



VIDYASAGAR UNIVERSITY

“Library Management System”

Certificate Course in Agile Software Development (Under CCAE)

Session: 2024-2025

Submitted by

Sanjiban Dey (VU/CCAЕ/ASD-II-CC/2024/31)

Bikram Roy (VU/CCAЕ/ASD-II-CC/2024/32)

Surajit Maji (VU/CCAЕ/ASD-II-CC/2024/33)

Anamika Das (VU/CCAЕ/ASD-II-CC/2024/34)

Srishti Shil (VU/CCAЕ/ASD-II-CC/2024/35)

DEPARTMENT OF COMPUTER SCIENCE
Vidyasagar University, Midnapore-721102
West Bengal, India

Certificate

This is to certify that the project entitled “**Library Management System**” is being submitted for the **Certificate Course in Agile Software Development** at Vidyasagar University. It is a record of bonafide project work carried out under my guidance and supervision.

Dr. Uttam Mandal

Dr. Partha Chaudhuri

Mr. Debkumar Bera

Certificate Course in Agile Software Development

Vidyasagar University

ACKNOWLEDGEMENT

We would like to express our sincere gratitude to the Certificate Course in Agile Software Development (Under CCAE) ,Vidyasagar University for providing us with the opportunity to undertake and complete this Agile software development project.

We are deeply thankful to our project guide, **Prof.(Dr.) Uttam mondal, Prof. Partha Chaudhuri , Prof. Debkumar Bera**, for their invaluable guidance, constructive feedback, and constant support throughout the development of this project. Their mentorship played a key role in helping us understand Agile methodologies and implement them effectively.

We also extend our appreciation to all faculty members and peers who provided insights and encouragement during various phases of this project.

Finally, we are grateful for the collaborative environment, tools, and resources provided by the university, which were crucial to the successful completion of this Agile project.

Submitted by:

1. Sanjiban Dey

2. Bikram Roy

3. Surajit Maji

4. Anamika Das

5. Srishti Shil

Date: 30.06.2025

Table of Contents

Sl. No	Content		Page No
1.	Table of Contents		4
2.	Abstract		5
3.	Introduction		6-7
4.	Methodology		8-9
5.	Implementation		10
6.	7.1	Core Backend Functions	11-19
7.	7.2	Database Tables and Schema	20-23
8.			
9.	7.3	Entity Relationship Diagram (ERD)	24
	Result and Discussion		25-31
	Conclusion		33-34
10.	Appendix		35-43
11.	References		45

Abstract

In today's academic and institutional environments, traditional library systems face challenges such as manual errors, inefficient tracking of borrowed books, and lack of real-time data access. To overcome these limitations, this project presents a **web-based Library Management System** that streamlines administrative and student interactions with library resources.

The system is developed using the **Agile Software Development methodology**, ensuring flexibility and continuous improvement across sprints. Major modules—such as login/registration, book issue/return, and admin dashboard—were developed incrementally with continuous testing and feedback integration. Technologies used include **HTML5, CSS3, JavaScript** for the frontend, **PHP** for backend logic, and **MySQL** for database management.

The core objective of the system is to allow administrators to efficiently manage book inventories and student records, while also enabling students to borrow and re-borrow books with ease. Features like real-time date tracking, issue-return logic, and secure authentication enhance operational efficiency. The result is a scalable, modular, and user-friendly system that offers significant improvements over traditional manual systems.

This report outlines the system's design, development methodology, architecture, key features, and implementation challenges, culminating in a robust solution for library resource management.

Introduction

Libraries have long served as the backbone of academic institutions, offering essential resources for students, researchers, and faculty members. However, conventional methods of managing library operations—such as handwritten logs and paper-based tracking—are prone to human error, data loss, and inefficiency. With the increasing demand for automation and digitization, the need for a robust and user-friendly Library Management System (LMS) has become more pressing than ever.

A. Objective of the Project

The main objective of this project is to develop a **web-based Library Management System** that:

- I. Simplifies the process of **student registration and login**
- II. Allows **admins to manage books and issue records efficiently**
- III. Automates the **book issue, return, and re-borrow process** with accurate date tracking
- IV. Ensures **data consistency and security** through proper backend validation and database integration
- V. Provides a **responsive and intuitive user interface** for both students and administrators

B. Scope of the Project

This project is primarily designed for **educational institutions** and can be scaled for libraries in other domains. The system supports:

- I. Student and admin authentication
- II. Book management with metadata like title, author, and publisher
- III. Borrowing and return tracking with duration limits

- IV. Secure and scalable database design
- V. An admin dashboard to monitor and manage the overall system

Though currently built as a standalone system, the architecture allows for future integration with features such as barcode scanning, SMS/email reminders, and analytics dashboards. The current implementation is a solid foundation that can evolve into a fully automated library system for broader use.

Methodology

This project follows the **Agile Software Development** methodology, which emphasizes iterative development, continuous feedback, and modular implementation. The system was developed incrementally over several sprints, where each sprint focused on building, testing, and refining a specific functionality such as user login, book issue/return logic, and the admin dashboard.

Agile Approach

The Agile framework helped ensure flexibility and adaptability during the development process. Although this project was carried out by a small team (or individual), Agile roles were simulated to manage responsibilities effectively:

- **Developer:** Focused on coding the frontend, backend logic, and database queries.
- **Tester:** Carried out manual testing after each sprint to identify and fix bugs.
- **UI Designer:** Designed simple, intuitive interfaces for both student and admin users.

Regular internal reviews were held to decide the next sprint's priority, similar to Agile stand-ups, helping streamline task management and ensure progress.

Sprint-Based Development

Each functionality was implemented in a separate sprint:

- **Sprint 1:** Database schema design and login page
- **Sprint 2:** Admin dashboard and book entry module
- **Sprint 3:** Book issue, return, and re-borrow logic
- **Sprint 4:** UI integration, user testing, and bug fixing

Tools and Technologies Used

- **Frontend:** HTML5, CSS3, JavaScript
- **Backend:** PHP
- **Database:** MySQL
- **Editor/IDE:** Visual Studio Code
- **Local Server:** XAMPP / WAMP
- **Version Control:** Git (used optionally for tracking code updates)

Testing & Debugging

Manual testing was performed at each stage by simulating various user scenarios. This included:

- Invalid login attempts
- Book issue beyond limit
- Re-borrowing before return
- Admin access verification

Bugs identified during testing were fixed promptly, ensuring each module worked independently and in integration with others.

Implementation

This section provides technical insight into how the core functionalities of the Library Management System were implemented using PHP, HTML, and MySQL. The focus is on essential code logic, database structure, and system architecture.

Code Snippets

Login Functionality (PHP + HTML)

Login and registration were handled using separate form sections inside Login.html, with backend logic written in login.php (not shown here for brevity). Below is a **sample HTML login snippet** from Login.html

```
<form class="login" action="login.php" method="post" id="login">
    <input type="text" class="uname" name="uname"
placeholder="Username" required>
    <input type="password" class="pass" name="pass"
placeholder="Password" required>
    <button type="submit" class="submit" name="submit">Sign
in</button>
    <br>
</form>
```

Book Issue SQL Logic

Issued books are stored in the issue table. Here's an example **SQL query** used to insert a new issue record:

```
INSERT INTO issue (usin_id, isdn, date_of_borrow, borrow_book)
VALUES ('STUD001', 'BOOK123', CURRENT_DATE(), true);
```

Core Backend Functions (functions.php)

The file functions.php contains reusable backend logic that supports admin operations such as book management, issue status tracking, and database interactions. These functions are included and invoked in multiple PHP files across the project (e.g., dashboard.php, overdues.php).

Below is the complete code used in the functions.php file:

```
<?php
```

```
include("connection.php");
```

```
function all_books(){
```

```
    include("connection.php");
```

```
    $q="SELECT *from `book`";
```

```
    return mysqli_query($conn,$q);
```

```
}
```

```
function available_books($book=null){
```

```
    include("connection.php");
```

```
    $q="SELECT *from `book` JOIN `issue` on (`book`.`isbn`=`issue`.`isbn`)
```

```
where
```

```
    (`book`.`title` like '%$book%' or `book`.`title` like '%$book%' or `book`.`title`  
like '%$book%') and
```

```
    `issue`.`borrow_book`='0';";
```

```
    return mysqli_query($conn,$q); }
```

```
function number_of_copy(){  
    include("connection.php");  
    $q="SELECT *from `book`";  
    $l=mysqli_query($conn,$q);  
    return $l;  
}
```

```
function total_student($usin_id=null){  
    include("connection.php");  
    $q="SELECT *from `student`";  
    $l=mysqli_query($conn,$q);  
    return $l;  
}
```

```
function number_of_borrowed_book(){  
    include("connection.php");  
    $q="SELECT *from `issue` where `borrow_book`=1";  
    return mysqli_query($conn,$q);  
}
```

```

function overdue() {
    include("connection.php");

    // Get the current date and subtract 15 days
    $current_date = new DateTime();

    $overdate = $current_date->modify('-15 days')->format('Y-m-d'); // Format to
string for SQL

    // Prepare SQL query
    $q = "SELECT * FROM `issue` WHERE `borrow_book` = 1 AND
`date_of_borrow` < '$overdate'";

    // Run the query
    return mysqli_query($conn, $q);
}

```

```

function book_available($isdn){
    include("connection.php");

    // check availability
    $q="SELECT `borrow_book` from `issue` where `isdn`='$isdn' and
`borrow_book`=1";

    $l=mysqli_query($conn,$q);

    $val=mysqli_fetch_assoc($l);

    $flag=true;

```

```
if($val){  
    if($val['borrow_book']==1){  
        $flag=false;  
    }  
    else{  
        $flag=true;  
    }  
}  
return $flag;  
}  
  
function add_book_in_stack($isdn,$title,$author,$publisher,$photo,$tmp_photo){  
    include("connection.php");  
  
    $query="INSERT INTO `book` (`isdn`,`title`,`author`,`publisher`,`photo`)  
values  
    ('$isdn','$title','$author','$publisher','$photo')";  
    $link=mysqli_query($conn,$query);  
    if($link){  
        $folder="books/".$photo;  
        move_uploaded_file($tmp_photo,$folder);  
        return true;  
    }  
}
```

```
else{
    return false;
}
}

function borrow_book($isdn,$usin_id){
    include("connection.php");
    // check availability
    if(book_available($isdn)){
        $q="INSERT INTO `issue` (`isdn`,`usin_id`) values ('$isdn','$usin_id');";
        $insert=mysqli_query($conn,$q);
        if($insert){
            return true;
        }
    }
    return false;
}

function renew_book($isdn,$usin_id){
    include("connection.php");
    $q="SELECT * `issue` where `isdn`='$isdn' and `usin_id`='$usin_id' and `borrow_book`=1;";
    $l=mysqli_query($conn,$q);
```

```
$row=mysqli_num_rows($link);

if($row==0){

    return false;

    // break the function

}

// other wise

$crntd=current_timestamp();

$q2="UPDATE `issue` set `date_of_re_borrow`='$crntd', `re_borrow_book`=1
WHERE `isdn`='$isdn' and `usin_id`='$usin_id' ;";

$l2=mysqli_query($q2);

if($l2){

    return true;

}

}


function return_book($isdn,$usin_id){

    include("connection.php");

    $q="SELECT * `issue` where `isdn`='$isdn' and `usin_id`='$usin_id' and
`borrow_book`=1;";

    $l=mysqli_query($conn,$q);

    $row=mysqli_num_rows($link);

if($row==0){

    return false;
```



```
// break the function

}

// other wise

$crntd=current_timestamp();

$q2="UPDATE `issue` set `date_of_return`='$crntd', `borrow_book`=0,
`return_book`=1 WHERE `isdn`='$isdn' and `usin_id`='$usin_id' ";

$l2=mysqli_query($q2);

if($l2){

    return true;

}

}

function student_register($usin_id,$registration_no,$session,$dob){

    include("connection.php");

    $insert="INSERT into `student`
(`usin_id`,`registration_no`,`student_name`,`student_session`,`dob`) values
('$usin_id','$registration_no','$name','$session','$dob');"

    $c=mysqli_query($conn,$insert);

    if($c){

        return true;

    }

}
```

```
else{  
    return false;  
}  
}  
  
function admin_login($admin_id,$admin_pass){  
    include("connection.php");  
  
    $q="SELECT *from `admin` where `admin_id`='$admin_id' and  
`admin_password`='$admin_pass';";  
  
    $l=mysqli_query($conn,$q);  
  
    $row=mysqli_num_rows($l);  
  
    if($row==0){  
        return false;  
  
        // break the function for unvalid admin id and password  
    }  
    else{  
        session_start();  
  
        $_SESSION['admin_id']=$admin_id;  
  
        // login sucessfully  
  
        return true;  
    }  
}
```

```
function student_profile($usin_id){  
    include("connection.php");  
    $q="SELECT * from `student` where `usin_id`='$usin_id';";  
    $l=mysqli_query($conn,$q);  
    $profile['student']=mysqli_fetch_assoc($l);  
  
    $q2="SELECT * from `student` , `issue` where  
`student`.`usin_id`=`issue`.`usin_id` and `student`.`usin_id`='$usin_id' and  
    `issue`.`borrow_book`=1 ;";  
    $l2=mysqli_query($conn,$q2);  
    $profile['borrowed_books']=$l2;  
  
    $q3="SELECT * from `student` , `issue` where  
`student`.`usin_id`=`issue`.`usin_id` and `student`.`usin_id`='$usin_id' and  
    `issue`.`borrow_book`=0 ;";  
    $l3=mysqli_query($conn,$q3);  
    $profile['returned_books']=$l3;  
  
    return $profile;  
}  
?>
```

Database Tables and Schema

All SQL tables were created in library.sql. Below is a breakdown of the schema:

Data Base:

Table	Action	Rows	Type	Collation	Size	Overhead
<input type="checkbox"/> admin		2	InnoDB	utf8mb4_general_ci	16.0 KiB	-
<input type="checkbox"/> book		5	InnoDB	utf8mb4_general_ci	16.0 KiB	-
<input type="checkbox"/> issue		1	InnoDB	utf8mb4_general_ci	16.0 KiB	-
<input type="checkbox"/> student		4	InnoDB	utf8mb4_general_ci	16.0 KiB	-
4 tables	Sum	12	InnoDB	utf8mb4_general_ci	64.0 KiB	0 B

Fig. 1:- All tables list

Table: admin (Stores admin login credentials) :

```
CREATE TABLE admin (
    admin_id VARCHAR(30) PRIMARY KEY,
    admin_name VARCHAR(30),
    admin_password VARCHAR(30)
);
```

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
<input type="checkbox"/> 1	admin_id	varchar(30)	utf8mb4_general_ci		No	None			
<input type="checkbox"/> 2	admin_name	varchar(30)	utf8mb4_general_ci		Yes	NULL			
<input type="checkbox"/> 3	admin_password	varchar(30)	utf8mb4_general_ci		Yes	NULL			

Fig 2 : Admin table structure

Table: student (Holds student registration details) :

```
CREATE TABLE student (
    usin_id VARCHAR(30) PRIMARY KEY,
    registration_no INT(30),
    student_name VARCHAR(30),
    student_session VARCHAR(30),
    dob DATE
);
```

Student:

















#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
<input type="checkbox"/> 1	usin_id 	varchar(30)	utf8mb4_general_ci		No	None			 Change  Drop  More
<input type="checkbox"/> 2	registration_no	int(30)			Yes	NULL			 Change  Drop  More
<input type="checkbox"/> 3	student_name	varchar(30)	utf8mb4_general_ci		Yes	NULL			 Change  Drop  More
<input type="checkbox"/> 4	student_session	varchar(30)	utf8mb4_general_ci		Yes	NULL			 Change  Drop  More
<input type="checkbox"/> 5	dob	date			Yes	NULL			 Change  Drop  More

Fig 3 :- Student Table Structure

Table: book

Contains book metadata.

```
CREATE TABLE book (
    isdn VARCHAR(30) PRIMARY KEY,
    title VARCHAR(30),
```

```

author VARCHAR(30),
publisher VARCHAR(30),
photo VARCHAR(100) DEFAULT NULL
);

```

book:

















#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
<input type="checkbox"/> 1	isdn 	varchar(30)	utf8mb4_general_ci		No	None			 Change  Drop  More
<input type="checkbox"/> 2	title	varchar(30)	utf8mb4_general_ci		Yes	NULL			 Change  Drop  More
<input type="checkbox"/> 3	author	varchar(30)	utf8mb4_general_ci		Yes	NULL			 Change  Drop  More
<input type="checkbox"/> 4	publisher	varchar(30)	utf8mb4_general_ci		Yes	NULL			 Change  Drop  More
<input type="checkbox"/> 5	photo	varchar(100)	utf8mb4_general_ci		Yes	NULL			 Change  Drop  More

Fig 4:- Books Table Structure

Table: issue (Tracks borrow, re-borrow, and return status with timestamps):

```
CREATE TABLE issue (
```

```

    issue_id INT(30) AUTO_INCREMENT PRIMARY KEY,
    usin_id VARCHAR(30),
    isdn VARCHAR(30),
    date_of_borrow DATE DEFAULT CURRENT_TIMESTAMP(),
    borrow_book BOOLEAN DEFAULT true,
    borrowed_duration INT(30) DEFAULT 15,
    date_of_re_borrow DATE DEFAULT NULL,
    re_borrow_book BOOLEAN DEFAULT 0,
    date_of_return DATE DEFAULT NULL,
    return_book BOOLEAN DEFAULT 0

```

);

issue:



































#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
<input type="checkbox"/> 1	issue_id 	int(30)			No	None		AUTO_INCREMENT	 Change  Drop  More
<input type="checkbox"/> 2	usin_id	varchar(30)	utf8mb4_general_ci		Yes	NULL			 Change  Drop  More
<input type="checkbox"/> 3	isdn	varchar(30)	utf8mb4_general_ci		Yes	NULL			 Change  Drop  More
<input type="checkbox"/> 4	date_of_borrow	date			Yes	current_timestamp()			 Change  Drop  More
<input type="checkbox"/> 5	borrow_book	tinyint(1)			Yes	1			 Change  Drop  More
<input type="checkbox"/> 6	borrowed_duration	int(30)			Yes	15			 Change  Drop  More
<input type="checkbox"/> 7	date_of_re_borrow	date			Yes	NULL			 Change  Drop  More
<input type="checkbox"/> 8	re_borrow_book	tinyint(1)			Yes	0			 Change  Drop  More
<input type="checkbox"/> 9	date_of_return	date			Yes	NULL			 Change  Drop  More
<input type="checkbox"/> 10	return_book	tinyint(1)			Yes	0			 Change  Drop  More
<input type="checkbox"/> 11	pdf	varchar(30)	utf8mb4_general_ci		Yes	NULL			 Change  Drop  More

Fig 5:- Issue Table Structure

Entity Relationship Diagram (ERD)

This ERD shows how the four core tables (admin, student, book, issue) are connected:

- student and book both relate to issue through foreign keys.

admin is independent and used only for login validation

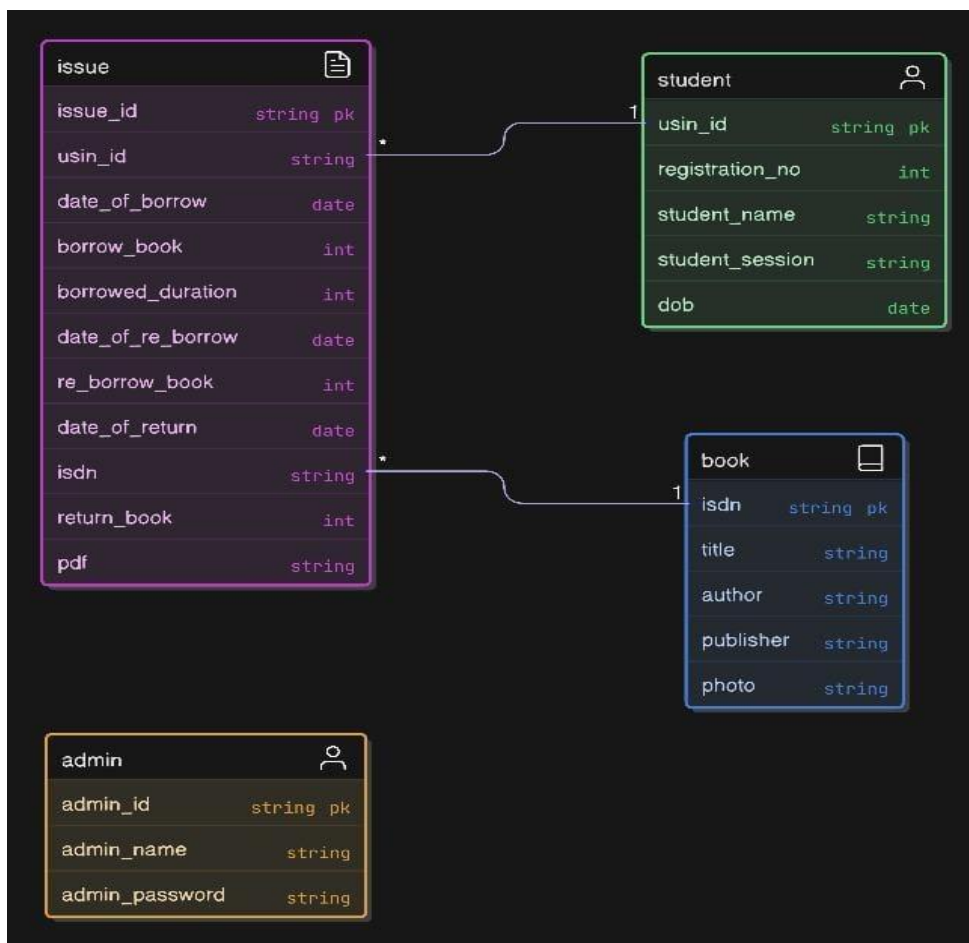


Fig 6 : ERD of Library Management System

Result and Discussion

This section illustrates the output interfaces of the Library Management System and discusses the effectiveness of the design and logic implementation based on manual testing and feedback.

Output Screenshots

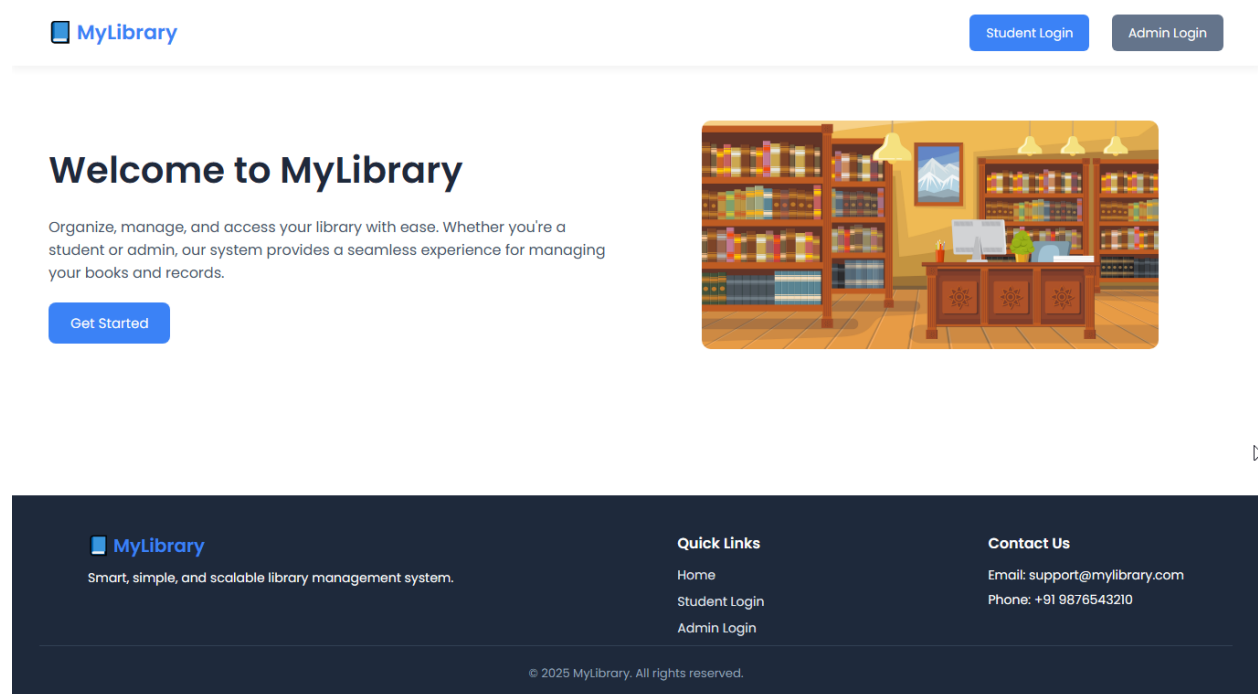
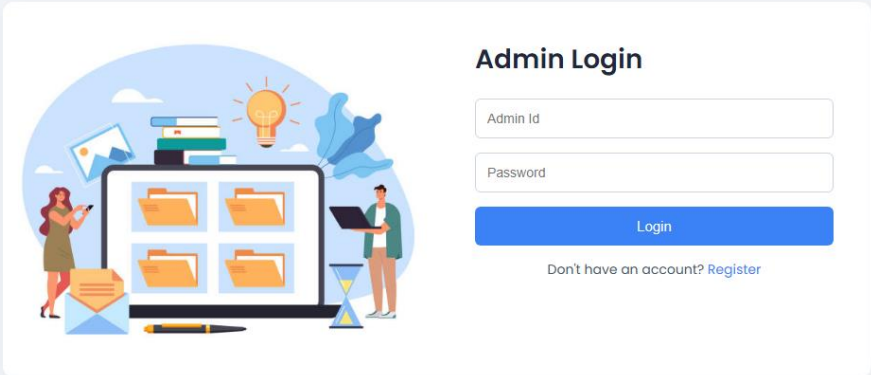


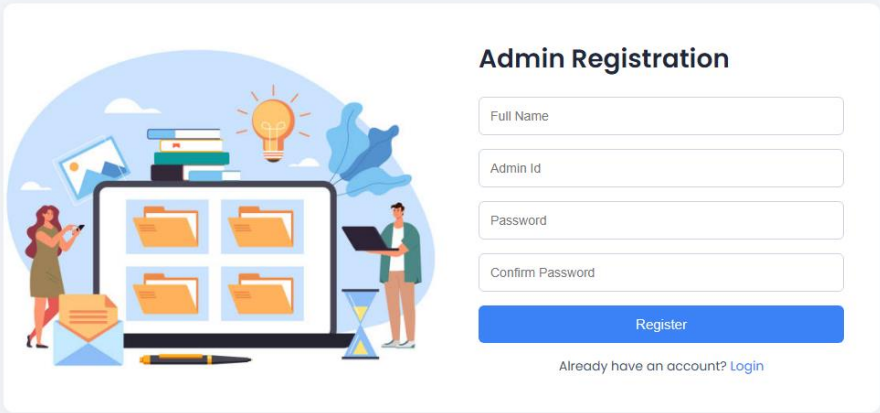
Fig 7: Home page (index.html)

The landing page offers clear navigation for both students and administrators. It is visually clean and adapts well to different screen sizes, using HTML5 and CSS3 for responsiveness.



The Admin Login page features a light blue background. On the left, there is a colorful illustration of a woman holding a smartphone, a man with a laptop, a large monitor displaying folders, a lightbulb, and a stack of books. On the right, the 'Admin Login' form is displayed. It includes two input fields: 'Admin Id' and 'Password'. Below these is a blue 'Login' button. At the bottom, there is a link that says 'Don't have an account? [Register](#)'.

Fig 8:- Admin Login Page



The Admin Registration page features a light blue background. On the left, there is a colorful illustration of a woman holding a smartphone, a man with a laptop, a large monitor displaying folders, a lightbulb, and a stack of books. On the right, the 'Admin Registration' form is displayed. It includes four input fields: 'Full Name', 'Admin Id', 'Password', and 'Confirm Password'. Below these is a blue 'Register' button. At the bottom, there is a link that says 'Already have an account? [Login](#)'.

Fig 8 :- Admin Registration Page

This page provides tabbed login/register functionality. JavaScript is used to toggle

the form views dynamically, while form submissions are securely handled via PHP on the backend.

This page allows users to either log in or register using toggled tabs. The JavaScript logic switches between login and register forms dynamically. Submitted data is passed to the backend (login.php or register.php) using POST. The system validates user credentials against the database before granting access. The form includes required field validation to prevent empty submissions.

Figure 6: Admin Dashboard (dashboard.php)

The Admin Dashboard is the central control panel for library administrators. Once the admin successfully logs in, they are redirected to this interface. From here, the admin can perform key tasks such as:

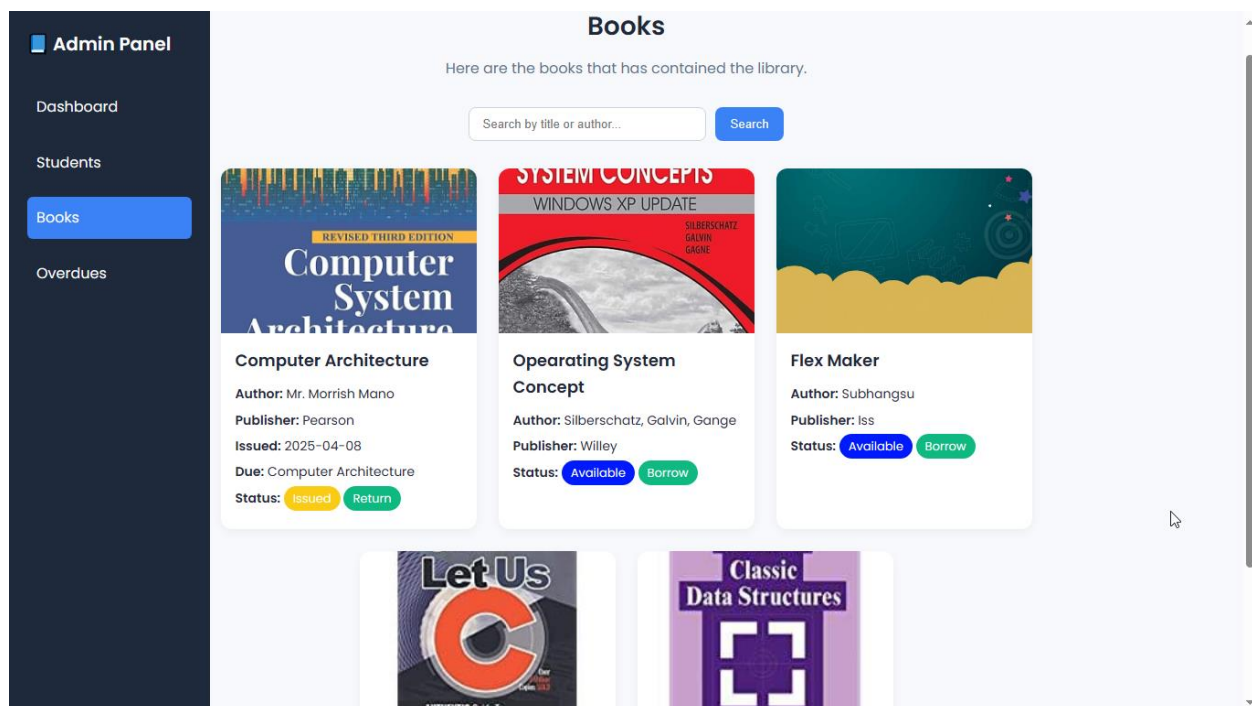


Fig 9 :- List of Books Page

Fig 6.1 : Viewing all books currently in the system

Admin Panel
Dashboard
Students
Books
Overdues

search by student name

Search

Name	Registration No	Session	Date of Birth
Bikram Roy	10025	2023-2025	0000-00-00
Koushik Mondal	10026	2023-2025	0000-00-00
Srishti Shil	10032	2023-2025	2012-09-02
Anamika Das	10006	2023-2025	2012-04-02

Fig 10 :- List of Books page

Fig 6.2 : Monitoring which books are issued and to whom

Admin Panel
Dashboard
Students
Books
Overdues

Overdue Books

Those borrowed books are over its return date.

Search by title or author...

Search

REVISED THIRD EDITION

Computer System Architecture

Computer Architecture

ISBN: 1000

Author: Mr. Morrish Mano

Issued By: Bikram Roy

USIN Id: 25

Fine to be Pay: Rs. 5.25 /-

Pay and return: Return

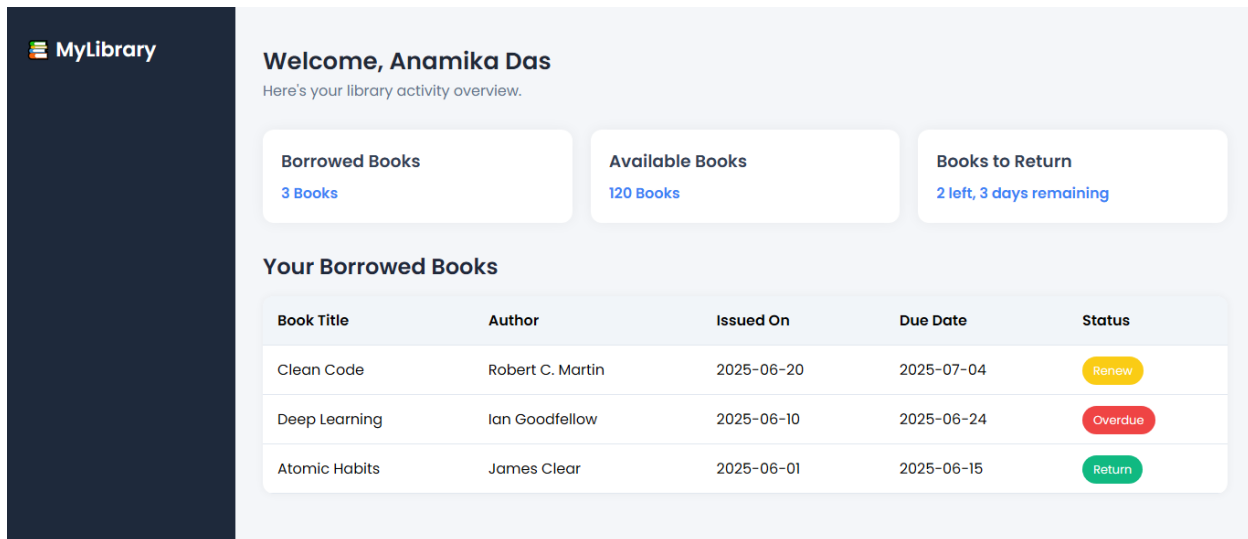
Fig. 11 :- List of Overdues

The screenshot displays the Admin Panel Dashboard Overview. On the left is a dark sidebar with the 'Admin Panel' header and a menu containing 'Dashboard' (highlighted), 'Students', 'Books', and 'Overdues'. The main content area is titled 'Dashboard Overview' and features four summary cards: 'Available Books' (5), 'Borrowed Books' (1), 'Total Students' (5), and 'Overdue Books' (1). Below these are two form sections. The 'Add New Book to Stock' form includes fields for 'ISDN Id', 'Book Title', 'Author', 'Publisher', and 'Book Cover Image' (with a file upload button). The 'Add New Student' form includes fields for 'Student USIN ID', 'Student Name', 'Registration', 'Session', and 'Date of birth' (with a date picker).

Fig 12 :- Dasboard Page (Managing reborrow and return actions)

The dashboard is dynamically connected to the MySQL database, fetching and displaying real-time data using PHP and SQL queries. Admins can use action buttons (like “Return” or “Re-issue”) for direct updates, which trigger backend logic in PHP to update the corresponding fields in the issue table (e.g., `return_book`, `date_of_return`, `re_borrow_book`).

The design ensures simplicity and efficiency, giving administrators quick access to all critical library records.



MyLibrary

Welcome, Anamika Das
Here's your library activity overview.

Borrowed Books
3 Books

Available Books
120 Books

Books to Return
2 left, 3 days remaining

Your Borrowed Books

Book Title	Author	Issued On	Due Date	Status
Clean Code	Robert C. Martin	2025-06-20	2025-07-04	Renew
Deep Learning	Ian Goodfellow	2025-06-10	2025-06-24	Overdue
Atomic Habits	James Clear	2025-06-01	2025-06-15	Return

Fig 13 :- Book Issue Record Output

This output section displays the list of books that have been issued by students, along with associated metadata. Each entry shows:

- Student ID (usin_id)
- Book ID (isdn)
- Date of issue
- Whether the book is still borrowed or returned
- Re-borrow and return dates (if applicable)

The data is fetched from the issue table and rendered using PHP. Conditional logic in the backend checks the borrow_book, return_book, and re_borrow_book fields

to determine the current status of each book. This output helps both the admin and system maintain proper tracking of book lifecycle events.

This section proves that the issue and return flow works as expected, and data is accurately reflected from the backend in the admin interface.

Discussion

What Worked Well

- **Responsive Design:** The interface adjusted well to different screen sizes and devices.
- **Secure Backend:** User inputs were validated both on the frontend and backend, reducing error risks.
- **Modular Code:** The use of functions.php allowed for clean, reusable logic in multiple modules.
- **Accurate Date Tracking:** The system correctly handled borrow and return date calculations.

Challenges Faced

- **Session Management:** In some cases, maintaining PHP session data between pages required extra debugging.
- **Re-borrow Logic:** Implementing re-borrow conditions without affecting original return records took multiple test iterations.
- **UI Bugs:** During testing, some layout misalignments occurred in older browsers, later fixed with CSS adjustments.

Conclusion

The primary objective of this project was to design and develop a **web-based Library Management System** that simplifies and streamlines the process of book lending, returning, and overall library record management. The system successfully addresses the common inefficiencies of manual library operations by enabling secure login for students and administrators, managing book records through an admin dashboard, and automating issue/re-issue/return logic with accurate date tracking.

Using **HTML5, CSS3, JavaScript, PHP, and MySQL**, the project was implemented following the **Agile Software Development** methodology, which allowed for incremental development, regular testing, and effective problem-solving. Each functional component was designed, tested, and integrated systematically to ensure smooth user experience and backend consistency.

Achievements

- I. Implemented full login and registration functionality for users.
- II. Created an admin dashboard for managing books and monitoring issue records.
- III. Designed a structured MySQL database with appropriate foreign keys and boolean flags for book status.
- IV. Ensured secure backend interaction and form validation.
- V. Developed a responsive and user-friendly interface.

Limitations

- i. No email/SMS integration for notifying users about due or overdue books.
- ii. No barcode scanning or QR code system for quick book search.
- iii. Limited to local server (XAMPP/WAMP); not deployed on live web hosting.

Future Enhancements

- i. Add automated overdue alerts via email or SMS.
- ii. Integrate barcode scanning for faster book entry and issue.
- iii. Create a separate student dashboard for tracking their own borrow history.
- iv. Deploy the system on a live server with proper authentication and SSL security.

Appendix

A: SQL Schema (library.sql)

```
drop table issue;
```

```
drop table book;
```

```
drop table student;
```

```
drop table admin;
```

```
create table admin(
```

```
    admin_id varchar(30) not null primary key,
```

```
    admin_name varchar(30),
```

```
    admin_password varchar(30)
```

```
);
```

```
create table student(
```

```
    usin_id varchar(30) not null primary key,
```

```
    registration_no int(30) ,
```

```
    student_name varchar(30),
```

```
    student_session varchar(30),
```

```
    dob date
```

```
);
```

```
create table book(  
    isdn varchar(30) not null primary key,  
    title varchar(30),  
    author varchar (30),  
    publisher varchar(30),  
    photo varchar(100) default null  
);
```

```
create table issue(  
    issue_id integer(30) not null AUTO_INCREMENT primary key,  
    usin_id varchar(30) references `student`(`usin_id`),  
    isdn varchar(30)references `book`(`isdn`),  
    date_of_borrow date DEFAULT current_timestamp(),  
    borrow_book boolean default true,  
    borrowed_duration integer(30)DEFAULT 15, -- set 15 day for initial duration  
    date_of_re_borrow date DEFAULT null,  
    re_borrow_book boolean default 0,  
    date_of_return date DEFAULT null,  
    return_book boolean default 0,  
);
```

B: Full Code (functions.php)

```
<?php
```

```
include("connection.php");
```

```
function all_books(){
```

```
    include("connection.php");
```

```
    $q="SELECT * from `book`";
```

```
    return mysqli_query($conn,$q);
```

```
}
```

```
function available_books($book=null){
```

```
    include("connection.php");
```

```
    $q="SELECT * from `book` JOIN `issue` on (`book`.`isbn`=`issue`.`isbn`)  
where
```

```
    (`book`.`title` like '%$book%' or `book`.`title` like '%$book%' or `book`.`title`  
like '%$book%') and
```

```
    `issue`.`borrow_book`='0';";
```

```
    return mysqli_query($conn,$q);
```

```
}
```

```
function number_of_copy(){
```

```
    include("connection.php");
```

```
$q="SELECT * from `book`";  
$l=mysqli_query($conn,$q);  
return $l;  
  
}  
  
function total_student($usin_id=null){  
    include("connection.php");  
    $q="SELECT * from `student`";  
    $l=mysqli_query($conn,$q);  
    return $l;  
  
}  
  
function number_of_borrowed_book(){  
    include("connection.php");  
    $q="SELECT * from `issue` where `borrow_book`=1";  
    return mysqli_query($conn,$q);  
}  
  
function overdue() {  
    include("connection.php");
```

```
// Get the current date and subtract 15 days

$current_date = new DateTime();

$soverdate = $current_date->modify('-15 days')->format('Y-m-d'); // Format to
string for SQL


// Prepare SQL query

$q = "SELECT * FROM `issue` WHERE `borrow_book` = 1 AND
`date_of_borrow` < '$soverdate'";


// Run the query

return mysqli_query($conn, $q);
}


function book_available($isdn){
    include("connection.php");

    // check availability

    $q="SELECT `borrow_book` from `issue` where `isdn`='$isdn' and
`borrow_book`=1";

    $l=mysqli_query($conn,$q);

    $val=mysqli_fetch_assoc($l);

    $flag=true;

    if($val){

        if($val['borrow_book']==1){

            $flag=false;
        }
    }
}
```

```
    }  
    else{  
        $flag=true;  
    }  
}  
return $flag;  
  
}  
  
function add_book_in_stack($isdn,$title,$author,$publisher,$photo,$tmp_photo){  
    include("connection.php");  
  
    $query="INSERT INTO `book` (`isdn`,`title`,`author`,`publisher`,`photo`)  
values  
    ('$isdn','$title','$author','$publisher','$photo')";  
    $link=mysqli_query($conn,$query);  
    if($link){  
        $folder="books/".$photo;  
        move_uploaded_file($tmp_photo,$folder);  
        return true;  
    }  
    else{  
        return false;  
    }  
}
```

```
}

}

function borrow_book($isdn,$usin_id){

    include("connection.php");

    // check availability

    if(book_available($isdn)){

        $q="INSERT INTO `issue` (`isdn`,`usin_id`) values ('$isdn','$usin_id');";

        $insert=mysqli_query($conn,$q);

        if($insert){

            return true;

        }

    }

    return false;

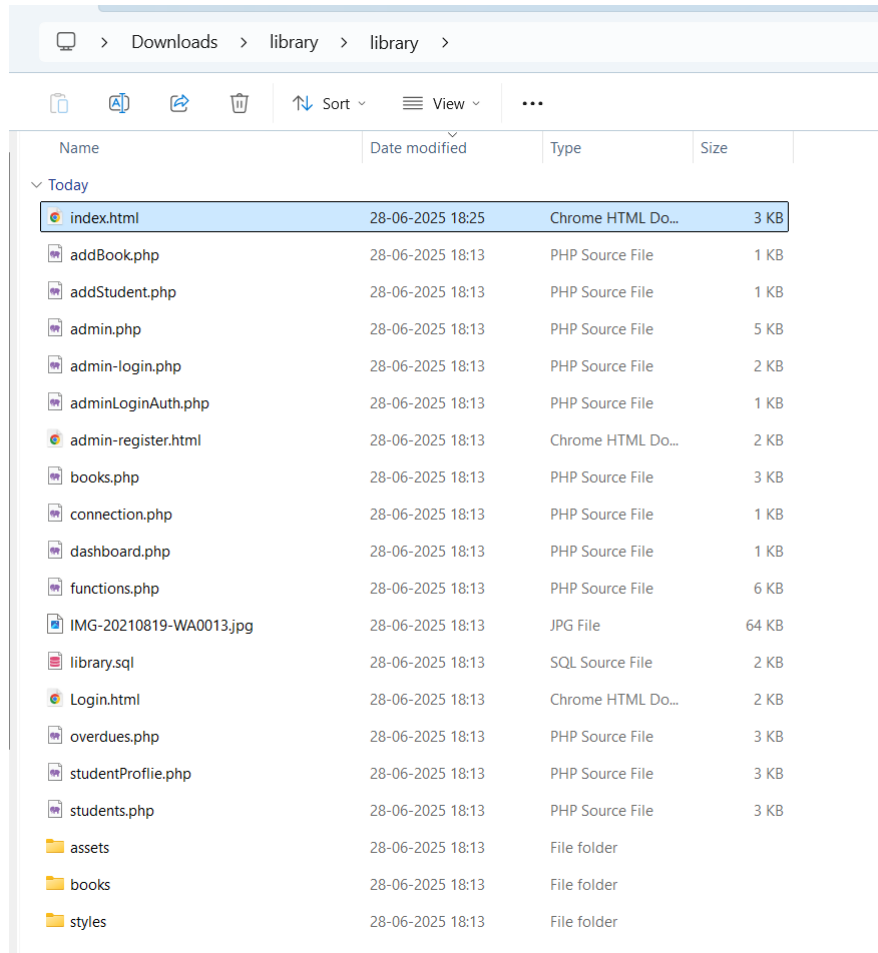
}
```

C: Other Code Files

The following code files are included

- index.html
- Login.html
- dashboard.php
- functions.php

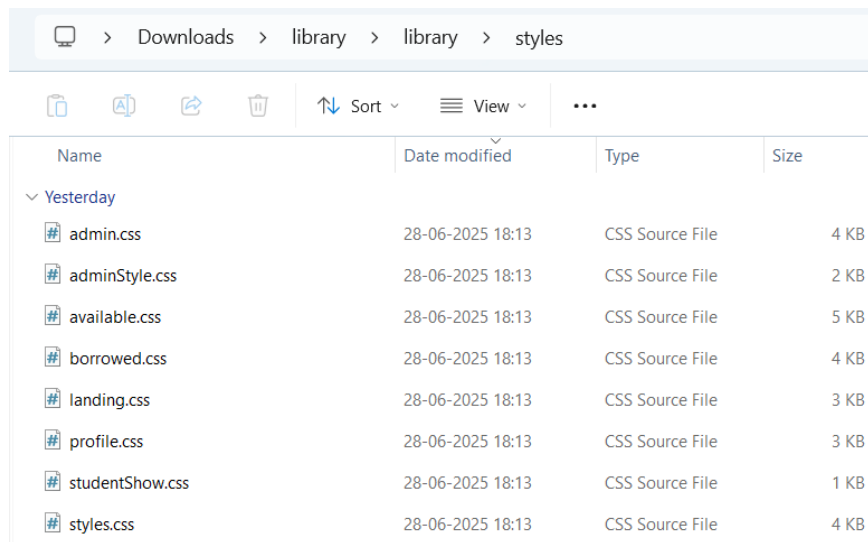
- overdues.php



Name	Date modified	Type	Size
▼ Today			
index.html	28-06-2025 18:25	Chrome HTML Do...	3 KB
addBook.php	28-06-2025 18:13	PHP Source File	1 KB
addStudent.php	28-06-2025 18:13	PHP Source File	1 KB
admin.php	28-06-2025 18:13	PHP Source File	5 KB
admin-login.php	28-06-2025 18:13	PHP Source File	2 KB
adminLoginAuth.php	28-06-2025 18:13	PHP Source File	1 KB
admin-register.html	28-06-2025 18:13	Chrome HTML Do...	2 KB
books.php	28-06-2025 18:13	PHP Source File	3 KB
connection.php	28-06-2025 18:13	PHP Source File	1 KB
dashboard.php	28-06-2025 18:13	PHP Source File	1 KB
functions.php	28-06-2025 18:13	PHP Source File	6 KB
IMG-20210819-WA0013.jpg	28-06-2025 18:13	JPG File	64 KB
library.sql	28-06-2025 18:13	SQL Source File	2 KB
Login.html	28-06-2025 18:13	Chrome HTML Do...	2 KB
overdues.php	28-06-2025 18:13	PHP Source File	3 KB
studentProfile.php	28-06-2025 18:13	PHP Source File	3 KB
students.php	28-06-2025 18:13	PHP Source File	3 KB
assets	28-06-2025 18:13	File folder	
books	28-06-2025 18:13	File folder	
styles	28-06-2025 18:13	File folder	

Fig 14 : All codes folder structure

- CSS folder: styles/



Name	Date modified	Type	Size
Yesterday			
admin.css	28-06-2025 18:13	CSS Source File	4 KB
adminStyle.css	28-06-2025 18:13	CSS Source File	2 KB
available.css	28-06-2025 18:13	CSS Source File	5 KB
borrowed.css	28-06-2025 18:13	CSS Source File	4 KB
landing.css	28-06-2025 18:13	CSS Source File	3 KB
profile.css	28-06-2025 18:13	CSS Source File	3 KB
studentShow.css	28-06-2025 18:13	CSS Source File	1 KB
styles.css	28-06-2025 18:13	CSS Source File	4 KB

Fig 15 : All styles sturcture

References

- <https://library.vidyasagar.ac.in/>
- <https://ndl.iitkgp.ac.in/>
- <https://openlibrary.org/>