UNIVERSITI POLY-TECH MALAYSIA

Name: **MUHAMMAD HAIKAL BIN CHE HAMDAM**	
Id Number: **AM2211012880**	

Lecturer: **MOHD AKMAL BIN MOHD AZMER**	**Lab group / Tutorial group / Tutor (if applicable)**
Course and Course Code: SWC3503	**Submission Date:**
Assignment No. / Title FINAL PROJECT	**Extension & Late submission:** **Allowed / Disallowed**

Assignment type **INDIVIDUAL**	**40% of Assignment Mark**	**Returning Date:**

Penalties:

1. **10% of the original mark will be deducted for every one week period after the submission date**
2. **No work will be accepted after two weeks of the deadline**
3. **If you were unable to submit the coursework on time due to extenuating circumstances you may be eligible for an extension**
4. **Extension will not exceed one week**

Declaration: I/we the undersigned confirm that I/we have read and agree to abide by these regulations on plagiarism and cheating. I/we confirm that this piece of work is my/our own. I/we consent to appropriate storage of our work for checking to ensure that there is no plagiarism/ academic cheating.

   Signature: HAIKAL

This section may be used for feedback or other information

Table of Contents

1. Introduction

The objective of the Library Management System project is to create a computerized system that efficiently oversees and synchronizes the many duties and operations carried out on a regular basis in a library. The present project encompasses certain elements that are often absent in conventional library management systems, such as the inclusion of user login functionalities and an administrator login capability. Additionally, the system has a feature for administrative login, which gives the administrator the capability to oversee the whole of the system. In addition, the portal offers a feature that allows students to access a comprehensive list of books allocated to them, along with the corresponding dates of issue and return, upon login into their accounts. The primary objective of this project is to enhance the efficiency of library management by minimizing the reliance on human labor and offering assistance to both students and staff members.

BACKGROUND OF PROJECT

Library Management System is an application which refers to library systems which are generally small or medium in size. It is used by librarian to manage the library using a computerized system where he/she can record various transactions like issue of books, return of books, addition of new books, addition of new students etc.

Books and student maintenance modules are also included in this system which would keep track of the students using the library and also a detailed description about the books a library contains. With this computerized system there will be no loss of book record or member record which generally happens when a non-computerized system is used.

SYSTEM OBJECTIVES
1. Improvement in control and performance
- The system is developed to cope up with the current issues and problems of library. The system can add user, validate user and is also bug free.
2. Save more time
- A librarian can save precious time by searching records using only a few mouse clicks and search terms.
3. Reduce cost
- The cost of maintaining the library will go down overall after the computerized system is put in place since less labor will be needed to maintain it.

Chosen Programing

HTML

- HTML is the major markup language for building web pages and other information that can be viewed in a web browser.HTML is written in the form of HTML components consisting of tags contained in angle brackets (like <html>), within the web page text. HTML tags most usually appear in pairs like <h1> and </h1>, while certain tags indicate empty items and hence are unpaired, for example <img>. Referred to as opening and closing tags, the start and finish tags of a pair are the first and second tags, respectively. In between these tags web designers may add text, extra tags, comments, and other sorts of text-based content. The goal of a web browser is to read HTML texts and assemble them into visual or audio web pages. The browser does not show the HTML tags, but uses the tags to read the content of the page. The fundamental components of every website are HTML elements. HTML may be used to build interactive forms by embedding objects and graphics. By indicating structural semantics for text elements like headers, paragraphs, lists, links, quotations, and other objects, it offers a way to generate structured texts. It may embed programs written in languages such as JavaScript which influence the functionality of HTML web pages.

CSS

- A style sheet language called Cascading Style Sheets (CSS) is used to specify how a document produced in a markup language should look and be formatted. The language may be used to any type of XML document, including plain XML, SVG, and XUL, but it is most frequently used to style web pages and interfaces written in HTML and XHTML. CSS style sheets are a fundamental component of the web, and they define the presentation of nearly every webpage. The main purpose of CSS is to make it possible to separate document presentation—which includes things like layout, color scheme, and font selection—from text. This division can increase the flexibility and control in the specification, as well as the accessibility of the material.

Database MYSQL

- MYSQL is utilized as a database because it is straightforward to manage and retrieve entries using English-language queries that are easy to construct and comprehend.

PHP

- PHP is a server-side scripting language that may be used for general-purpose programming as well as web development. Originally standing for Personal Home Page, PHP is A web server equipped with a PHP processor module decodes Hypertext Preprocessor, a recursive backronym code, and produces the following webpage: Instead of requiring the user to open an external file in order to handle data, PHP instructions can be directly integrated within an HTML source document. It may now be used in independent graphical apps and has expanded to include a command-line interface. The PHP License governs the use of PHP, which is free software. PHP is available for free on the majority of web servers and may be used as a stand-alone shell on nearly all platforms and operating systems, which is used to develop databases, is utilized to create the back end.

2. Design and architecture

Admin case diagram
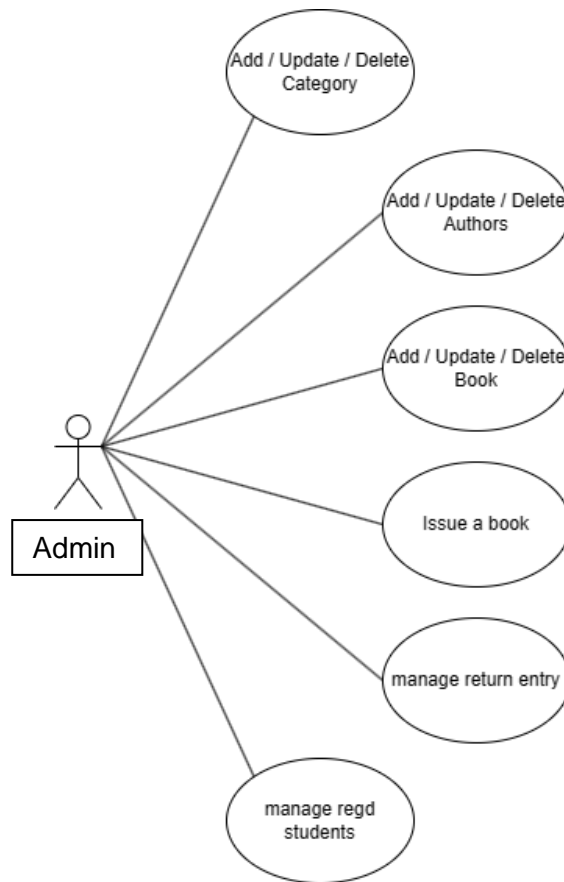
Student case diagram
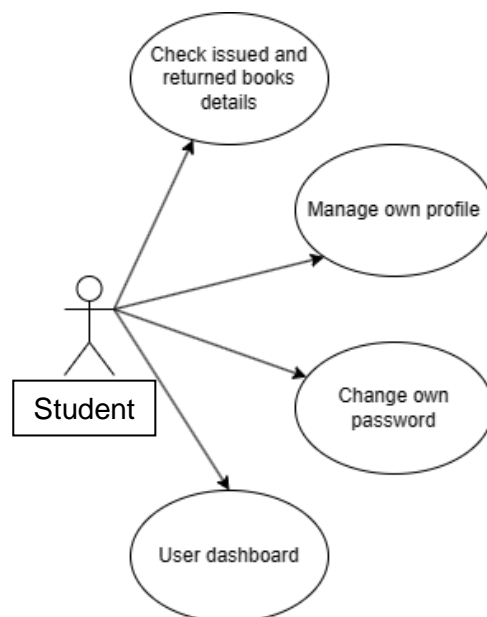
Issue book case diagram
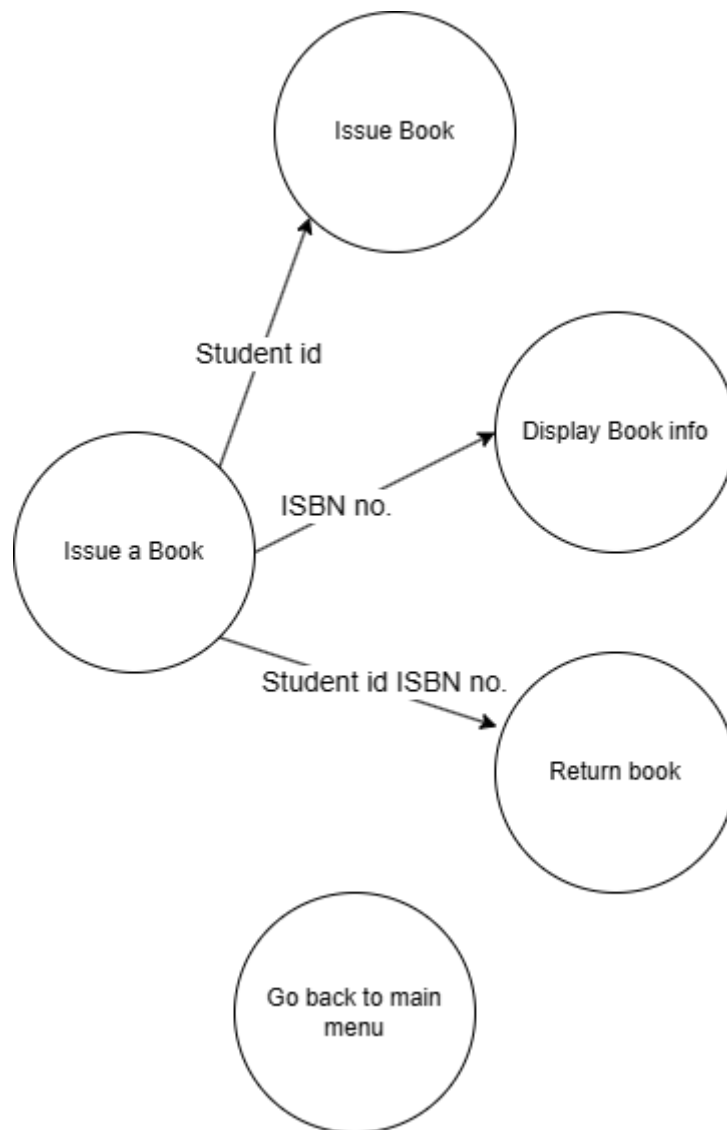
1. Student Requests Book: This process initiates the book issuance workflow. The student presents their student ID and the ISBN of the desired book to the librarian.
2. Verify Student Eligibility: This process ensures that the student is authorized to borrow books. The librarian checks if the student has an active library membership and any outstanding fines or overdue books.
3. Check Book Availability: This process confirms whether the requested book is available for loan. The librarian accesses the library's inventory system to determine the book's status.
4. Issue Book: If both the student is eligible and the book is available, the librarian proceeds to issue the book. They scan the book's barcode and update the library's records to reflect the book's checked-out status.

Flow Diagram

```
                    ( Start )
                        |
                        v
                 [ take choice ]  <-----------------------------+
                        |                                       |
                        v                                       |
                    < Choice >                                  |
                        |                                       |
          +-------------+-------------+                         |
          |                           |                         |
          v                           v                         |
   [ Admin Login ]               [ Exit ]                       |
          |                                                     |
          v                                                     |
   [ Enter password ]                                           |
          |                                                     |
          v                                                     |
      < Match >                                                 |
          |                                                     |
          v                                                     |
   [ take choice ]                                              |
          |                                                     |
          v                                                     |
     < Choice >                                                 |
          |                                                     |
   +------+--------+----------+------------+-----------+        |
   v      v        v          v            v           |
[Book  [Manage  [Feedback] [Student    [Logout]--------+
 detail] Book]              Detail]
```
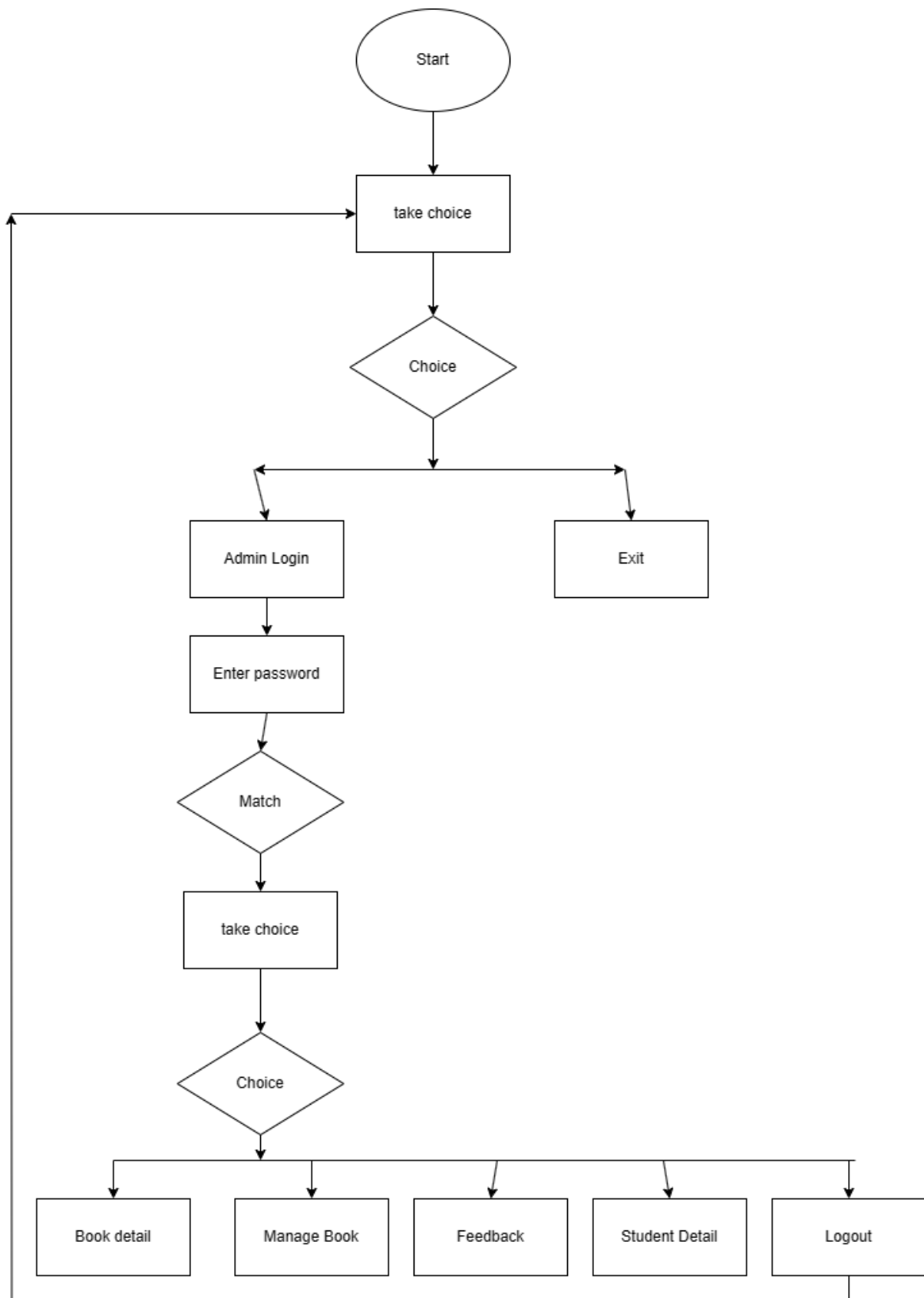
Class Diagram
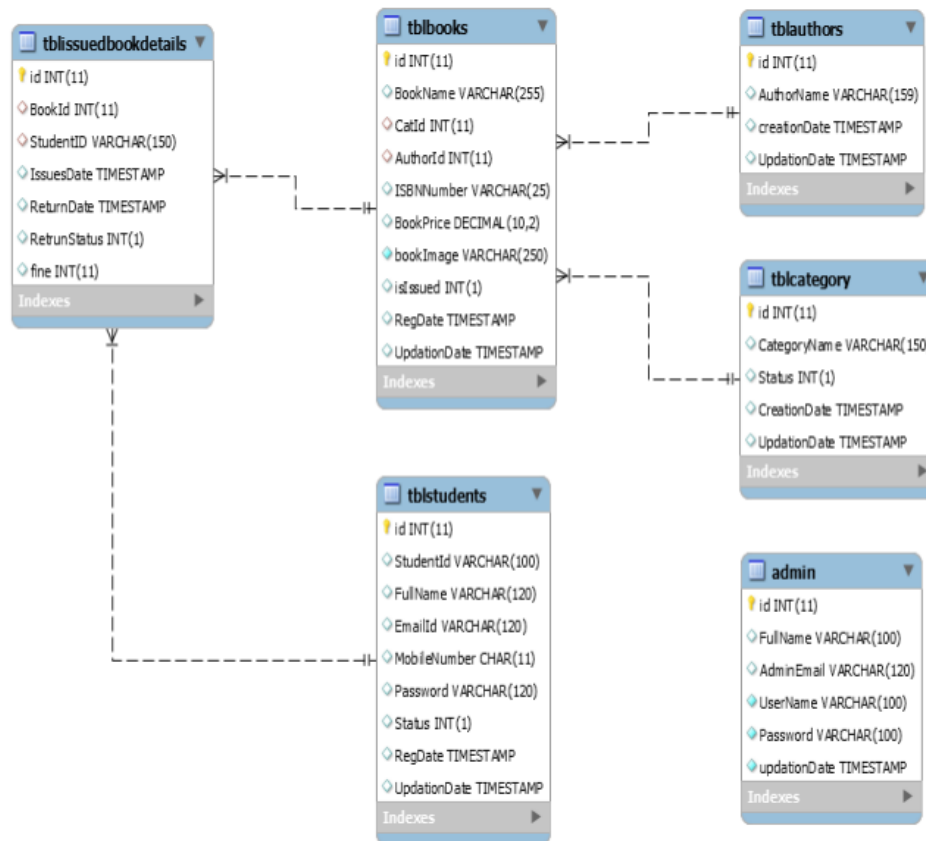
Class for Library Table

3. Contents

User Login

This feature is used by the user to login into system. They are required to enter the user id and password before they are allowed to enter the system. The user id and password will be verified and if invalid id is their user is allowed to not enter the system.

Functional requirements

- user id is provided when they register.
- The system must only allow users with valid id and password to enter the system.
- The system performs an authorization process which decides what user level can access to.
- The user must be able to logout after they finished using the system.

Register New User

This feature can be performed by all users to register new user to create account.

Functional requirements

- System must be able to verify information
- System must be able to delete information if information is wrong

Features

Admin:

- Admin can add/update/ delete category.
- Admin can add/update/ delete author.
- Admin can add/update/ delete books.
- Admin can issue a new book to student and also update the details when student return book.
- Admin can search student by using their student ID.
- Admin can also view student details.
- Admin can change own password.

Students:

- Student can register yourself and after registration they will get studentid
- After login student can view own dashboard.
- Student can update own profile.
- Student can view issued book and book return date-time
- Student can also view the available books in the library
- Student can also change own password
- Student can also recover own password

Database System

1. Registering New Books:

- New books are registered in the system by inserting records into the tblbooks table
- When a new book is added, the system captures information such as BookName, CatId (Category ID), AuthorId (Author ID), ISBNNumber, BookPrice, bookImage, and other details.
- The RegDate field captures the registration date, and the UpdationDate field is updated whenever the record is modified.

2. Viewing Books:

- To view books, you can perform SELECT queries on the tblbooks table.
- You can retrieve information about books, such as BookName, AuthorName, CategoryName, ISBNNumber, BookPrice, and other details by joining relevant tables (tblbooks, tblauthors, tblcategory).

3. Updating Existing Books:

- Existing books can be updated by executing UPDATE queries on the tblbooks table.
- You can modify information such as BookName, CatId, AuthorId, ISBNNumber, BookPrice.
- The UpdationDate field is automatically updated to reflect the timestamp of the last modification.

4. Deleting Existing Books:

- Books can be deleted from the system by executing DELETE queries on the tblbooks table.
- Deleting a book will remove the corresponding record from the database.

Role Based

The structure and sample data for several tables in a database connected to a library system are contained in the supplied SQL dump. admin, tblauthors, tblbooks, tblcategory, tblissuedbookdetails, and tblstudents are among the tables. The structure of the admin table and maybe other associated tables must be examined in order to comprehend how the role-based account for the system functions.

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action		
1	id 🔑	int(11)			No	None		AUTO_INCREMENT	🖉 Change	⊖ Drop	More
2	FullName	varchar(100)	latin1_swedish_ci		Yes	NULL			🖉 Change	⊖ Drop	More
3	AdminEmail	varchar(120)	latin1_swedish_ci		Yes	NULL			🖉 Change	⊖ Drop	More
4	UserName	varchar(100)	latin1_swedish_ci		No	None			🖉 Change	⊖ Drop	More
5	Password	varchar(100)	latin1_swedish_ci		No	None			🖉 Change	⊖ Drop	More
6	updationDate	timestamp			No	0000-00-00 00:00:00		ON UPDATE CURRENT_TIMESTAMP()	🖉 Change	⊖ Drop	More

Admin Structure

information about administrators appears to be stored in the admin database, and each administrator has a distinct id. Password and UserName are the two main elements that are important for role-based access. The hashed password is likely stored in the Password field, and the UserName is utilized as the login username.

Roles
- Although roles aren't stated clearly in the existing structure, a role-based access control (RBAC) system would usually include a mechanism to link users (administrators) to certain roles. Admin and student are two examples of roles that might exist.

Access Control
- Access to certain system components, such managing students and books, would be restricted according to the tasks that each administrator was given. This control might be included into the application logic, allowing or restricting particular actions according to the role of the administrator who is currently signed in.

Authentication
- Password and UserName fields are probably used for authentication. The system compares these credentials to the values kept in the admin table when an administrator signs in. The user gets access if they match.

Authorization
- Following authentication, the system would look up the administrator's position to see what tasks they are allowed to carry out. Admin may have complete system access, admin may have more access, and a 'Librarian' may be allowed to handle books and provide book information.

Implementation security

1. Database Security

User Authentication and Authorization:
- For database users, be sure to use secure and distinct passwords.
- Refrain from applying default passwords or database users.
- Allocate the bare minimum of rights to every database user. For instance, just the essential CRUD (Create, Read, Update, Delete) rights should be granted to the application accessing the database.

Secure Password Storage:
- Continually change credentials and passwords for databases.
- Use more secure hashing methods for your passwords.

2. PHP Security:

Secure Database Connection:
- To mitigate the risk of SQL injection attacks, it is recommended to use parameterized queries or prepared statements.
- It is essential to refrain from using user input directly in SQL queries. Instead, it is crucial to sanitize and verify inputs to ensure data integrity and prevent any security vulnerabilities.
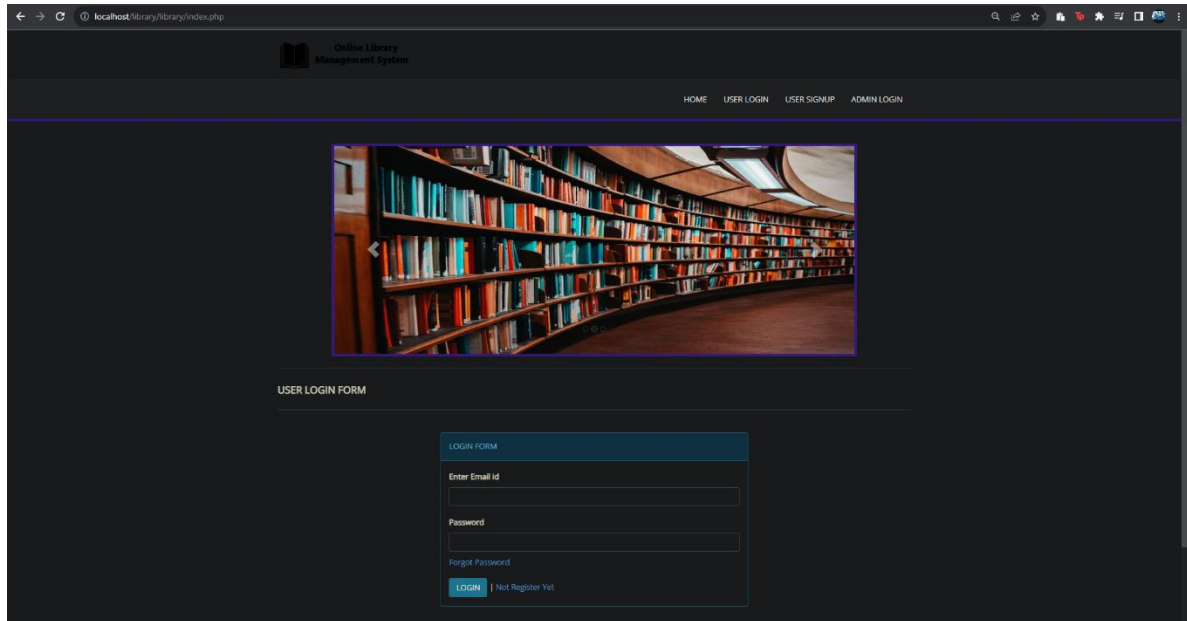
Error Handling:
- In the production environment, it is advisable to either disable error reporting or redirect error logs to a secure destination.
- In order to safeguard sensitive information, it is recommended to provide consumers with general error messages instead than providing specific details.

4. Testing

Student Testing
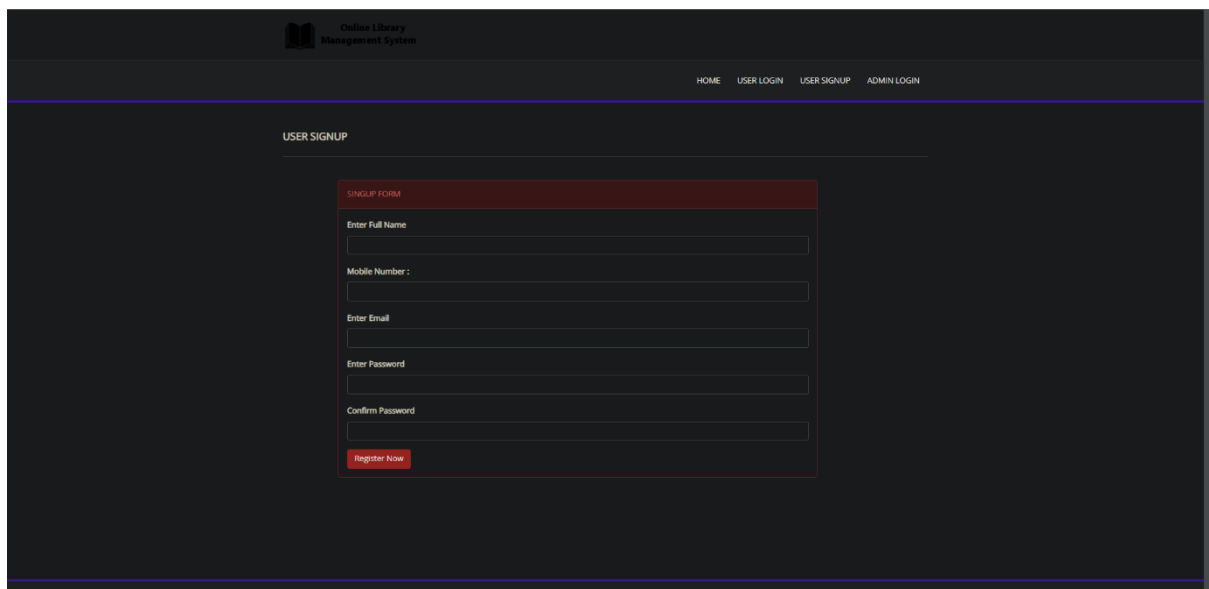
1. Home Page

Show homepage that user need to enter email and password

2. User Signup

The interface show user signup

3. User Dashboard
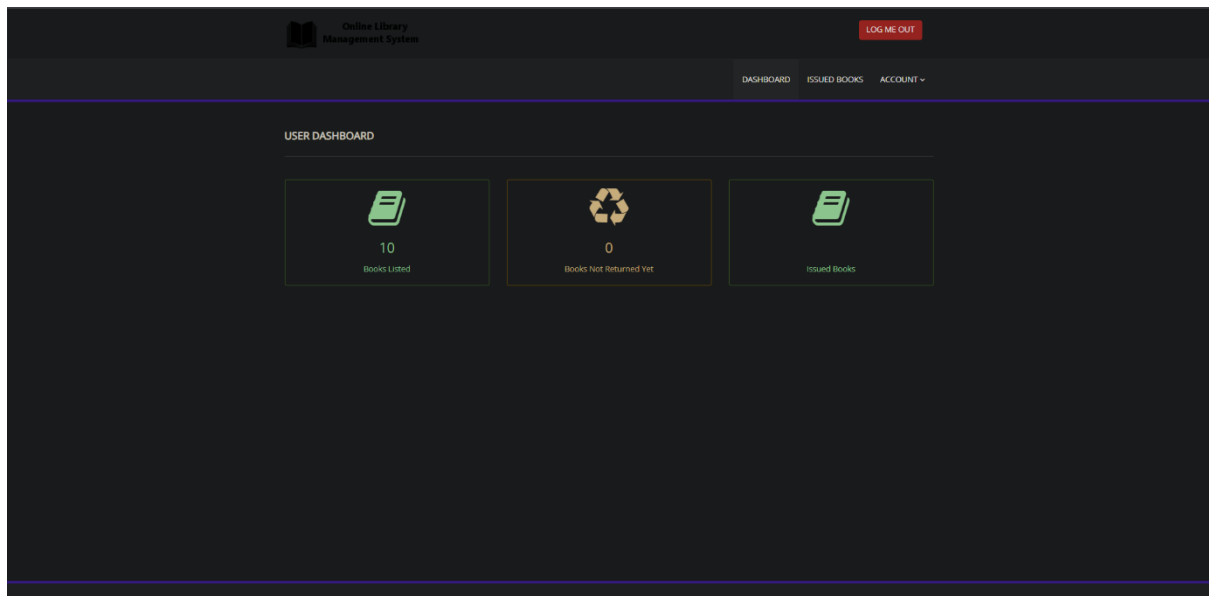
User dashboard show booklisted, books not returned and issued books

4. User Profile
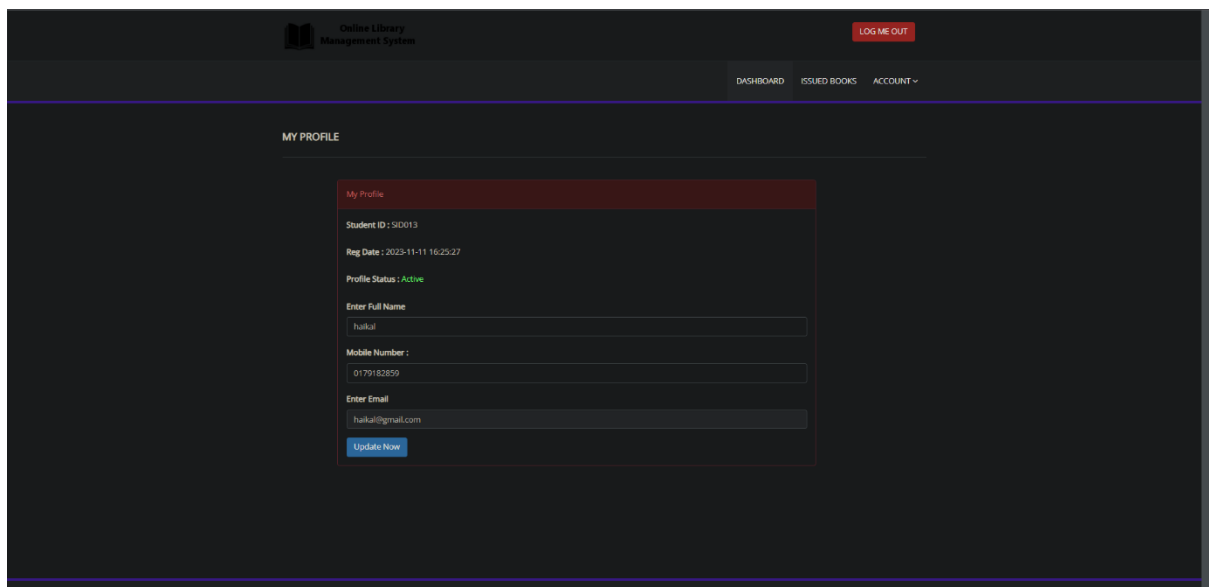
Interface show user profile such as full name, mobile number and email address

5. Change Password

Interface show user can change password anytime

6. Listed Books

Interface show All listed book
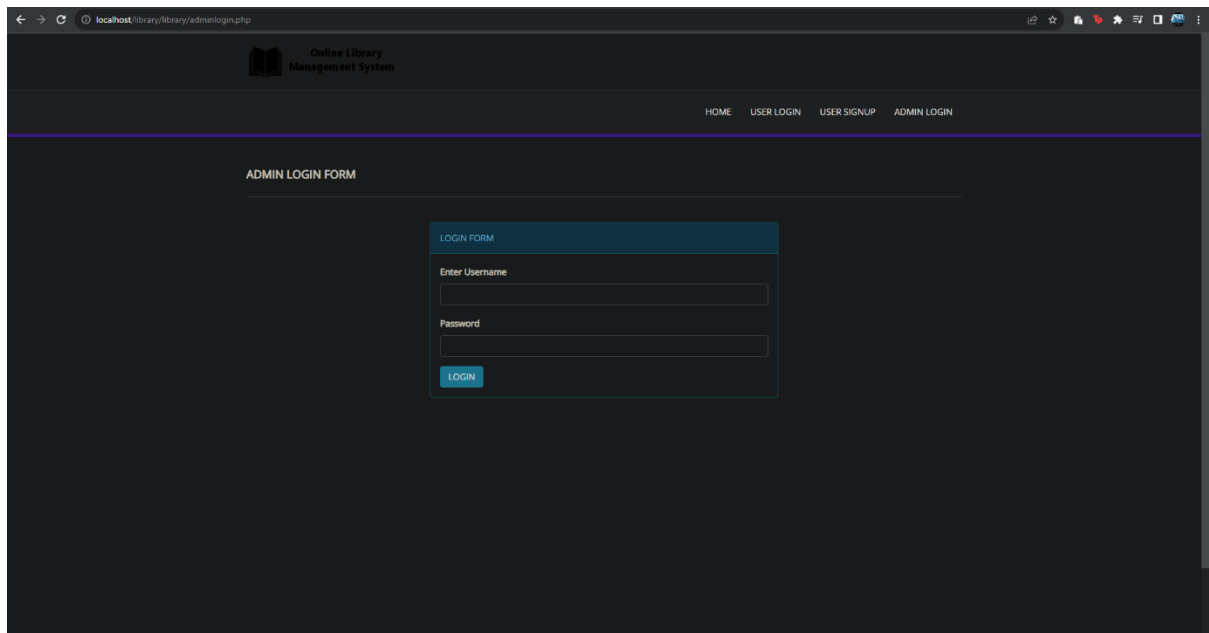
7. Issued Books
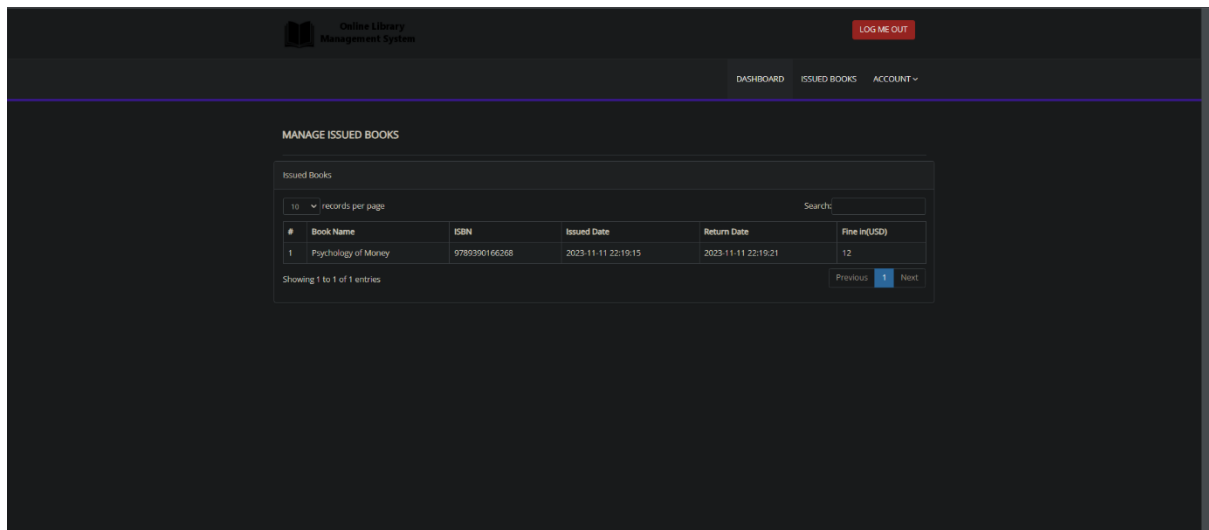
Interface show issued book report

Admin Testing
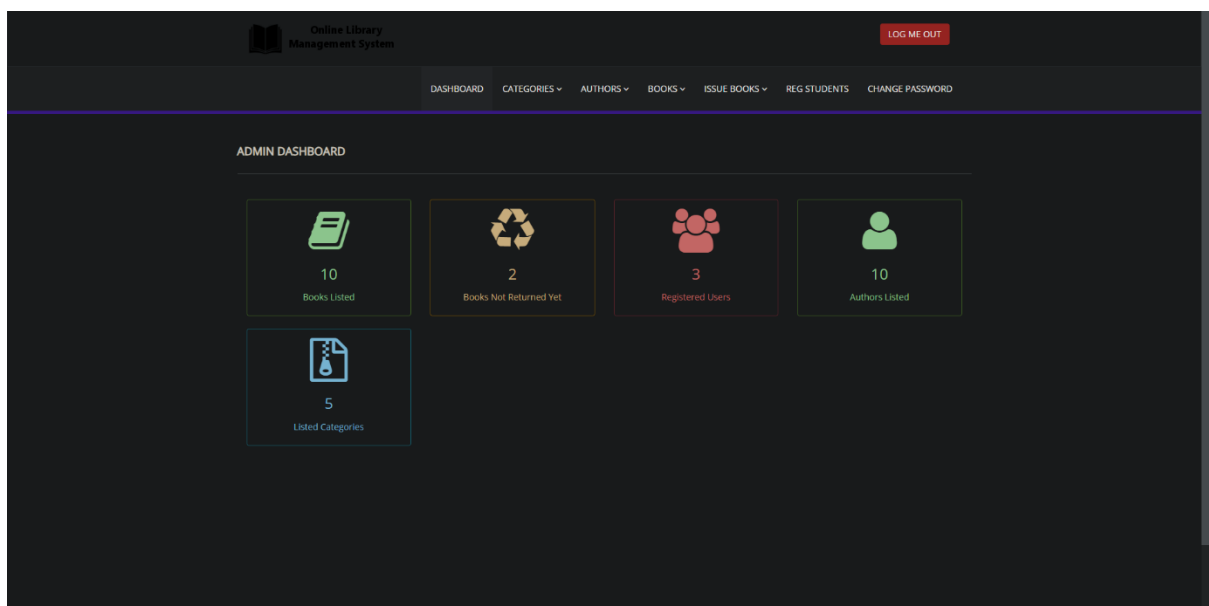
1. Admin Login

Admin login Homepage

2. Admin Dashboard

Admin dashboard that show all such as booklisted, books not returned, registered user, authors listed and listed categories

3. Add Category

Interface show admin can add category

4. Manage Category

Interface show admin can manage category such as delete and edit category

5. Add Author

Admin can add author for new book

6. Manage Author

Interface show all author with all function such as edit and delete author

7. Add Books

Admin can add new book and need to fill all the information book

8. Manage Books

Interface show admin can edit or delete books

9. Issue New Books

Interface show that admin check the issue new books

10. Manage Issue Books

Admin can check all history report about issue books

11. Manage Register Students
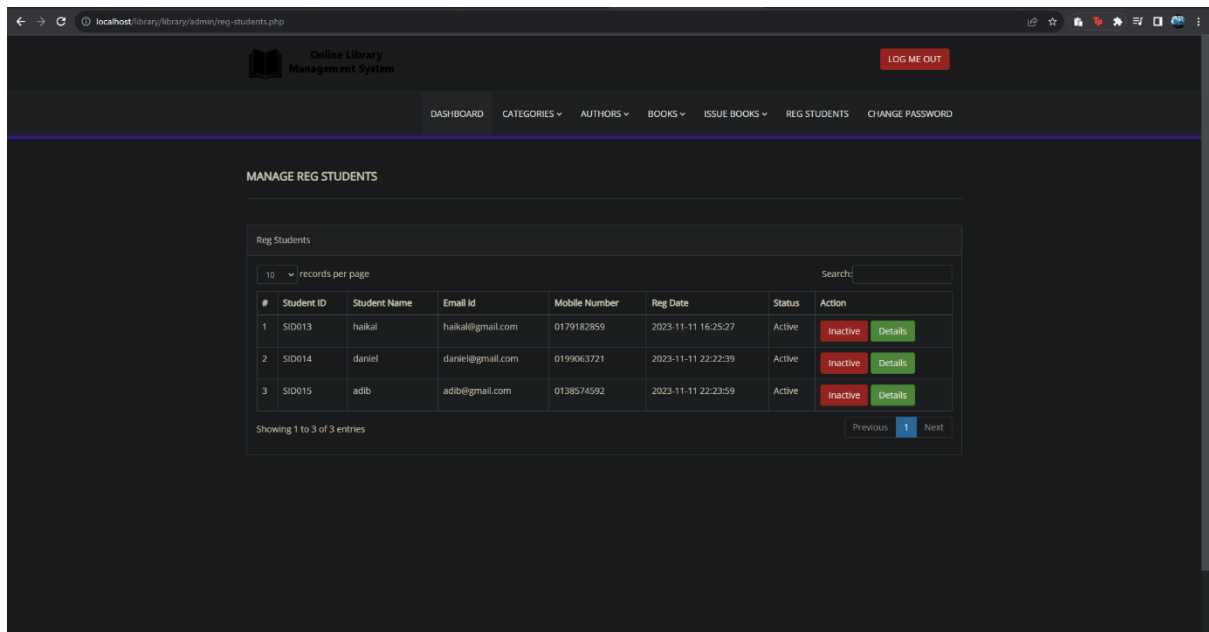
The interface show admin can inactive student id or check details about user

12. Admin change Password
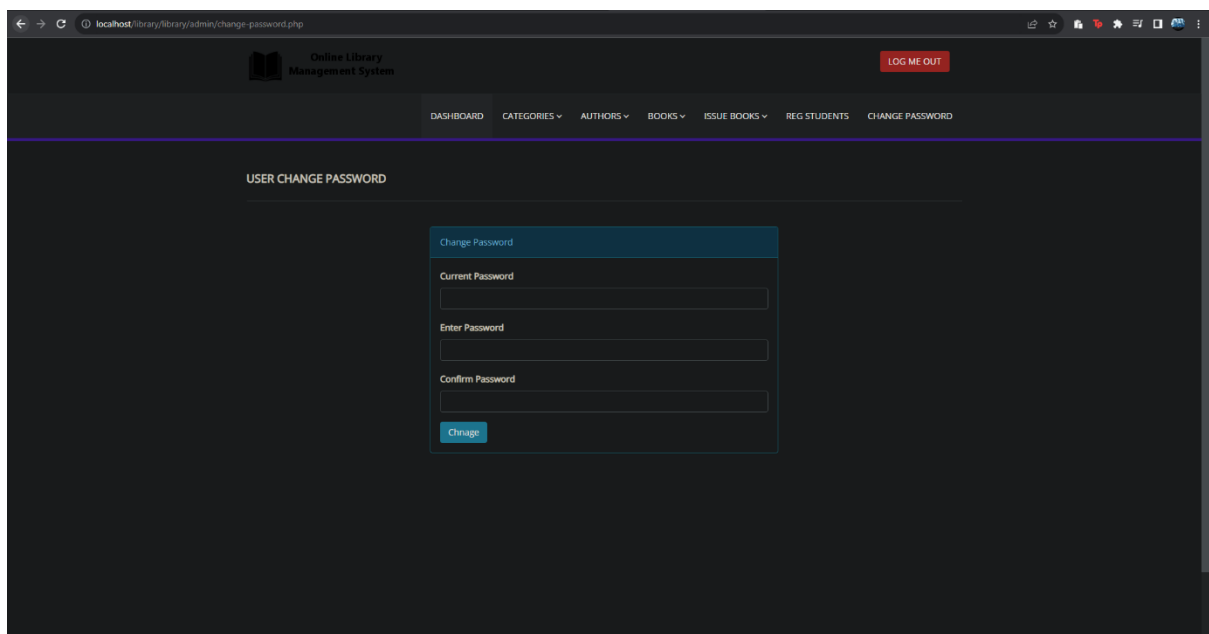
The interface show that admin can change old password to new password

Test For the admin

- The purpose of this form is to facilitate the login process for the system administrator. In this process, the user is prompted to input their username and password. If both credentials are accurate, access to the administration page is granted. However, if any of the provided data is incorrect, the user is routed back to the login page and prompted to re-enter their username and password.

- The student account addition section allows the administrator to verify the student details obtained from the academic information system. Only after the verification process, the administrator can proceed to add the student details to the main library database. This section includes both an "add" and a "delete" button. When the user clicks the "add" button, the student data will be added to the student database. Conversely, if the user clicks the "delete" button, the corresponding student data will be removed from the database.

- The book addition feature allows the administrator to input book data and then add them to the primary book table. Additionally, the administrator has the capability to examine requests for books.

Test for Student login

- The purpose of this form is to facilitate the login process for students. The user is required to input the library ID, username, and password. If all the provided information is accurate, the student login page will be accessible. However, if any of the data entered is incorrect, the user will be sent back to the login page and prompted to re-enter the library ID, username, and password.

- The purpose of this form is to facilitate the creation of new user accounts. If a student fails to complete the form in its entirety, they will be prompted to fill out all required fields. Once the student successfully completes the form, they will be redirected to a page that displays a message indicating that their account creation is pending confirmation. It is important to note that the student's data will only be added to the system by an administrator after it has been verified.

5. Reflective conclusion

A web system use of secure coding is essential for protecting sensitive data and thwarting illegal access. A thorough knowledge of potential dangers is made possible by the application of the STRIDE threat assessment model, which directs the creation of strong security measures. By ensuring that various users have the proper rights, role-based access management helps to prevent possible abuse. A more robust system is made possible by input validation and output encoding, which guard against typical vulnerabilities like SQL injection and cross-site scripting. Additional levels of protection are added by preventing unauthorized access to admin pages and by putting in place suitable authentication and permission procedures. Mechanisms for managing errors and recording them improve the system's capacity to identify problems and take immediate action.

The efficacy of these security measures is confirmed by secure testing, giving users confidence in the system's resistance to prospective threats. In conclusion, preserving the security and integrity of data requires a well-thought-out and securely developed online system. To maintain a safe and resilient application and respond to changing security threats, constant monitoring, upgrades, and adherence to best practices are required.

6. Reference

1. Admin. (2022, May 4). *10 secure coding practices you can implement now*. Code Signing Store. https://codesigningstore.com/secure-coding-practices-to-implement

2. Cawley, C. (2023, August 14). *How to build a simple PHP website*. MUO. https://www.makeuseof.com/tag/build-simple-php-website/

3. freeCodeCamp.org. (2022, November 4). *The best PHP examples*. https://www.freecodecamp.org/news/the-best-php-examples/

4. GeeksforGeeks. (2019, July 31). *Basics of phpmyadmin*. GeeksforGeeks. https://www.geeksforgeeks.org/basics-of-phpmyadmin/

5. GeeksforGeeks. (2020, June 16). *Use case diagram for library management system*. GeeksforGeeks. https://www.geeksforgeeks.org/use-case-diagram-for-library-management-system/

6. How do I import a database to phpmyadmin? – support | one.com. (n.d.). https://help.one.com/hc/en-us/articles/115005588189-How-do-I-import-a-database-to-phpMyAdmin-

7. *How to - build a website*. How To Build a Website. (n.d.). https://www.w3schools.com/howto/howto_website.asp

8. *How to build a website with HTML*. DigitalOcean. (n.d.). https://www.digitalocean.com/community/tutorial-series/how-to-build-a-website-with-html

9. *Library Management System*. Creately. (n.d.). https://creately.com/diagram/example/hszw2rmn7/library-management-system

10. raghwendra Me a freelance web developer from new delhi india. visit me View all of raghwendra's posts. (2018, December 29). *Home*. PHP development company. https://www.raghwendra.com/blog/how-to-connect-html-to-database-with-mysql-using-php-example/

7. Appendix

https://github.com/muhdhaikal3/SWC3503-FINAL-PROJECT.git