

**EASTERN INTERNATIONAL UNIVERSITY**  
**SCHOOL OF COMPUTING AND INFORMATION TECHNOLOGY**  
**DEPARTMENT OF SOFTWARE ENGINEERING**



**PROJECT REPORT**

**LIBRARY MANAGEMENT SYSTEM**

**Student(s)**

Nguyen Duc Trong – 2031200030

Luong Anh Kiet – 2031200050

Van Thi Huong Duyen – 2031209003

**Supervisor(s)**

Nguyen Xuan Cuong

**Binh Duong, November, 2024**

## **ABSTRACT**

The effective management of library resources has become increasingly essential in the digital era, as libraries seek to enhance accessibility and streamline operations. Traditional library management methods face challenges such as manual record-keeping inefficiencies and limited access for users. To address these issues, this project focuses on developing a comprehensive Library Management System using C# programming, ASP.NET Core MVC, ASP.NET Core Identity, and Entity Framework. The system incorporates authentication, authorization, and AJAX for a dynamic, user-friendly experience.

The motivation behind the project is to provide a modern solution that simplifies the management of library resources while ensuring secure user access and efficient resource allocation. This system offers administrators the ability to manage books, authors, and categories with CRUD (Create, Read, Update, Delete) operations, as well as track book lending and returns. For users, the platform delivers an intuitive interface to search for and filter books based on specific criteria, ensuring ease of access and navigation.

ASP.NET Core MVC was chosen for its powerful framework features, including modular development, seamless integration with Entity Framework, and support for role-based authentication and authorization through ASP.NET Core Identity. Additionally, AJAX, HTML, CSS, and JavaScript are used to enhance interactivity and responsiveness.

The current stage of the project includes the implementation of administrative functionalities, such as the ability to manage the library's catalog and user accounts. Role-based authorization ensures that administrators and users have distinct access privileges. Users can browse, search, and filter books, while administrators maintain an updated catalog and track lending activities. The system's lending and return features provide a streamlined process, ensuring efficient resource management.

This project aims to bridge the gap between traditional library systems and modern digital solutions, offering a robust, scalable platform tailored to meet the needs of libraries and their patrons.

## **ACKNOWLEDGEMENT**

The successful development of this Library Management System utilizing C# programming, ASP.NET Core MVC, ASP.NET Core Identity, and Entity Framework owes a debt of gratitude to several individuals and resources.

Firstly, I would like to express my deepest appreciation to my project supervisor, Mr. Nguyen Xuan Cuong. His invaluable guidance, insightful feedback, and unwavering encouragement throughout the development process were instrumental in ensuring the project's successful completion.

Secondly, I extend my gratitude to the faculty of Eastern International University (EIU) for their dedication to fostering a stimulating academic environment and providing access to valuable resources that significantly contributed to my academic and professional growth. Their support and resources, including access to development tools and frameworks, played a crucial role in this project.

Finally, I recognize the invaluable role of the ASP.NET Core community and its extensive ecosystem of resources. The framework's robust features and the readily available tutorials and documentation significantly streamlined the development process. Additionally, the wealth of online resources related to C#, Entity Framework, and AJAX provided crucial insights and practical guidance on various aspects of web application development.

I am deeply thankful for the collective efforts and support of all these individuals and resources, which made the successful completion of this project possible.

## TABLE OF CONTENTS

|  |            |
|--|------------|
| <b>ABSTRACT .....</b>                                  | <b>i</b>   |
| <b>ACKNOWLEDGEMENT .....</b>                           | <b>ii</b>  |
| <b>TABLE OF CONTENTS .....</b>                         | <b>iii</b> |
| <b>LIST OF FIGURES.....</b>                            | <b>v</b>   |
| <b>LIST OF TABLES.....</b>                             | <b>vi</b>  |
| <b>LIST OF ABBREVIATIONS .....</b>                     | <b>vii</b> |
| <b>CHAPTER 1. OVERVIEW .....</b>                       | <b>1</b>   |
| 1.1. Introduction .....                                | 1          |
| 1.2. Project objectives.....                           | 1          |
| 1.3. Challenge .....                                   | 2          |
| 1.4. Report structure .....                            | 2          |
| <b>CHAPTER 2. INTRODUCTION TO TECHNOLOGIES .....</b>   | <b>3</b>   |
| 2.1. Hyper Text Markup Language .....                  | 3          |
| 2.1.1. Introduction to Hyper Text Markup Language..... | 3          |
| 2.1.2. Advantages of HTML.....                         | 3          |
| 2.2. Cascading Style Sheets.....                       | 3          |
| 2.2.1. Introduction to Cascading Style Sheets .....    | 3          |
| 2.2.2. Advantages of CSS.....                          | 4          |
| 2.3. JavaScript .....                                  | 4          |
| 2.3.1. Introduction to JavaScript.....                 | 4          |
| 2.3.2. Advantages of JavaScript .....                  | 4          |
| 2.4. C# .....  | 5          |
| 2.4.1. Introduction to C#.....                         | 5          |
| 2.4.2. Advantages of C# .....                          | 5          |
| 2.5. ASP.NET Core MVC .....                            | 5          |
| 2.5.1. Introduction to ASP.NET Core MVC .....          | 5          |
| 2.5.2. Advantages of ASP.NET Core MVC.....             | 6          |
| 2.6. ASP.NET Core Identity .....                       | 6          |
| 2.6.1. Introduction to ASP.NET Core Identity .....     | 6          |
| 2.6.2. Advantages of ASP.NET Core Identity.....        | 6          |
| 2.7. Entity Framework.....                             | 6          |
| 2.7.1. Introduction to Entity Framework .....          | 6          |
| 2.7.2. Advantages of Entity Framework.....             | 6          |

|   |           |
|---|-----------|
| 2.8. SQL Server .....   | 7         |
| 2.8.1. Introduction to SQL Server .....                                 | 7         |
| 2.8.2. Advantages of SQL Server .....                                   | 7         |
| 2.9. AJAX (Asynchronous JavaScript and XML) .....                       | 7         |
| 2.9.1. Introduction to AJAX .....                                       | 7         |
| 2.9.2. Advantages of AJAX .....   | 7         |
| 2.10. System architecture and technology stack .....                    | 8         |
| 2.10.1. Front-end .....   | 8         |
| 2.10.2. Back-end .....  | 8         |
| <b>CHAPTER 3. APPLICATION ANALYSIS, DESIGN AND IMPLEMENTATION .....</b> | <b>9</b>  |
| 3.1. Database .....   | 9         |
| 3.1.1. Users table .....  | 10        |
| 3.1.2. Loans table .....  | 11        |
| 3.1.3. Books table .....  | 11        |
| 3.1.4. Authors table .....  | 12        |
| 3.1.5. Categories table .....   | 12        |
| 3.1.6. Carousel table .....   | 13        |
| <b>CHAPTER 4. EXPERIMENT AND DISCUSSION .....</b>                       | <b>14</b> |
| 4.1. Installation Environment .....                                     | 14        |
| 4.2. Results and Discussion .....                                       | 14        |
| 4.2.1. Admin Dashboard .....  | 14        |
| 4.2.2. User Management .....  | 15        |
| 4.2.3. Admin Books CRUD .....   | 16        |
| 4.2.4. Admin Book Category CRUD .....                                   | 18        |
| 4.2.5. Admin Author CRUD .....  | 19        |
| 4.2.6. Admin Loans CRUD .....   | 20        |
| 4.2.7. Home Page .....  | 21        |
| 4.2.8. Login/SignUp Page .....  | 23        |
| 4.2.9. Book List Page .....   | 24        |
| 4.2.10. Book Details Page .....   | 25        |
| <b>CHAPTER 5. CONCLUSION AND FUTURE WORKS .....</b>                     | <b>26</b> |
| 5.1. Conclusion .....   | 26        |
| 5.2. Future works .....   | 26        |

## LIST OF FIGURES

|  |    |
|--|----|
| Figure 1. ERD diagram of library management system .....           | 9  |
| Figure 2. Admin Dashboard .....                                    | 14 |
| Figure 3. User Management (Show, Detail, Edit, Delete).....        | 15 |
| Figure 4. Admin User CRUD (Add) .....                              | 15 |
| Figure 5. Admin User CRUD (Edit).....                              | 16 |
| Figure 6. Admin Books CRUD (Show, Add, Edit, Delete) .....         | 16 |
| Figure 7. Admin Books CRUD (Add).....                              | 17 |
| Figure 8. Admin Books CRUD (Edit) .....                            | 17 |
| Figure 9. Admin Book Category CRUD (Show, Add, Edit, Delete) ..... | 18 |
| Figure 10. Admin Book Category CRUD (Add).....                     | 18 |
| Figure 11. Admin Book Category CRUD (Edit).....                    | 18 |
| Figure 12. Admin Author CRUD (Show, Add, Edit, Delete) .....       | 19 |
| Figure 13. Admin Author CRUD (Add).....                            | 19 |
| Figure 14. Admin Author CRUD (Edit).....                           | 20 |
| Figure 15. Admin Loans CRUD (Show, Add, Edit, Delete) .....        | 20 |
| Figure 16. Admin Loans CRUD (Add) .....                            | 20 |
| Figure 17. Admin Loans CRUD (Edit) .....                           | 21 |
| Figure 18. Home Page .....   | 22 |
| Figure 19. Login/SignUp Page .....                                 | 23 |
| Figure 20. Book List Page .....                                    | 24 |
| Figure 21. Book Details Page .....                                 | 25 |

## LIST OF TABLES

|                                 |    |
|---------------------------------|----|
| Table 1. Users table .....      | 10 |
| Table 2. Loans table.....       | 11 |
| Table 3. Books table .....      | 11 |
| Table 4. Authors table .....    | 12 |
| Table 5. Categories table ..... | 12 |
| Table 6. Carousel table .....   | 13 |

## LIST OF ABBREVIATIONS

| No. | Term       | Meaning  |
|-----|------------|--|
| 1   | HTML       | Hyper Text Markup Language                       |
| 2   | CSS        | Cascading Style Sheets                           |
| 3   | JS         | JavaScript                                       |
| 4   | ASP.NET    | Active Server Pages Network Enabled Technologies |
| 5   | MVC        | Model-View-Controller                            |
| 6   | SQL Server | Microsoft Structured Query Language Server       |
| 7   | RDBMS      | Relational Database Management System            |
| 8   | IDE        | Integrated Development Environment               |
| 9   | CRUD       | Create, Read, Update, Delete                     |
| 10  | AJAX       | Asynchronous JavaScript and XML                  |
| 11  | OOP        | Object-Oriented Programming                      |



# **CHAPTER 1. OVERVIEW**

## **1.1. Introduction**

In the digital age, efficient management of resources has become increasingly important, particularly in libraries, which serve as repositories of knowledge and gateways to education. Traditional library management faces challenges such as manual operations, time-consuming processes, and accessibility limitations. To address these issues, this project aims to develop a modern Library Management System that leverages advanced web technologies to streamline operations and enhance user experience.

The project is driven by a passion for fostering education and making library resources more accessible. Libraries are not merely storage for books but are also hubs for learning and collaboration. By utilizing digital tools, we aim to bridge the gap between users and resources, ensuring convenience for students, researchers, and general users.

This system provides administrators with tools to manage books, authors, categories, and user accounts efficiently. For users, it offers a seamless experience for discovering and borrowing books. By leveraging C#, ASP.NET Core MVC, ASP.NET Core Identity, and Entity Framework, this project integrates robust security, scalability, and interactivity, making library resources easily accessible and ensuring an efficient management process.

## **1.2. Project objectives**

Our project is designed to address the unique needs of administrators and users, creating a comprehensive Library Management System that prioritizes efficiency, security, and usability. The system is developed using modern technologies, including ASP.NET Core MVC, which provides a robust Model-View-Controller architecture, AJAX for dynamic content updates, and Entity Framework for efficient and scalable database interactions. This ensures a seamless and reliable experience for all stakeholders.

For administrators, the system offers a role-based authentication and authorization mechanism to secure access and ensure that only authorized users can perform administrative tasks. It supports full CRUD (Create, Read, Update, Delete) operations for managing books, authors, and categories, enabling efficient resource management.

For users, the user interface is designed to be secure, intuitive, and user-friendly, supporting seamless book discovery.

By integrating these features, the Library Management System not only meets the diverse needs of its users but also establishes a foundation for scalability and enhanced user engagement.

### 1.3. Challenge

Despite the successful implementation of CRUD functions for the admin section and basic functions for users of our project, there are still a number of limits and difficulties that must be acknowledged:

- **Local Environment Dependency:** The application currently operates in a local environment, limiting its accessibility for external users.
- **Security Enhancements:** While ASP.NET Core Identity provides basic security, there is scope for implementing advanced measures to protect sensitive data.
- **Time Management:** Some delays occurred due to unforeseen complexities in integrating multiple technologies and features.

These drawbacks highlight areas that would require more focus and development in order to ensure the continued effectiveness and scalability of the website. Our team are still dedicated to addressing these challenges and providing administrators and users with a reliable, safe, and easy-to-use solution through teamwork and strategic planning.

### 1.4. Report structure

To provide a comprehensive understanding of our project's development and implementation, we have structured it into five chapters, each addressing distinct facets of the endeavor:

- **Chapter 1:** Introduction, which delves into the motivation behind the project, presents the project objectives, and outlines the project's structure.
- **Chapter 2:** Introduction of the technologies employed in the project, providing insights into the tools and frameworks utilized to realize our objectives.
- **Chapter 3:** Detailed application analysis, design, and implementation, offering a comprehensive overview of the development process from conception to execution.
- **Chapter 4:** Description of the installation environment of the project and showcasing the project results and discussions.
- **Chapter 5:** Conclusion of the project, summarizing key findings and outcomes, and outlining future avenues for exploration and development.

This structured approach ensures a comprehensive understanding of the project's development lifecycle and provides a solid foundation for future enhancements and scalability.

## CHAPTER 2. INTRODUCTION TO TECHNOLOGIES

### 2.1. Hyper Text Markup Language

#### 2.1.1. Introduction to Hyper Text Markup Language

Hyper Text Markup Language (HTML) is used to design web pages using a standardized language. It makes use of plain text tags and elements, which the browser decodes to render the website.

The headings, paragraphs, lists, images, and other content on a web page are defined by HTML tags. Additionally, HTML specifies how these items must appear on the page. Many HTML components can also have attributes add to them to further alter how they behave or look on a web page.

#### 2.1.2. Advantages of HTML

The use of HTML in web browsers has numerous benefits. The following are some noteworthy advantages of HTML:

- **Ease of use:** HTML's simplicity to learn and use is one of its primary benefits. The fundamentals of HTML are simple to learn and can be done quickly, even by people with little coding expertise.
- **Browser support:** Websites built with HTML code are guaranteed to be cross-browser compatible, meaning that users of any browser will be able to access your content correctly. This is because the majority of modern browsers support HTML code.
- **Compatibility with search engines:** HTML's compliance with search engines, such as Google and Bing, is another benefit. This helps guarantee that your page is appropriately indexed and shows up in relevant searches.
- **Multimedia elements:** HTML enables developers to effortlessly incorporate multimedia components, such as music files, movies, and photos, into their web pages without the need for further software installations or plugins. End consumers will find it more convenient to browse website content as a result.

### 2.2. Cascading Style Sheets

#### 2.2.1. Introduction to Cascading Style Sheets

Cascading Style Sheets (CSS) is a simply designed language intended to make the process of creating visually appealing web pages easier. Developers and designers can apply styles to HTML documents via CSS. It describes how a webpage should look. It specifies fonts, colors, spacing, how columns are sized and laid out, what background images or colors are used, layout designs, variations in display for different devices and screen sizes as well as a variety of other effects.

### 2.2.2. Advantages of CSS

CSS is simple to learn and comprehend, it offers an HTML document's display significant control. Here are some of advantages of CSS:

- **CSS saves time:** We can create a CSS sheet once and use it for various HTML pages. Every HTML element has a style that we may specify and apply to as many Web pages as we like.
- **Pages load faster:** We do not have to write HTML tag attributes every time if we use CSS. Simply create a single CSS rule for a tag and apply it to every instance of that tag. So, quicker download times correspond to less code.
- **Easy maintenance:** To make a global change, simply change the style, and all the elements across all the web pages will update instantly.
- **Better styles than HTML:** CSS offers a significantly greater range of properties than HTML, allowing developers and designers to give HTML page a far nicer appearance than HTML attributes can offer.
- **Multiple Device Compatibility:** Style sheets facilitate the optimization of content for multiple device types. Different versions of a website can be given for printing or for handheld devices such as cell phones or for printing utilizing the same HTML document.
- **Global web standards:** CSS is now advised over HTML characteristics, which are now being deprecated. Therefore, it's a good idea to begin utilizing CSS in all HTML pages in order to ensure browser compatibility in the future.

## 2.3. JavaScript

### 2.3.1. Introduction to JavaScript

JavaScript (JS) is a compiled programming language that is lightweight, cross-platform, single-threaded, and interpretable. It is sometimes referred to as the language used for webpage scripts. Although numerous non-browser contexts also use it, it is most renowned for being used in the building of online pages.

JavaScript is a dynamically typed language with weak typing. Both Client-side and Server-side development can make use of JavaScript. JavaScript is both an imperative and declarative type of language.

### 2.3.2. Advantages of JavaScript

Some noteworthy advantages of JavaScript include the following:

- **Save time and bandwidth:** JavaScript is always executed in the client environment to minimize bandwidth usage and speed up the execution process, regardless of where it is hosted.

- **Suitable for every browser:** The primary benefit of JavaScript is its ability to function with all current browsers and generate a comparable output.
- **Community support:** Major programs are developed by international corporations to promote community development.
- **Environment support:** JavaScript is compatible with a wide range of programming languages and can be used in a vast array of applications.
- **Open source:** A lot of open-source initiatives offer helpful assistance to developers in adding JavaScript.
- **Simple to use:** Learning JavaScript does not require much experience and the ability to create complex interfaces is provided.
- **Programming language:** You may easily and fast increase your understanding of JavaScript by enrolling in one of the many courses available in this sector.
- **Backend usage:** JavaScript can be used on servers running Node.js in a few different ways. With just JavaScript, an entire JavaScript application may be created from start to finish.

## 2.4. C#

### 2.4.1. Introduction to C#

C# is a modern, object-oriented programming language developed by Microsoft. Designed for building a variety of applications, including web, desktop, and mobile applications, C# is widely used for its simplicity, versatility, and powerful features. It integrates seamlessly with the .NET ecosystem, making it ideal for enterprise-level applications.

### 2.4.2. Advantages of C#

- **Object-Oriented Programming (OOP):** Encourages code reusability and modularity.
- **Rich libraries:** Includes a comprehensive set of libraries that simplify development.
- **Type safety:** Reduces runtime errors by enforcing strict type-checking.
- **Cross-platform compatibility:** Supported by .NET Core, enabling application deployment on various platforms.
- **Integration with Microsoft tools:** Works seamlessly with tools like Visual Studio and Azure.
- **Community support:** Extensive documentation and community resources for learning and troubleshooting.

## 2.5. ASP.NET Core MVC

### 2.5.1. Introduction to ASP.NET Core MVC

ASP.NET Core MVC is a framework for building web applications using the Model-View-Controller (MVC) design pattern. It allows developers to separate application concerns,

resulting in clean, maintainable, and testable code. ASP.NET Core MVC is a part of the ASP.NET Core platform, optimized for performance and cross-platform deployment.

### 2.5.2. Advantages of ASP.NET Core MVC

- **Cross-platform:** Allows deployment on Windows, macOS, and Linux.
- **Performance:** Highly optimized for speed and scalability.
- **Modular architecture:** Developers can add only the required packages, reducing application size.
- **Testability:** Promotes separation of concerns for easier unit testing.
- **Security:** Includes built-in protection against common vulnerabilities like CSRF and XSS.
- **Integration:** Seamlessly integrates with modern front-end frameworks and libraries.

## 2.6. ASP.NET Core Identity

### 2.6.1. Introduction to ASP.NET Core Identity

ASP.NET Core Identity is a membership system that provides authentication and authorization for web applications. It offers features such as user registration, password hashing, role-based access control, and external login providers (Google, Facebook, etc.).

### 2.6.2. Advantages of ASP.NET Core Identity

- **Built-in security:** Implements best practices for password management and encryption.
- **Customizability:** Easily extendable to meet specific application requirements.
- **Role-based access control:** Simplifies the implementation of user roles and permissions.
- **Third-party login integration:** Supports external authentication providers like Google, Microsoft, and Facebook.
- **Session management:** Offers features like two-factor authentication and account logout.

## 2.7. Entity Framework

### 2.7.1. Introduction to Entity Framework

Entity Framework (EF) is an Object-Relational Mapping (ORM) framework for .NET developers. It enables developers to interact with databases using .NET objects, eliminating the need for complex SQL queries.

### 2.7.2. Advantages of Entity Framework

- **Simplifies database access:** Allows developers to work with data using .NET objects without writing SQL.

- **Productivity:** Reduces boilerplate code, speeding up development.
- **Maintainability:** Easy to update and maintain with a code-first or database-first approach.
- **Cross-platform:** Supported by .NET Core for cross-platform applications.
- **Strongly typed data:** Provides compile-time checking, reducing runtime errors.
- **Lazy loading:** Efficiently fetches only the required data.

## 2.8. SQL Server

### 2.8.1. Introduction to SQL Server

SQL Server is a relational database management system developed by Microsoft. It is widely used for storing, retrieving, and managing data in enterprise applications. It supports structured query language (SQL) and integrates seamlessly with the .NET ecosystem.

### 2.8.2. Advantages of SQL Server

- **Scalability:** Handles large volumes of data with ease.
- **Security:** Provides robust security features, including data encryption and access control.
- **Performance:** Includes features like indexing, query optimization, and in-memory processing for high-speed operations.
- **Integration:** Works seamlessly with other Microsoft products, such as .NET and Azure.
- **Backup and recovery:** Provides comprehensive backup and disaster recovery options.
- **Tools and support:** Includes built-in tools for database management and debugging.

## 2.9. AJAX (Asynchronous JavaScript and XML)

### 2.9.1. Introduction to AJAX

AJAX is a web development technique that allows web applications to update data asynchronously without reloading the entire page. It uses a combination of JavaScript and XML or JSON for data exchange between the client and server.

### 2.9.2. Advantages of AJAX

- **Improved user experience:** Enables partial updates, reducing page reloads and enhancing responsiveness.
- **Reduced server load:** Exchanges only the necessary data, minimizing bandwidth usage.
- **Cross-browser support:** Works with most modern web browsers.
- **Seamless integration:** Easily integrates with front-end frameworks and libraries.
- **Faster development:** Simplifies dynamic content loading, speeding up development.

- **Flexibility:** Supports various data formats, including JSON and XML.

## **2.10. System architecture and technology stack**

### **2.10.1. Front-end**

- **HTML:** Used for structuring the content of web pages.
- **CSS:** Utilized for styling the web pages, ensuring a consistent look and feel across the application.
- **JavaScript:** Employed for adding interactivity and dynamic content to a web pages.

### **2.10.2. Back-end**

- **C#:** A modern, object-oriented programming language developed by Microsoft.
- **ASP.NET Core MVC:** A framework for building scalable, maintainable, and testable web applications using the Model-View-Controller (MVC) pattern.
- **ASP.NET Core Identity:** Provides authentication and authorization capabilities, enhancing security for users.
- **Entity Framework:** Simplifies database interactions with object-relational mapping (ORM) techniques.
- **SQL Server:** Manages the application's data storage and retrieval operations.
- **AJAX:** Ensures seamless data exchanges between client and server, enhancing interactivity.

These technologies were carefully chosen to ensure the development of a robust, secure, and user-friendly Library Management System.

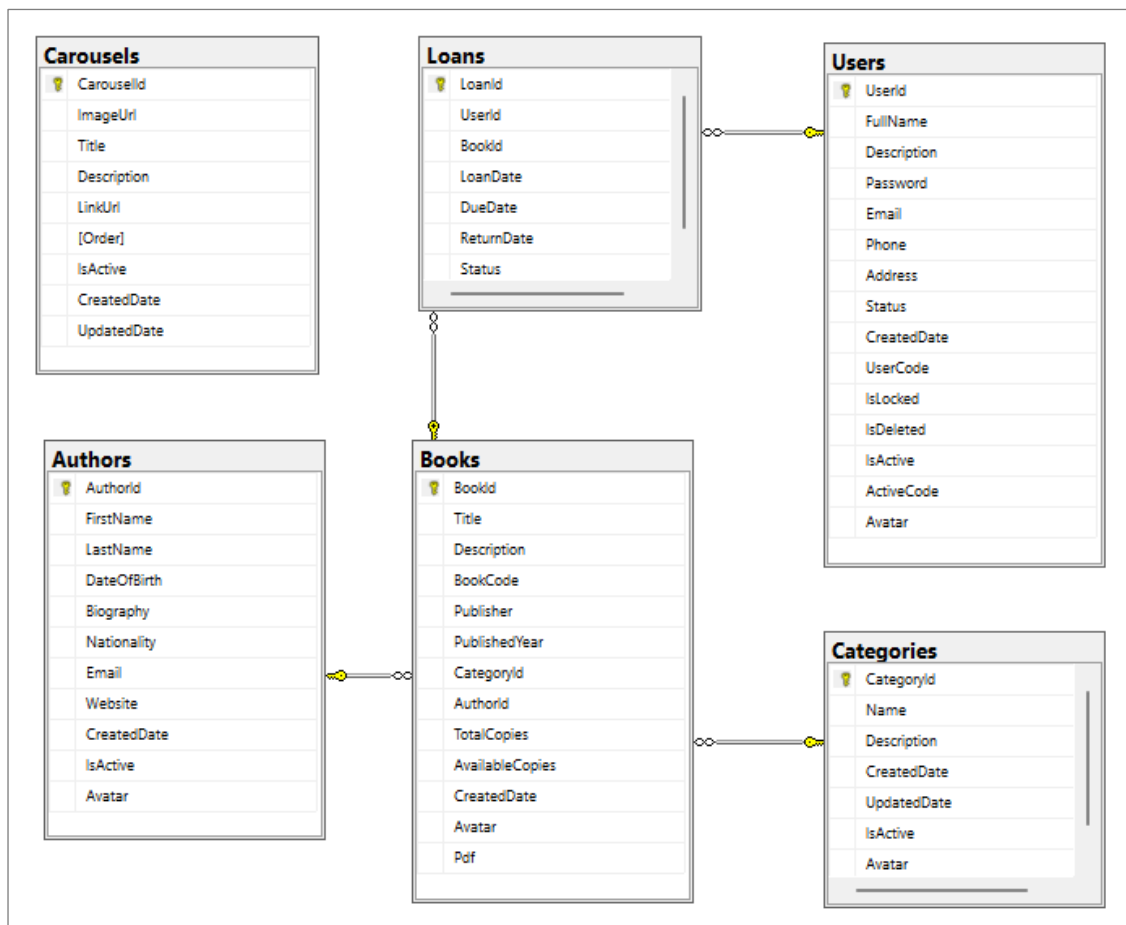


## CHAPTER 3. APPLICATION ANALYSIS, DESIGN AND IMPLEMENTATION

### 3.1. Database

Microsoft SQL Server is a powerful and widely used relational database management system (RDBMS) ideal for managing library data and supporting web applications. In this stage of the project, we focused on organizing the database for administration, specifically for managing books, authors, book categories, carousels and users.

The following is a simplified Entity-Relationship Diagram (ERD) representing the structure of the database:



*Figure 1. ERD diagram of library management system*

Specifically, the project include these table:

### 3.1.1. Users table

*Table 1. Users table*

| No. | Field Name  | Type          | Description  |
|-----|-------------|---------------|--|
| 1   | UserId      | int           | Unique identifier for each user, auto-incremente                 |
| 2   | Fullname    | nvarchar(200) | Full name of the user  |
| 3   | Description | nvarchar(MAX) | Additional info about the user, like interests or bio            |
| 4   | Password    | nvarchar(MAX) | Hashed password for secure authentication                        |
| 5   | Email       | nvarchar(100) | User's email address, used for communication and verification    |
| 6   | Phone       | nvarchar(20)  | User's contact phone number                                      |
| 7   | Address     | nvarchar(MAX) | User's physical address for mailing or location purposes         |
| 8   | Status      | int           | Represents the user's status                                     |
| 9   | CreatedDate | datetime      | Date and time when the user account was created                  |
| 10  | UserCode    | nvarchar(MAX) | Unique code for internal identification of the user              |
| 11  | IsLocked    | bit           | Indicates if the account is locked (1) or active (0)             |
| 12  | IsDeleted   | bit           | Marks the account as deleted (1) or active (0) for soft deletion |
| 13  | IsActive    | bit           | Shows if the account is active (1) or inactive (0)               |
| 14  | ActiveCode  | nvarchar(MAX) | Code for account activation, sent via email during registration  |
| 15  | Avatar      | nvarchar(MAX) | Avatar's User - Local location in the server to get the picture  |

### 3.1.2. Loans table

Table 2. Loans table

| No. | Field Name | Type     | Description  |
|-----|------------|----------|--|
| 1   | LoanId     | int      | (Primary Key, Auto-Increment)  |
| 2   | UserId     | int      | References the user who borrowed the book  |
| 3   | BookId     | int      | References the borrowed book   |
| 4   | LoanDate   | datetime | Date when the book was borrowed  |
| 5   | DueDate    | datetime | Date when the book is due  |
| 6   | ReturnDate | datetime | Date when the book was returned  |
| 7   | Status     | int      | Status of the loan (e.g., "Active", "Returned", "Overdue"). 0: Active , 1: Returned , 2: Overdue |

### 3.1.3. Books table

Table 3. Books table

| No. | Field Name      | Type          | Description  |
|-----|-----------------|---------------|--|
| 1   | BookId          | int           | (Primary Key, Auto-Increment)  |
| 2   | Title           | nvarchar(200) | Title of the book  |
| 3   | Description     | nvarchar(MAX) | Description of the book  |
| 4   | BookCode        | nvarchar(MAX) | Standard Book Number   |
| 5   | Publisher       | nvarchar(MAX) | Publisher description  |
| 6   | PublishedYear   | varchar(255)  | Year the book was published  |
| 7   | CategoryId      | int           | References the category  |
| 8   | AuthorId        | int           | References the author  |
| 9   | TotalCopies     | int           | Total number of physical copies of the book in the library                                     |
| 10  | AvailableCopies | int           | Total number of physical copies of the book currently in the library, excluding copies on loan |
| 11  | CreatedDate     | datetime      | Date when the book record was created  |
| 12  | Avatar          | nvarchar(MAX) | Cover image of the book - Local location in the server to get the picture                      |
| 13  | Pdf             | nvarchar(MAX) | Store the version pdf for reading online   |

### 3.1.4. Authors table

Table 4. Authors table

| No. | Field Name  | Type          | Description   |
|-----|-------------|---------------|---|
| 1   | AuthorId    | int           | (Primary Key, Auto-Increment).<br>Unique identifier for each author                                       |
| 2   | FirstName   | nvarchar(100) | Author's first name   |
| 3   | LastName    | nvarchar(100) | Author's last name  |
| 4   | DateOfBirth | datetime      | Author's date of birth  |
| 5   | Biography   | nvarchar(MAX) | A short biography of the author   |
| 6   | Nationality | nvarchar(100) | Nationality of the author   |
| 7   | Email       | nvarchar(100) | Author's Email  |
| 8   | Website     | nvarchar(100) | Author's Website  |
| 9   | CreatedDate | datetime      | Date when the author record was created   |
| 10  | IsActive    | bit           | The IsActive column helps indicate whether a record is currently active and usable within the application |
| 11  | Avatar      | nvarchar(MAX) | Authors' Avatar - Local location in the server to get the picture   |

### 3.1.5. Categories table

Table 5. Categories table

| No. | Field Name  | Type          | Description  |
|-----|-------------|---------------|--|
| 1   | CategoryId  | int           | Unique identifier for each category<br>(Primary Key)                 |
| 2   | Name        | nvarchar(MAX) | Name of the category; must be unique for identification              |
| 3   | Description | nvarchar(MAX) | Additional information about the category                            |
| 4   | CreatedDate | datetime      | Date and time when the category was created                          |
| 5   | UpdatedDate | datetime      | Date and time when the category was last updated                     |
| 6   | IsActive    | bit           | Indicates if the category is active (1) or inactive (0)              |
| 7   | Avatar      | nvarchar(MAX) | Categories's image - Local location in the server to get the picture |

### 3.1.6. Carousel table

*Table 6. Carousel table*

| No. | Field Name  | Type          | Description  |
|-----|-------------|---------------|--|
| 1   | CarouselId  | int           | Unique identifier for each carousel (Primary Key)                        |
| 2   | ImageUrl    | nvarchar(MAX) | URL of the image to be displayed in the carousel                         |
| 3   | Title       | nvarchar(200) | The title or heading associated with the carousel item                   |
| 4   | Description | nvarchar(MAX) | Additional text or details about the carousel item                       |
| 5   | LinkUrl     | nvarchar(MAX) | URL that the carousel item links to when clicked                         |
| 6   | Order       | int           | Defines the display order of carousel items                              |
| 7   | IsActive    | bit           | Indicates whether the carousel item is currently active (visible) or not |
| 8   | CreatedDate | datetime      | The date and time when the carousel item was created                     |
| 9   | UpdatedDate | datetime      | The date and time when the carousel item was last updated                |

## CHAPTER 4. EXPERIMENT AND DISCUSSION

### 4.1. Installation Environment

The installation environment plays a crucial role in ensuring the performance and functionality of the Library Management System. To set up the necessary configuration, we utilized the following environment:

- Operating System: Window 11
- Programming Language: C#
- Integrated Development Environment (IDE): Visual Studio 2022
- Framework:
  - + ASP.NET Core: Version 8.0.10
  - + Entity Framework Core: Version 8.0.10
- Project Management Tool: NuGet Package Manager
- Relational Database Management System (RDBMS): Microsoft SQL Server
- Database Management Tool: SQL Server Management Studio (SSMS)
- Frontend Technologies: HTML5, CSS3, Bootstrap5, JavaScript, AJAX
- Browser for Testing: Google Chrome, Microsoft Edge

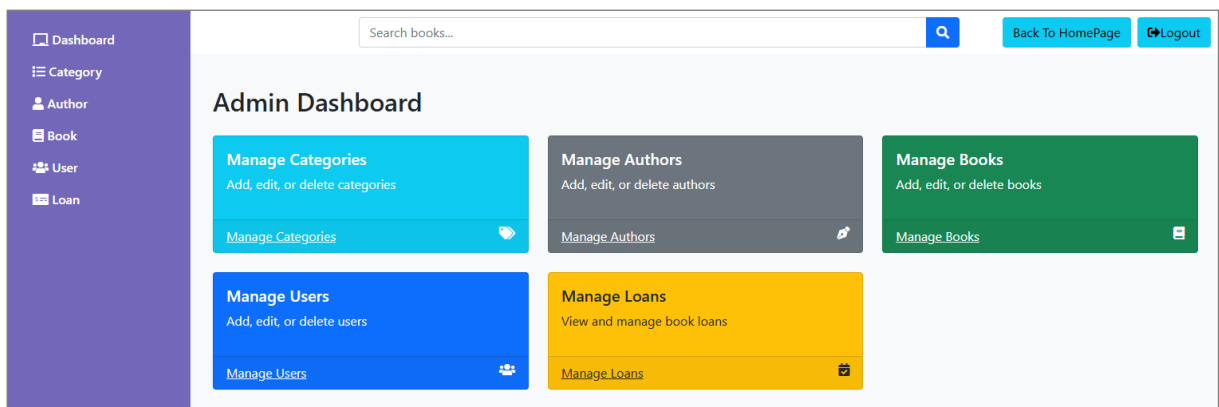
This configuration provided a stable and robust environment for developing, testing, and deploying the system, ensuring compatibility and performance during the project's lifecycle.

### 4.2. Results and Discussion

In this part, we will show all the results of our project:

#### 4.2.1. Admin Dashboard

The Admin Dashboard is the main hub for administrators, providing a comprehensive overview of key metrics and functions. It offers quick access to various management features, enabling efficient monitoring and control of system operations.



*Figure 2. Admin Dashboard*

### 4.2.2. User Management

The User Management interface allows administrators to efficiently manage user accounts. It includes functions to display user lists, view detailed information, edit user profiles, and delete accounts, ensuring optimized user management.

Dashboard

Category

Author

Book

User

Loan

Search books...

Back To HomePage

Logout

User

Create New

| FullName      | Description      | Password    | Email  | Phone      | Address      | Status | CreatedDate           | UserCode | IsLocked                 | IsDeleted                | IsActive                            | ActiveCode | Avatar             |  |
|---------------|------------------|-------------|--|------------|--------------|--------|-----------------------|----------|--------------------------|--------------------------|-------------------------------------|------------|--------------------|--|
| Alice Johnson | Regular user     | password123 | <a href="mailto:alice@example.com">alice@example.com</a> | 1234567890 | 123 Maple St | 1      | 11/21/2024 4:59:14 PM | U001     | <input type="checkbox"/> | <input type="checkbox"/> | <input checked="" type="checkbox"/> | ACT123     | /avatars/alice.jpg | <a href="#">Edit</a><br><a href="#">Detail</a><br><a href="#">Delete</a> |
| Bob Smith     | VIP user         | password456 | <a href="mailto:bob@example.com">bob@example.com</a>     | 2345678901 | 456 Oak St   | 1      | 11/21/2024 4:59:14 PM | U002     | <input type="checkbox"/> | <input type="checkbox"/> | <input checked="" type="checkbox"/> | ACT456     | /avatars/bob.jpg   | <a href="#">Edit</a><br><a href="#">Detail</a><br><a href="#">Delete</a> |
| Cathy Lee     | Library staff    | password789 | <a href="mailto:cathy@example.com">cathy@example.com</a> | 3456789012 | 789 Pine St  | 1      | 11/21/2024 4:59:14 PM | U003     | <input type="checkbox"/> | <input type="checkbox"/> | <input checked="" type="checkbox"/> | ACT789     | /avatars/cathy.jpg | <a href="#">Edit</a><br><a href="#">Detail</a><br><a href="#">Delete</a> |
| David Brown   | Frequent visitor | password101 | <a href="mailto:david@example.com">david@example.com</a> | 4567890123 | 101 Elm St   | 1      | 11/21/2024 4:59:14 PM | U004     | <input type="checkbox"/> | <input type="checkbox"/> | <input checked="" type="checkbox"/> | ACT101     | /avatars/david.jpg | <a href="#">Edit</a><br><a href="#">Detail</a><br><a href="#">Delete</a> |
| Eva Green     | Book enthusiast  | password102 | <a href="mailto:eva@example.com">eva@example.com</a>     | 5678901234 | 202 Birch St | 1      | 11/21/2024 4:59:14 PM | U005     | <input type="checkbox"/> | <input type="checkbox"/> | <input checked="" type="checkbox"/> | ACT102     | /avatars/eva.jpg   | <a href="#">Edit</a><br><a href="#">Detail</a><br><a href="#">Delete</a> |

Figure 3. User Management (Show, Detail, Edit, Delete)

Dashboard

Category

Author

Book

User

Loan

Search books...

Back To HomePage

Logout

Create New User

To exit full screen, press and hold **Esc**

FullName

Description

Password

Email

Phone

Address

Status

CreatedDate

UserCode

☐ IsLocked

☐ IsDeleted

☐ IsActive

ActiveCode

Avatar

Create

Back to List

Figure 4. Admin User CRUD (Add)

Dashboard
Category
Author
Book
User
Loan

Search books...
Back To HomePage
Logout

### Edit User

FullName

Cathy Lee

Description

Library staff

Password

password789

Email

cathy@example.com

Phone

3456789012

Address

789 Pine St

Status

1

CreateDate

11/21/2024 04:59:14.426 PM

UserCode

U003

IsLocked

☐

IsDeleted

☐

IsActive

☒

ActiveCode

ACT789

Avatar

/avatars/cathy.jpg

Save

Back to List

Figure 5. Admin User CRUD (Edit)

### 4.2.3. Admin Books CRUD

The Admin Books CRUD interface (Create, Read, Update, Delete) allows administrators to manage the book inventory. It includes functions to display the book catalog, add new titles, edit existing entries, and delete books from the inventory.

Dashboard
Category
Author
Book
User
Loan

Search books...
Back To HomePage
Logout

### Books

[Create New](#)

| Title                 | Description                   | BookCode | Publisher | PublishedYear          | Category        | Author  | TotalCopies | AvailableCopies | CreatedDate           | Avatar                                   | Pdf          |  |
|-----------------------|-------------------------------|----------|-----------|------------------------|-----------------|---------|-------------|-----------------|-----------------------|--|--------------|--|
| C# Programming        | Comprehensive guide to C#     | B001     | TechBooks | 1/1/2022 12:00:00 AM   | Programming     | John    | 5           | 3               | 11/21/2024 5:14:45 PM | images/books/am-nhac.png                 | pdf/Lab3.pdf | <a href="#">Edit</a><br><a href="#">Delete</a> |
| ASP.NET MVC           | Learn ASP.NET MVC framework   | B002     | WebBooks  | 5/10/2023 12:00:00 AM  | Fiction         | Jane    | 4           | 4               | 11/21/2024 5:14:45 PM | images/books/cong-nghe-6.png             | pdf/Lab4.pdf | <a href="#">Edit</a><br><a href="#">Delete</a> |
| Entity Framework Core | Master EF Core                | B003     | DataBooks | 3/15/2021 12:00:00 AM  | Science Fiction | Tom     | 3           | 2               | 11/21/2024 5:14:45 PM | images/books/gdcd-6.png                  | pdf/Lab5.pdf | <a href="#">Edit</a><br><a href="#">Delete</a> |
| LINQ in C#            | Querying data with LINQ       | B004     | DataBooks | 7/20/2020 12:00:00 AM  | Programming     | Emily   | 2           | 2               | 11/21/2024 5:14:45 PM | images/books/lich-su-6.png               | pdf/Lab6.pdf | <a href="#">Edit</a><br><a href="#">Delete</a> |
| Design Patterns in C# | Understanding design patterns | B005     | CodeBooks | 11/25/2019 12:00:00 AM | Fiction         | Michael | 6           | 6               | 11/21/2024 5:14:45 PM | images/books/tieng-anh.png               | pdf/Lab7.pdf | <a href="#">Edit</a><br><a href="#">Delete</a> |
| Web Programming       | Learn Web Programming         | B006     | WebBooks  | 12/25/2024 12:00:00 AM | Science Fiction | Jane    | 23          | 7               | 11/26/2024 5:14:45 PM | images/books/my-thuat-6.png              | pdf/Lab2.pdf | <a href="#">Edit</a><br><a href="#">Delete</a> |
| Mobile Programming    | Learn Mobile Programming      | B007     | TechBooks | 12/24/2024 12:00:00 AM | Programming     | Tom     | 9           | 8               | 11/26/2024 5:14:45 PM | images/books/shs-tieng-anh-6-tap-1-1.jpg | pdf/Lab1.pdf | <a href="#">Edit</a><br><a href="#">Delete</a> |
| Backend Programming   | Learn Backend Programming     | B008     | CodeBooks | 12/23/2024 12:00:00 AM | Science Fiction | Michael | 7           | 9               | 11/24/2024 5:14:45 PM | images/books/khoa-hoc-tu-nhien.png       | pdf/Lab6.pdf | <a href="#">Edit</a><br><a href="#">Delete</a> |
| Frontend Programming  | Learn Frontend Programming    | B009     | TechBooks | 12/27/2024 12:00:00 AM | Fiction         | Emily   | 8           | 5               | 11/12/2024 5:14:45 PM | images/books/am-nhac-6.png               | pdf/Lab3.pdf | <a href="#">Edit</a><br><a href="#">Delete</a> |

Figure 6. Admin Books CRUD (Show, Add, Edit, Delete)



Dashboard
Category
Author
Book
User
Loan

Search books...
Back To HomePage
Logout

### Create New Book

Title

Description

BookCode

Publisher

PublishedYear

CategoryId

AuthorId

TotalCopies

AvailableCopies

CreatedDate

*Figure 7. Admin Books CRUD (Add)*

Dashboard
Category
Author
Book
User
Loan

Search books...
Back To HomePage
Logout

### Edit Book

Title

Description

BookCode

Publisher

PublishedYear

CategoryId

AuthorId

TotalCopies

AvailableCopies

CreatedDate

Avatar

Pdf

Save

*Figure 8. Admin Books CRUD (Edit)*

### 4.2.4. Admin Book Category CRUD

The Admin Book Category CRUD interface allows for managing book categories. It includes functions to display category lists, add new categories, edit existing ones, and delete categories, ensuring organized and structured book classification.

Dashboard

Category

Author

Book

User

Loan

Search books...

Back To HomePage

Logout

Category

Create New

| Name            | Description                                       | CreatedDate           | UpdatedDate           | IsActive                 | Avatar                         |   |
|-----------------|---|-----------------------|-----------------------|--------------------------|--------------------------------|---|
| Programming     | Books on various programming languages            | 11/21/2024 5:13:17 PM | 11/21/2024 5:13:17 PM | <input type="checkbox"/> | /categories/image1.jpg         | <a href="#">Edit</a> <a href="#">Add</a> <a href="#">Delete</a> |
| Fiction         | Resources for web development                     | 11/21/2024 5:13:17 PM | 11/21/2024 5:13:17 PM | <input type="checkbox"/> | /categories/image2.jpg         | <a href="#">Edit</a> <a href="#">Add</a> <a href="#">Delete</a> |
| Science Fiction | Database management and development               | 11/21/2024 5:13:17 PM | 11/21/2024 5:13:17 PM | <input type="checkbox"/> | /categories/image3.jpg         | <a href="#">Edit</a> <a href="#">Add</a> <a href="#">Delete</a> |
| Software Design | Books on software design and patterns             | 11/21/2024 5:13:17 PM | 11/21/2024 5:13:17 PM | <input type="checkbox"/> | /categories/softwaredesign.jpg | <a href="#">Edit</a> <a href="#">Add</a> <a href="#">Delete</a> |
| Networking      | Resources for network administration and security | 11/21/2024 5:13:17 PM | 11/21/2024 5:13:17 PM | <input type="checkbox"/> | /categories/networking.jpg     | <a href="#">Edit</a> <a href="#">Add</a> <a href="#">Delete</a> |

Figure 9. Admin Book Category CRUD (Show, Add, Edit, Delete)

Dashboard

Category

Author

Book

User

Loan

Search books...

Back To HomePage

Logout

Create New Category

Name

Description

CreatedDate

mm/dd/yyyy --:-- --

UpdatedDate

mm/dd/yyyy --:-- --

☐ IsActive

Avatar

Create

Back to List

Figure 10. Admin Book Category CRUD (Add)

Dashboard

Category

Author

Book

User

Loan

Search books...

Back To HomePage

Logout

Edit Category

Name

Fiction

Description

Resources for web development

CreatedDate

11/21/2024 05:13:17.636 PM

UpdatedDate

11/21/2024 05:13:17.636 PM

☒ IsActive

Avatar

/categories/image2.jpg

Save

Back to List

Figure 11. Admin Book Category CRUD (Edit)

### 4.2.5. Admin Author CRUD

The Admin Author CRUD interface allows for managing author information. It includes functions to display author lists, add new authors, edit existing details, and delete authors from the database.

Dashboard

Category

Author

Book

User

Loan

Search books...

Back To HomePage

Logout

Author

Create New

| FirstName | LastName | DateOfBirth               | Biography                          | Nationality | Email  | Website   | CreatedDate              | IsActive                            | Avatar                      |   |
|-----------|----------|---------------------------|------------------------------------|-------------|--|---|--------------------------|-------------------------------------|-----------------------------|---|
| John      | Doe      | 6/15/1975<br>12:00:00 AM  | Expert in C# programming           | American    | <a href="mailto:johndoe@example.com">johndoe@example.com</a>               | <a href="https://johndoe.com">https://johndoe.com</a>               | 11/21/2024<br>5:13:37 PM | <input checked="" type="checkbox"/> | /authors/johndoe.jpg        | <div><div></div><div></div><div></div><div></div></div> |
| Jane      | Smith    | 2/28/1982<br>12:00:00 AM  | Specialist in web development      | British     | <a href="mailto:janesmith@example.com">janesmith@example.com</a>           | <a href="https://janesmith.com">https://janesmith.com</a>           | 11/21/2024<br>5:13:37 PM | <input checked="" type="checkbox"/> | /authors/janesmith.jpg      | <div><div></div><div></div><div></div><div></div></div> |
| Tom       | Brown    | 11/10/1979<br>12:00:00 AM | Database and EF Core expert        | Canadian    | <a href="mailto:tombrown@example.com">tombrown@example.com</a>             | <a href="https://tombrown.com">https://tombrown.com</a>             | 11/21/2024<br>5:13:37 PM | <input checked="" type="checkbox"/> | /authors/tombrown.jpg       | <div><div></div><div></div><div></div><div></div></div> |
| Emily     | Clark    | 8/8/1985<br>12:00:00 AM   | LINQ and data querying expert      | Australian  | <a href="mailto:emilyclark@example.com">emilyclark@example.com</a>         | <a href="https://emilyclark.com">https://emilyclark.com</a>         | 11/21/2024<br>5:13:37 PM | <input checked="" type="checkbox"/> | /authors/emilyclark.jpg     | <div><div></div><div></div><div></div><div></div></div> |
| Michael   | Johnson  | 3/21/1980<br>12:00:00 AM  | Expert in software design patterns | German      | <a href="mailto:michaeljohnson@example.com">michaeljohnson@example.com</a> | <a href="https://michaeljohnson.com">https://michaeljohnson.com</a> | 11/21/2024<br>5:13:37 PM | <input checked="" type="checkbox"/> | /authors/michaeljohnson.jpg | <div><div></div><div></div><div></div><div></div></div> |

Figure 12. Admin Author CRUD (Show, Add, Edit, Delete)

Dashboard

Category

Author

Book

User

Loan

Search books...

Back To HomePage

Logout

New Author

FirstName

LastName

DateOfBirth

mm/dd/yyyy --:-- --

Biography

Nationality

Email

Website

CreatedDate

mm/dd/yyyy --:-- --

☐ IsActive

Avatar

Create

Back to List

Figure 13. Admin Author CRUD (Add)

Dashboard
Category
Author
Book
User
Loan

Search books...
Back To HomePage
Logout

### Edit Author

FirstName

LastName

DateOfBirth

Biography

Nationality

Email

Website

CreatedDate

IsActive

☒

Avatar

Save
Back to List

Figure 14. Admin Author CRUD (Edit)

## 4.2.6. Admin Loans CRUD

The Admin Loans CRUD interface allows for managing loans information. It includes functions to display loans lists, add new loans, edit existing details, and delete loans from the database.

Dashboard
Category
Author
Book
User
Loan

Search books...
Back To HomePage
Logout

### Loan

[Create New](#)

| User                              | Book | LoanDate              | DueDate               | ReturnDate            | Status |   |
|-----------------------------------|------|-----------------------|-----------------------|-----------------------|--------|---|
| <a href="#">alice@example.com</a> | B001 | 11/12/2024 7:28:00 PM | 11/22/2024 7:28:00 PM | 11/26/2024 7:28:00 PM | 1      | <a href="#">Edit</a> <a href="#">Delete</a> |
| <a href="#">bob@example.com</a>   | B003 | 11/7/2024 7:28:00 PM  | 11/30/2024 7:28:00 PM | 12/6/2024 7:28:00 PM  | 1      | <a href="#">Edit</a> <a href="#">Delete</a> |
| <a href="#">david@example.com</a> | B004 | 11/3/2024 7:28:00 PM  | 11/18/2024 7:28:00 PM | 11/25/2024 7:28:00 PM | 0      | <a href="#">Edit</a> <a href="#">Delete</a> |
| <a href="#">eva@example.com</a>   | B005 | 11/23/2024 7:28:00 PM | 11/25/2024 7:28:00 PM | 11/28/2024 7:28:00 PM | 1      | <a href="#">Edit</a> <a href="#">Delete</a> |
| <a href="#">cathy@example.com</a> | B003 | 11/1/2024 7:29:00 PM  | 11/2/2024 7:29:00 PM  | 11/3/2024 7:29:00 PM  | 1      | <a href="#">Edit</a> <a href="#">Delete</a> |

Figure 15. Admin Loans CRUD (Show, Add, Edit, Delete)

Dashboard
Category
Author
Book
User
Loan

Search books...
Back To HomePage
Logout

### Create New Loan

UserId

BookId

LoanDate

DueDate

ReturnDate

Status

Create
Back to List

Figure 16. Admin Loans CRUD (Add)

Dashboard

Category

Author

Book

User

Loan

Search books...

Back To HomePage

Logout

### Edit Loan

UserId  
bob@example.com

BookId  
B003

LoanDate  
11/07/2024 07:28 PM

DueDate  
11/30/2024 07:28 PM

ReturnDate  
12/06/2024 07:28 PM

Status  
1

Save

[Back to List](#)

*Figure 17. Admin Loans CRUD (Edit)*

#### 4.2.7. Home Page

The Home Page is the main entry point for users. It presents the layout and features of the homepage, designed to provide a friendly and engaging experience for browsing and accessing different sections of the bookstore.

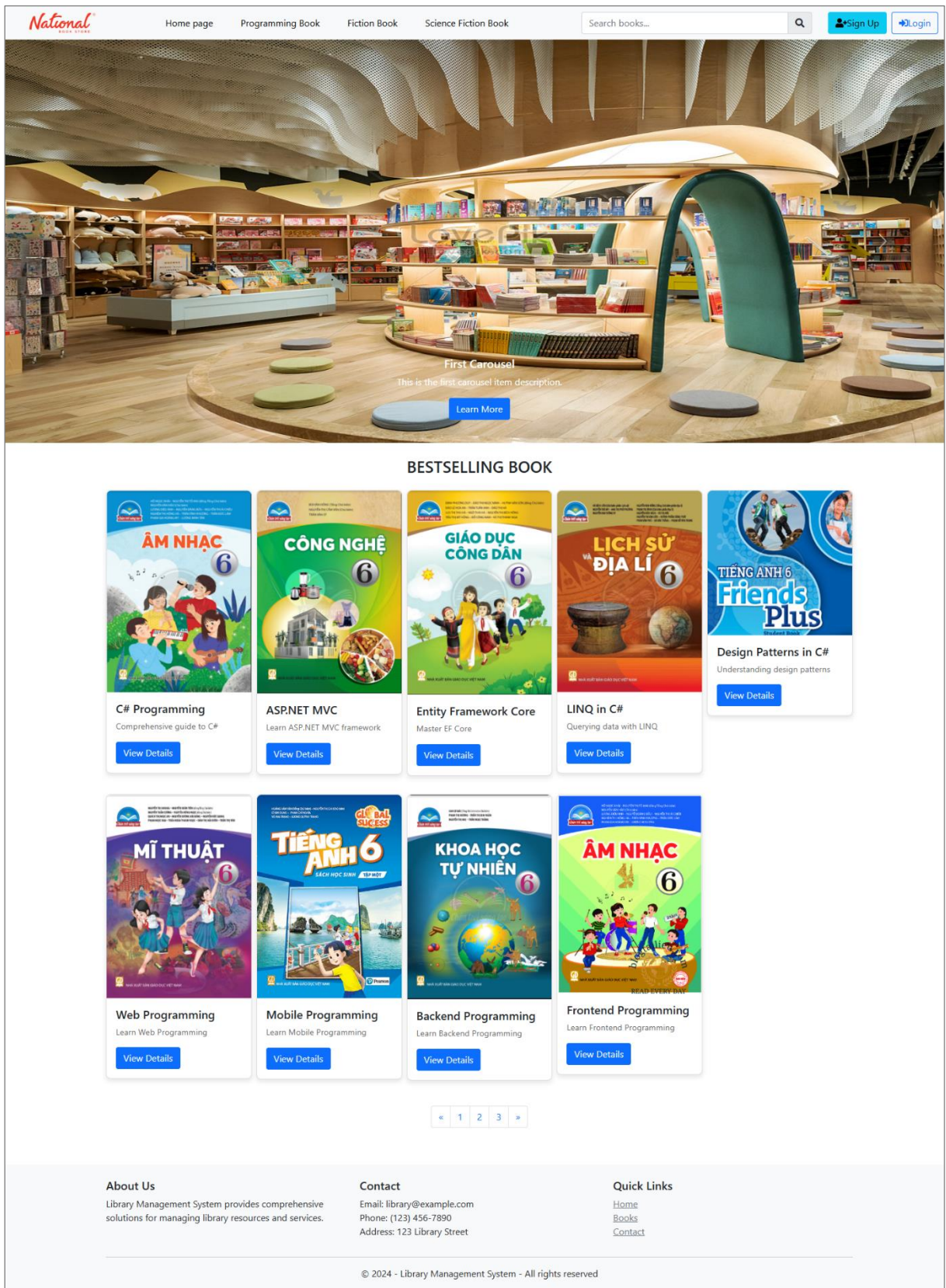


Figure 18. Home Page

### 4.2.8. Login/SignUp Page

The Home Page is the main entry point for users. It presents the layout and features of the homepage, designed to provide a friendly and engaging experience for browsing and accessing different sections of the bookstore.

National

Home pageProgramming BookFiction BookScience Fiction Book

Search books...

Sign UpLogin

Create your Account

Sign up to get started!

Google

Facebook

or sign up with email

Email

Password

Confirm Password

Sign Up

Already have an account? [Log in](#)

About Us

Library Management System provides comprehensive solutions for managing library resources and services.

Contact

Email: library@example.com  
Phone: (123) 456-7890  
Address: 123 Library Street

Quick Links

[Home](#)  
[Books](#)  
[Contact](#)

© 2024 - Library Management System - All rights reserved

National

Home pageProgramming BookFiction BookScience Fiction Book

Search books...

Sign UpLogin

Log in to your Account

Welcome back! Select a method to log in:

Google

Facebook

or continue with email

Email

Password

☐ Remember me

[Forgot Password?](#)

Login

Don't have an account? [Create an account](#)

About Us

Library Management System provides comprehensive solutions for managing library resources and services.

Contact

Email: library@example.com  
Phone: (123) 456-7890  
Address: 123 Library Street

Quick Links

[Home](#)  
[Books](#)  
[Contact](#)

© 2024 - Library Management System - All rights reserved

Figure 19. Login/SignUp Page

4.2.9. Book List Page

The Book List Page displays books in a detailed list format. It presents a structured view with comprehensive information about each book, including the title, author, price, and a brief description, making it easy for users to browse and select books based on their preferences.

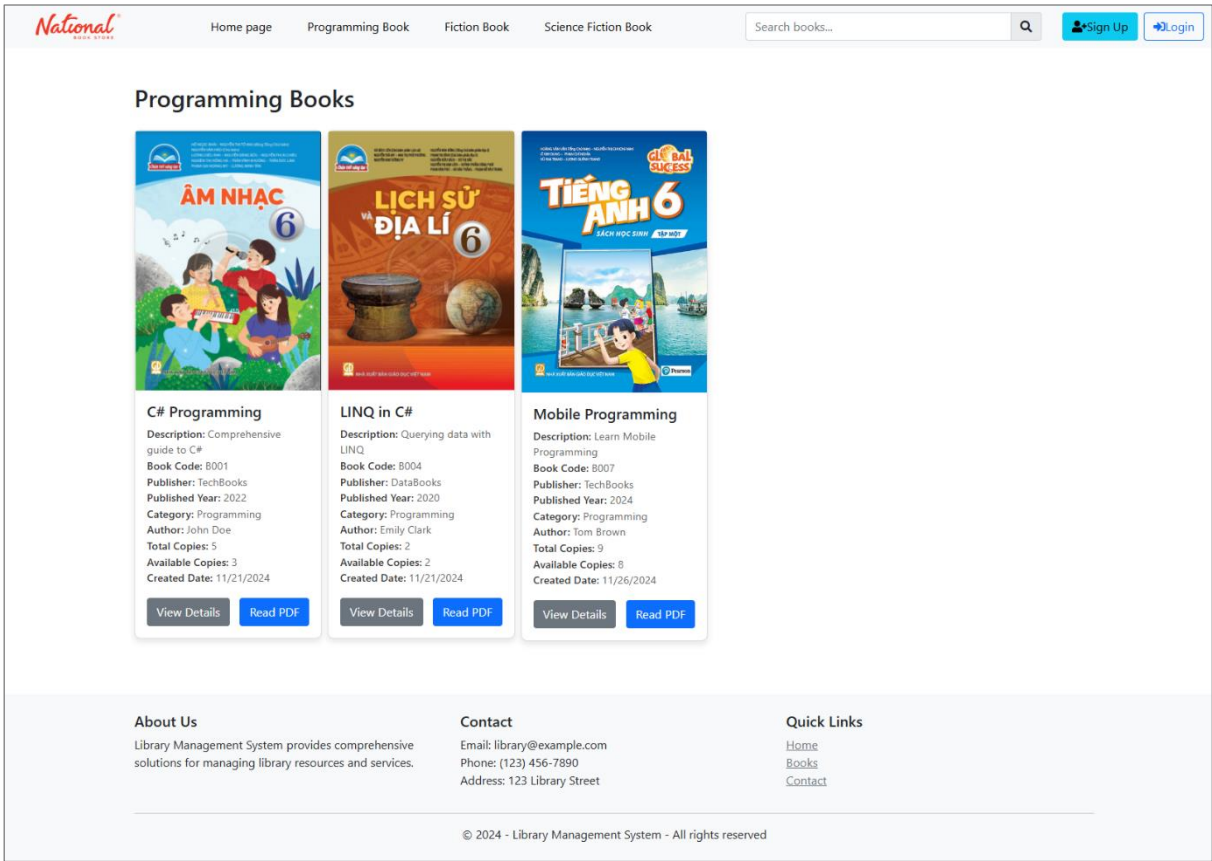


Figure 20. Book List Page



4.2.10. Book Details Page

The Book Detail Page provides an in-depth view of a selected book. It includes detailed information such as the book's title, author, price, description, and reviews. This page also features options for adding the book to the cart, offering a comprehensive overview to assist users in their purchasing decisions.

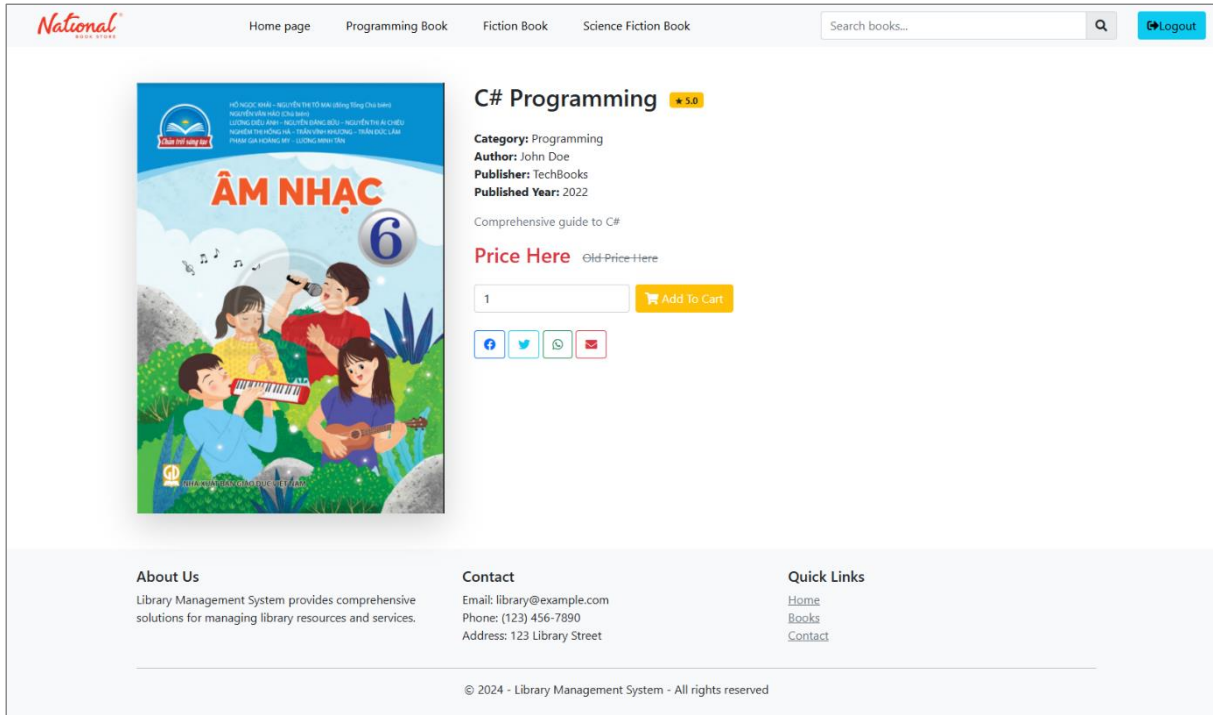


Figure 21. Book Details Page

## CHAPTER 5. CONCLUSION AND FUTURE WORKS

### 5.1. Conclusion

In conclusion, the primary objective of this project was to develop a robust Library Management System, focusing on providing administrators with comprehensive access to all CRUD tasks and implementing essential functionalities for library users. The project has successfully integrated CRUD functionalities for managing books, authors, and categories, enabling administrators to efficiently handle library resources. Additionally, features such as book lending and returning, and user authentication with role-based authorization, have enhanced the system's functionality and usability.

Through this project, we have gained valuable insights and skills. We applied the software development process comprehensively, starting from requirements analysis, through system design, to implementation and testing. This hands-on experience provided a deeper understanding of building web applications for real-world scenarios, highlighting the importance of meeting both functional and non-functional requirements.

Moreover, we recognized the critical role of User Interface (UI) and User Experience (UX) in ensuring the success of any application. A significant emphasis on these aspects ensured that the platform is not only functionally robust but also user-friendly and visually appealing. By integrating various technologies like ASP.NET Core MVC, AJAX, and Entity Framework, we developed a practical and comprehensive application that meets the needs of both administrators and users.

### 5.2. Future works

In the future, we aim to focus on potential upgrades and expansions for the Library Management System. Key areas for development include:

- **Enhancing Security:** Our team are considering the adoption of an additional framework to meet optimal standard as well as strengthen user protection.
- **Changing Running Environment:** We plan to deploy the website to a real-time environment.
- **Refining Time Management Strategies:** Embracing agile development practices and effective time allocation strategies will help expedite further development and maintenance of the system.

By implementing these improvements, we aim to enhance user experience, streamline administrative operations, and ensure the sustained growth and adaptability of the Library Management System.