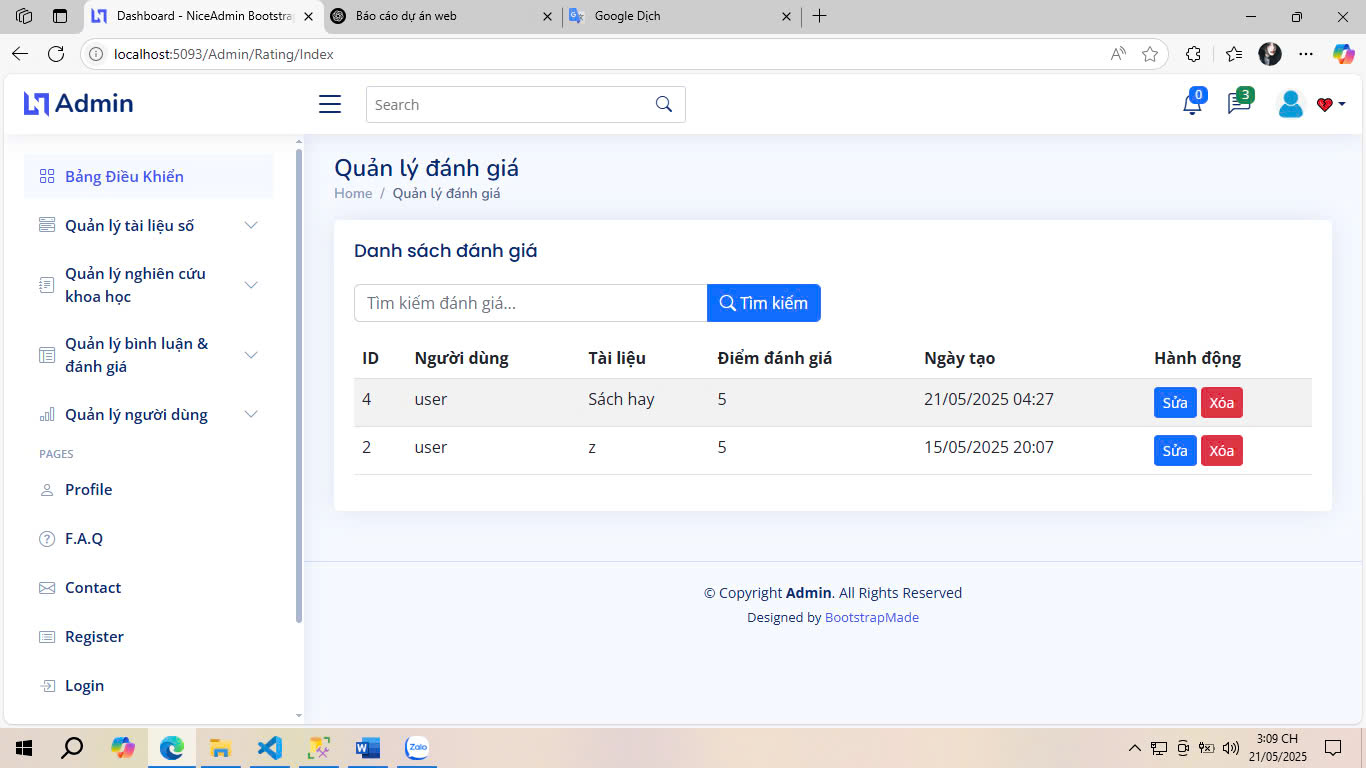


Fig 3.25. Admin Rating Management Interface

### 3.3.2.13. Admin Comment Statistics Interface

The Admin Comment Statistics interface is a data analysis tool that helps administrators gain an overview of user interaction activities in the form of comments in the digital library system. This is a component that supports decision making and content coordination, helping to evaluate user activity levels, response quality, and detect prominent trends in the community.

The interface displays visual charts and statistical tables showing important indicators such as:

- Total number of comments by day, week, month.

- Number of positive comments (with many likes) and negative comments (with many dislikes or reported).

- Rate of comments hidden, deleted, or requested for editing by the administrator.

- List of users who are most active in commenting.

- Documents or topics with the most comments, showing the level of interest from the community. The interface also allows filtering of statistical data by time, user, commenting subject (book or research).

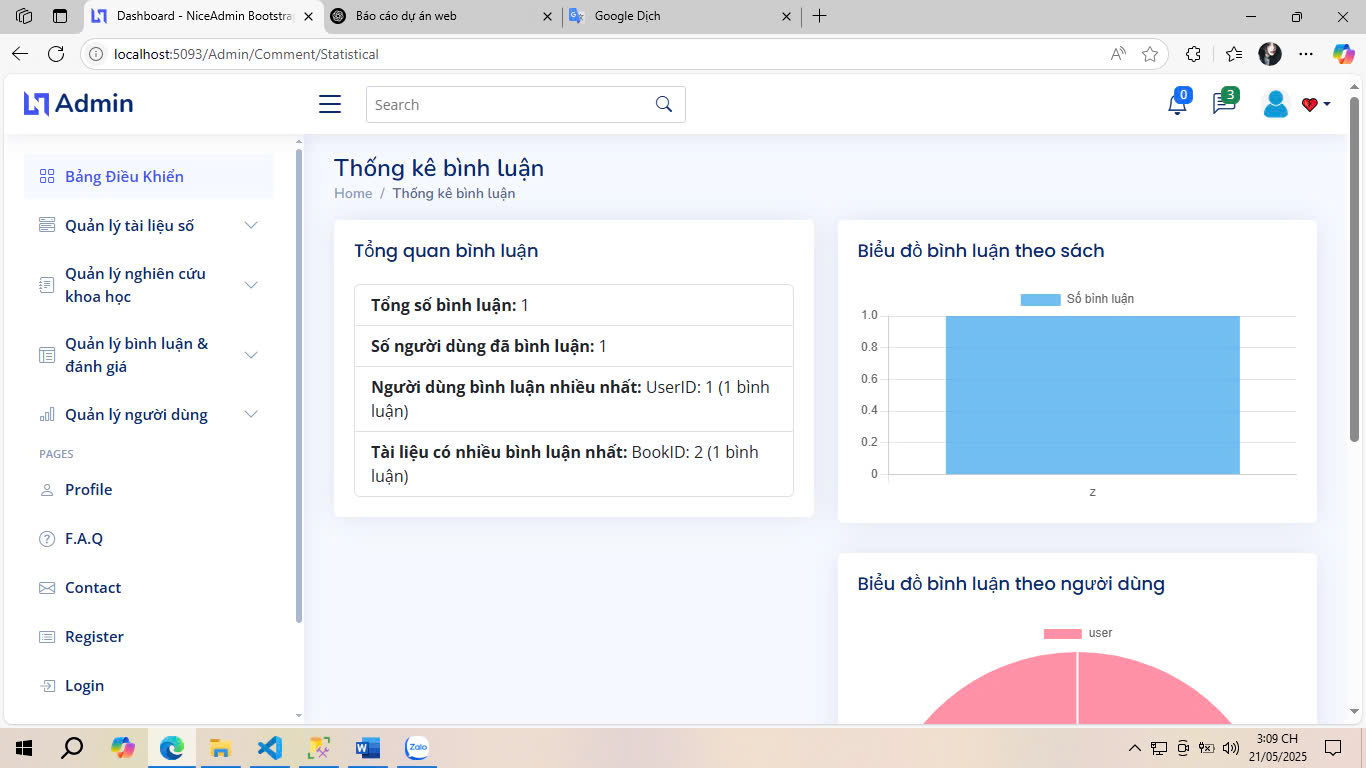


Fig 3.26. Admin Comment Statistics Interface

### 3.3.2.14. Admin User Management Interface

The Administrator's User Management Interface is the central hub for coordinating activities related to user accounts in the digital library system. This interface provides administrators with the ability to monitor, check, and promptly intervene in user accounts with the goal of ensuring safety, transparency, and efficiency in the system's operation.

On the interface, all accounts are displayed in a list format with important information such as: username, email, role (regular user, researcher, administrator), account status (active, locked, pending verification), registration date, and recent behaviors if any. Powerful filtering and searching features by name, email, role, or status allow administrators to access information quickly and accurately.

From this interface, administrators can perform management operations including:

- Activate, lock, or delete accounts when detecting violations of regulations or system maintenance requests.

- Change user roles (e.g. from regular user to researcher after approval).

- View user profile details including posting, rating, commenting activities.

- Send warnings or reminders to users via internal email integration.

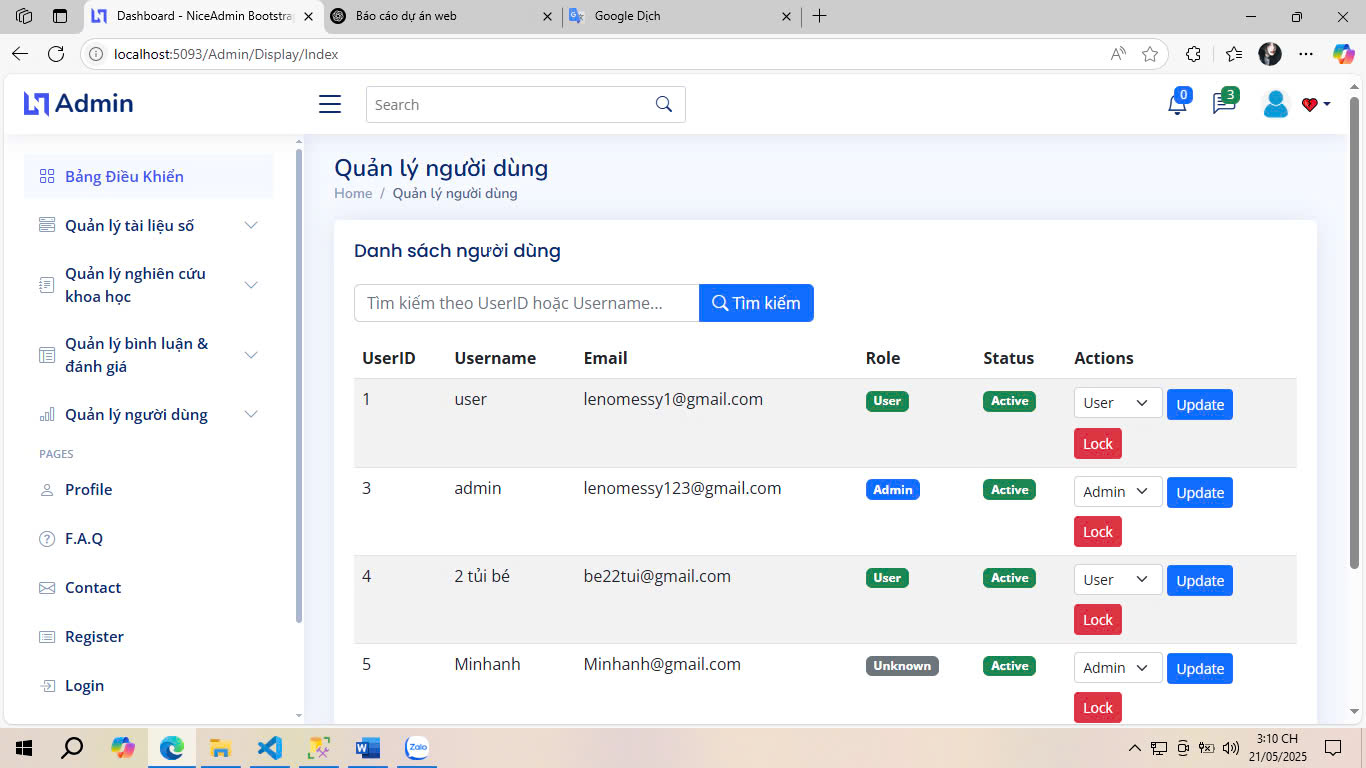


Fig 3.27. Admin User Management Interface

### 3.3.2.15. Admin Account Upgrade Approval Interface

The Administrator Account Upgrade Approval interface is a dedicated function that assists administrators in reviewing, verifying, and processing account upgrade requests from users. These requests typically come from users who wish to switch from a basic account to an advanced account (e.g., a researcher or lecturer account), in order to gain access or publish academic documents at a higher level.

On the interface, each upgrade request is presented in a list format with important information: the requester's full name, the time the request was submitted, the processing status (pending approval, approved, rejected), along with the attachments provided by the user such as a portrait photo, professional qualifications, an essay demonstrating ability, and self-presentation notes.

Administrators can select each request to open the detailed interface and perform actions such as:

- Preview all attached documents to verify identity and professional qualifications.

- Note internal comments for collective evaluation (if any).

- Approve the request, then the system will automatically update the account role and send a confirmation message to the user.

- Reject the request, with a clear reason so that the user knows and can resubmit the request if necessary.

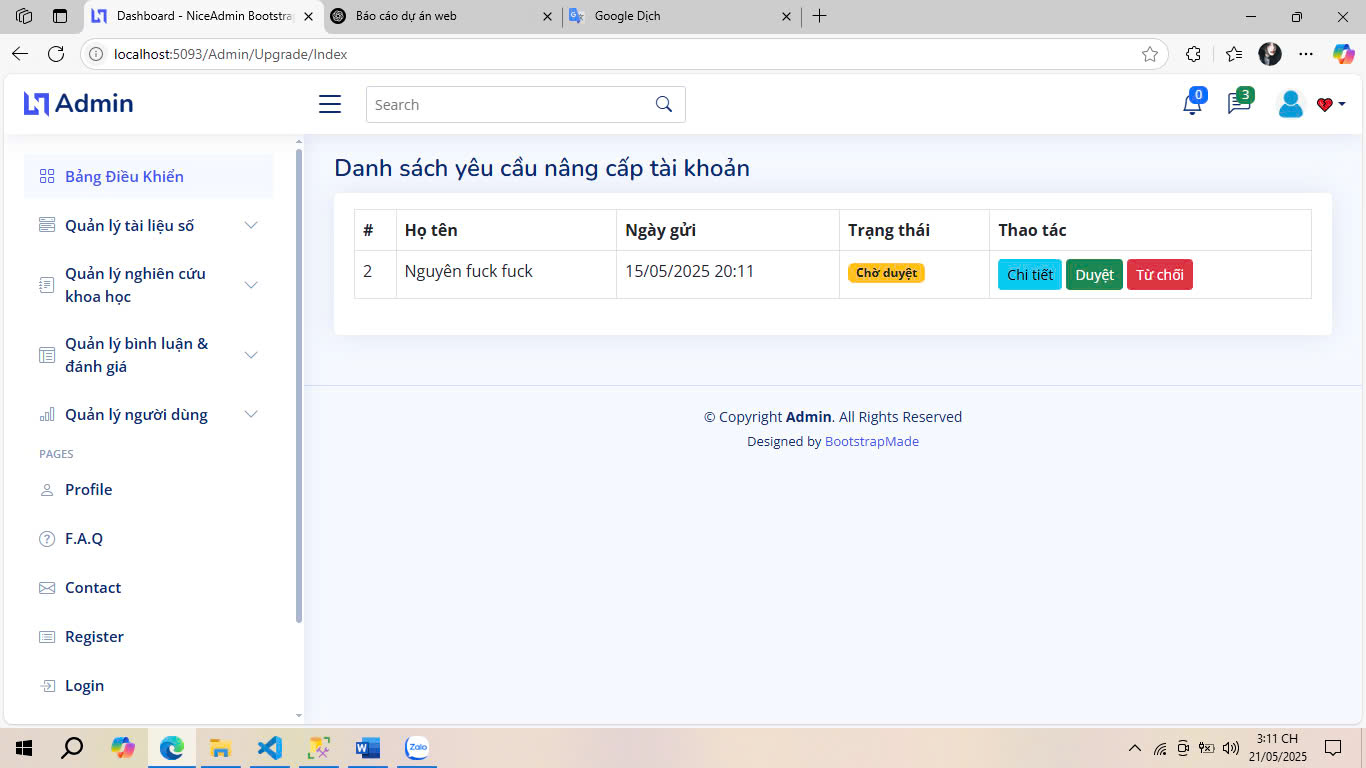


Fig 3.28. Admin Account Upgrade Approval Interface

### 3.3.2.16. Admin Account Upgrade Details Interface

The Administrator Account Upgrade Request Details interface is an extension of the upgrade approval interface, allowing administrators to view complete information and records related to each account upgrade request. This interface is designed to support a thorough, transparent, and well-founded review process. In this interface, administrators can access all data fields provided by users, including:

- Full name.

- Registered email.

- Type of account requesting upgrade.

- Date of request submission.

- Portrait.

- Degree/professional certificate (degree file).

- Essay describing the purpose of the upgrade (essay).

- Current status of the request (pending approval, approved, rejected).

- Administrator response notes (if any).

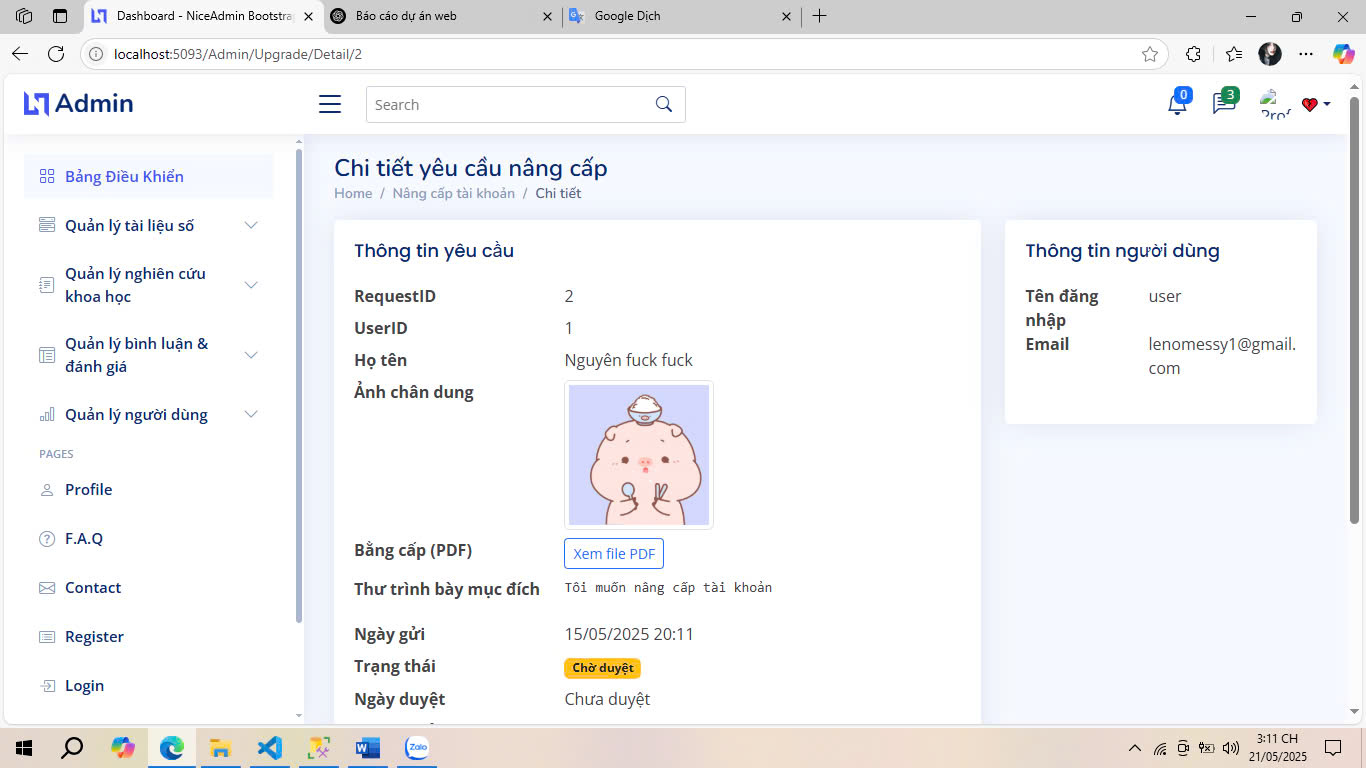


Fig 3.29. Admin Account Upgrade Details Interface

# CONCLUSTIONS

## 1. Achievement

Successfully designed and implemented a normalized relational schema covering all core entities: users (AdminUser), digital books (Book), research articles (Research), user‐generated labels (BookLabel/ResearchLabel), interactions (ratings, comments, bookmarks), and workflow requests (upgrade profiles).

## Limitations

The current design lacks a dedicated full-text search engine or faceted search capability; filtering is limited to exact-match queries on indexed fields.No built-in aggregation tables or dashboards for administrators to monitor trends (most-read books, active researchers, etc.) in real time.While relational normalization reduces redundancy, it may incur performance bottlenecks under very high traffic without caching or denormalized summary tables.The system does not yet enforce any content-protection or licensing controls, which may expose publishers’ content to unauthorized access.

## 3. Future direction

Integrate a full-text search engine (e.g., Elasticsearch or SOLR) and an AI-driven recommendation module to surface relevant books and research based on user behavior.Develop an analytics layer with materialized views or a data-warehouse to provide real-time reports on content usage, user engagement, and system health.

# REFERENCES

[1] Terry Felke-Morris, *Web Development and Design Foundations with HTML5,* 9th Edition, Pearson, 2018.

[2] Anemone Ratner, *Fundamentals Of ASP NET MVC Core (.NET 8): Basics To Advance*, Kindle Edition, 2023.

[3] Jessica Minnick, *Web Design with HTML5 & CSS3 - Comprehensive,* 8th Edition, Cengage Learning, 2017.

[4] Kameron Hussain, Frahaan Hussain, *Mastering Bootstrap 5: From Basics to Expert Projects*, Packt Publishing, 2023.