Mirpur University Of Science And Technology

(M.U.S.T)

Submitted By : Ahad , Masood , Noorain

Submitted To: Sir Hafeez Ahmed

Reg No: 073 , 059 , 051

Session: 2023-27

Section: ODD

Course: Database Project Report

Department: Computer Systems Engineering

LIBRARY MANAGEMENT SYSTEM

Project Report

1.INTRODUCTION

Project Title: **Library Management System**

Objective of the Project

The objective of this project is to design and develop a **web-based Library Management System** that automates basic library operations. The system helps the administrator manage books, issue books to students, return books, and maintain accurate records using a database.

Traditional manual library systems are time-consuming and error-prone. This project replaces manual record-keeping with a computerized system that ensures **accuracy, efficiency, and ease of use**.

Scope of the Project

This system is designed for **small to medium libraries** such as:

- College libraries
- Departmental libraries
- Lab-based academic projects

The project focuses on **admin-side functionality only**, keeping the system simple and easy to understand.

System Development Tools and Platforms

Component	Tool / Platform
Frontend	HTML, CSS
Backend	PHP
Database	MySQL
Server	WAMP

SYSTEM OVERVIEW

System Description

The Library Management System allows an **admin user** to:

- Register and log in securely
- Add new books to the library
- View available books
- Issue books to students
- Return issued books
- Automatically update book quantities

All data is stored and managed using a **relational MySQL database**.

Users of the System

Admin

- Controls the entire system
- Manages books and issued records

NOTE: (No student login is included to keep the project simple.)

ENTITY RELATIONSHIP (ER) DIAGRAM

Entities Identified

1. **Users**
2. **Books**
3. **Issued_Books**

Entity Descriptions

Users

Stores information about administrators who can access the system.

Books

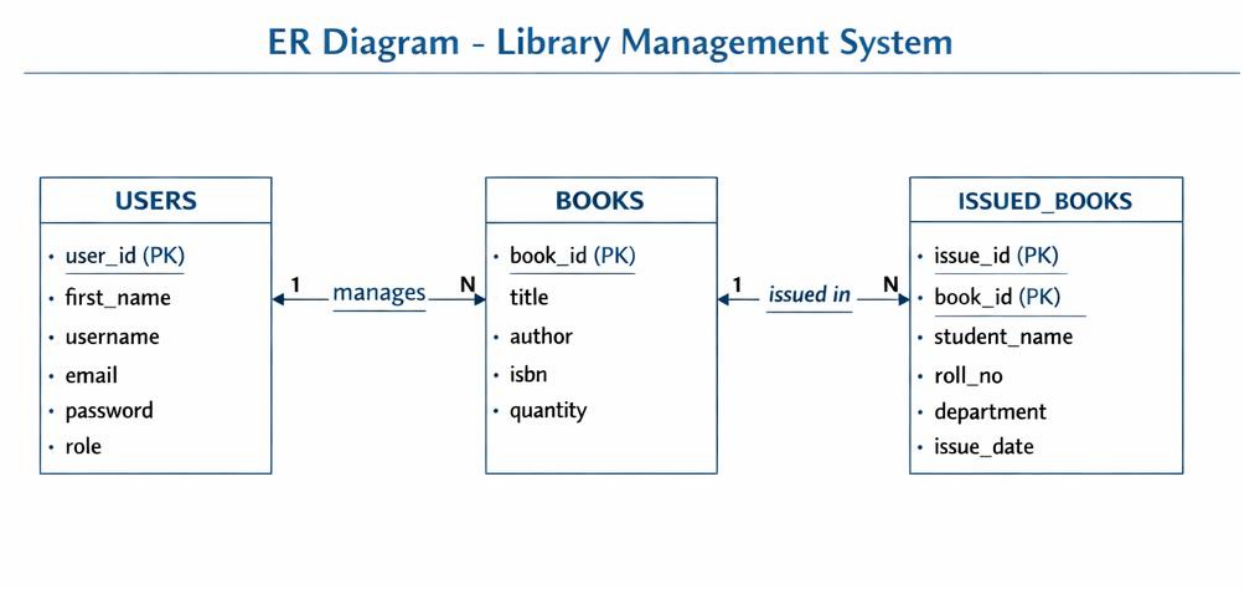
Stores information about all books available in the library.

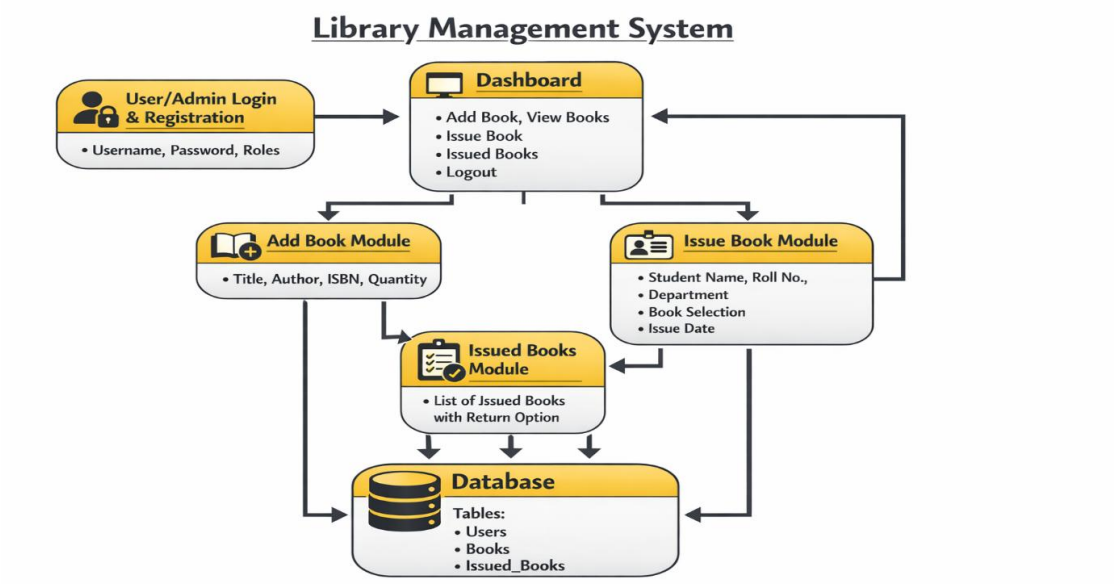
Issued_Books

Stores information about books that are issued to students.

Relationships

- One **admin** can issue **many books**
- One **book** can be issued **multiple times**
- **Issued_Books** acts as a bridge between **Books** and students' information





DATABASE DESIGN & SCHEMA

Database Name

library_db

Table Structure

USERS TABLE

```
CREATE TABLE users (
    user_id INT AUTO_INCREMENT PRIMARY KEY,
    first_name VARCHAR(50),
    middle_name VARCHAR(50),
    last_name VARCHAR(50),
    dob DATE,
    cnic VARCHAR(20),
    address TEXT,
    username VARCHAR(50),
    email VARCHAR(100),
    password VARCHAR(50),
    role VARCHAR(20)
);
```

Purpose is to stores admin login and personal information.

BOOKS TABLE

```
CREATE TABLE books (
    book_id INT AUTO_INCREMENT PRIMARY KEY,
    title VARCHAR(100),
    author VARCHAR(100),
    isbn VARCHAR(50),
    quantity INT
);
```

Purpose is to stores details of all books and their available quantity.

ISSUED_BOOKS TABLE

```
CREATE TABLE issued_books (
    issue_id INT AUTO_INCREMENT PRIMARY KEY,
    book_id INT,
    student_name VARCHAR(100),
    roll_no VARCHAR(50),
    department VARCHAR(50),
    issue_date DATE
);
```

Purpose is to keeps a record of books issued to students.

DATABASE DESIGN DECISIONS

Why Separate Tables Were Used?

- **Normalization:**
  Each table stores only relevant data to avoid redundancy.
- **Scalability:**
  Easy to add more features later.
- **Data Integrity:**
  Book records remain safe even when books are issued or returned.

Why Quantity Field in Books?

- Ensures that a book cannot be issued if it is unavailable.
- Automatically updates when books are issued or returned.

SOURCE CODE STRUCTURE

File Name	Description
db.php	Database connection
login.php	Admin login
register.php	Admin registration
dashboard.php	Main control panel
add_book.php	Add books
view_books.php	View books
issue_book.php	Issue books
issued_books.php	View & return books
logout.php	Logout
style.css	UI styling

1. db.php

Handles database connection using MySQL.
Used by all PHP files to interact with the database.

```
<?php
$conn = mysqli_connect("localhost", "root", "", "library_db");
if (!$conn) {
    die("Database connection failed");
}
?>
```

2. add_book.php

Allows admin to add new books to the library.
Stores book details like title, author, ISBN, and quantity.

```
<?php
```

```
include "db.php";
session_start();
if (isset($_POST['add'])) {
    $title = $_POST['title'];
    $author = $_POST['author'];
    $isbn = $_POST['isbn'];
    $quantity = $_POST['quantity']
  mysqli_query($conn,
        "INSERT INTO books (title, author, isbn, quantity)
         VALUES ('$title', '$author', '$isbn', '$quantity')");
    echo "<p style='color:green;'>Book Added Successfully</p>";
}
?>
<!DOCTYPE html>
<html>
<head>
    <title>Add Book</title>
    <link rel="stylesheet" href="style.css">
</head>
<body>
<h2>Add Book</h2>
<form method="POST">
    <input type="text" name="title" placeholder="Book Title" required><br><br>
    <input type="text" name="author" placeholder="Author" required><br><br>
    <input type="text" name="isbn" placeholder="ISBN" required><br><br>
    <input type="number" name="quantity" placeholder="Quantity" required><br><br>

    <!-- BUTTON ROW -->
    <div class="btn-row">
```

```
        <button type="submit" name="add">Add Book</button>
        <button type="button" onclick="window.location.href='dashboard.php'">
            Back to Dashboard
        </button>
    </div>
</form>
</body>
</html>
```

3. dashboard.php

Main control panel of the Library Management System.
Provides navigation to all major modules.

```
<?php
session_start();
if (!isset($_SESSION['user_id'])) {
    header("Location: login.php");
    exit();
}
?>
<!DOCTYPE html>
<html>
<head>
    <title>Dashboard</title>
    <link rel="stylesheet" href="style.css">
</head>
<body>

<h2><b>Library Management System</b></h2>
<br>
```

```
<ul>
    <li><a href="add_book.php"><b>Add Book</b></a></li>
    <li><a href="view_books.php"><b>View Books</b></a></li>
    <li><a href="issue_book.php"><b> Issue Book</b></a></li>
    <li><a href="issued_books.php"><b>Issued Books</b></a></li>
    <li><a href="logout.php"><b>Logout</b></a></li>
</ul>
</body>
</html>
```

4. issue_book.php

Issues a book to a student after availability check.
Decreases book quantity and records issue details.

```
<?php
include "db.php";
session_start();
if (!isset($_SESSION['user_id'])) {
    header("Location: login.php");
    exit();
}
$msg = "";
// ISSUE BOOK LOGIC
if (isset($_POST['issue'])) {
    $student_name = $_POST['student_name'];
    $roll_no      = $_POST['roll_no'];
    $department   = $_POST['department'];
    $book_id      = $_POST['book_id'];
    $issue_date   = $_POST['issue_date'];
    // Check book quantity
```

```
    $q = mysqli_query($conn, "SELECT quantity FROM books WHERE book_id =
$book_id");
    $row = mysqli_fetch_assoc($q);
    if ($row['quantity'] > 0) {
        // Insert into issued_books
        mysqli_query($conn,
            "INSERT INTO issued_books
            (book_id, student_name, roll_no, department, issue_date)
            VALUES
            ($book_id, '$student_name', '$roll_no', '$department', '$issue_date')"
        );
        // Decrease book quantity
        mysqli_query($conn,
            "UPDATE books SET quantity = quantity - 1 WHERE book_id = $book_id"
        );
        $msg = "Book Issued Successfully";
    } else {
        $msg = "Book Not Available";
    }
}
// Fetch books for dropdown
$books = mysqli_query($conn, "SELECT * FROM books WHERE quantity > 0");
?>
<!DOCTYPE html>
<html>
<head>
    <title>Issue Book</title>
    <link rel="stylesheet" href="style.css">
</head>
```

```
<body>
<h2>Issue Book</h2>
<?php
if ($msg != "") {
    echo "<p><b>$msg</b></p>";
}
?>
<form method="POST">
    <input type="text" name="student_name" placeholder="Student Name" required>
    <br><br>
    <input type="text" name="roll_no" placeholder="Roll No" required>
    <br><br>
    <select name="department" required>
        <option value="">Select Department</option>
        <option value="Computer Science">Computer Science</option>
        <option value="IT">IT</option>
        <option value="BBA">BBA</option>
        <option value="Electrical">Electrical</option>
    </select>
    <br><br>
    <select name="book_id" required>
        <option value="">Select Book</option>
        <?php while ($b = mysqli_fetch_assoc($books)) { ?>
            <option value="<?php echo $b['book_id']; ?>">
                <?php echo $b['title']; ?>
            </option>
        <?php } ?>
    </select>
    <br><br>
```

```
    <input type="date" name="issue_date" required>
    <br><br>
    <div class="btn-row">
        <button type="submit" name="issue">Issue Book</button>
        <button type="button" onclick="window.location.href='dashboard.php'">
            Back to Dashboard
        </button>
    </div>
</form>
</body>
</html>
```

5. issued_books.php

Shows list of all issued books.
Allows admin to return books and update quantity.

```
<?php
include "db.php";
session_start();
if (!isset($_SESSION['user_id'])) {
    header("Location: login.php");
    exit();
}
/* ===== RETURN BOOK LOGIC ===== */
if (isset($_GET['return_id'])) {
    $issue_id = $_GET['return_id'];
    // get book_id from issued_books
    $q = mysqli_query($conn,
        "SELECT book_id FROM issued_books WHERE issue_id = $issue_id");
    $row = mysqli_fetch_assoc($q);
```

```
    $book_id = $row['book_id'];
    // delete issued record
    mysqli_query($conn,
        "DELETE FROM issued_books WHERE issue_id = $issue_id");
    // increase book quantity
    mysqli_query($conn,
        "UPDATE books SET quantity = quantity + 1 WHERE book_id = $book_id");
    header("Location: issued_books.php");
    exit();
}
/* ===== FETCH ISSUED BOOKS ===== */
$result = mysqli_query($conn,
    "SELECT issued_books.*, books.title
     FROM issued_books
     JOIN books ON issued_books.book_id = books.book_id"
);
?>
<!DOCTYPE html>
<html>
<head>
    <title>Issued Books</title>
    <link rel="stylesheet" href="style.css">
</head>
<body>
<h2>Issued Books List</h2>
<table border="1" cellpadding="10">
    <tr>
        <th>Book Name</th>
        <th>Student Name</th>
```

```
        <th>Roll No</th>
        <th>Department</th>
        <th>Issue Date</th>
        <th>Action</th>
    </tr>
    <?php while ($row = mysqli_fetch_assoc($result)) { ?>
    <tr>
        <td><?php echo $row['title']; ?></td>
        <td><?php echo $row['student_name']; ?></td>
        <td><?php echo $row['roll_no']; ?></td>
        <td><?php echo $row['department']; ?></td>
        <td><?php echo $row['issue_date']; ?></td>
        <td>
            <a href="issued_books.php?return_id=<?php echo $row['issue_id']; ?>"
              onclick="return confirm('Return this book?');">
              Return
            </a>
        </td>
    </tr>
    <?php } ?>
</table>
<div class="back-btn">
    <button type="button" onclick="window.location.href='dashboard.php'">
        Back to Dashboard
    </button>
</div>
</body>
</html>
```

6. login.php

Authenticates admin user using username and password.
Starts session and redirects user to dashboard on success.

```
<?php
include "db.php";
session_start();
if (isset($_POST['login'])) {
    $username = $_POST['username'];
    $password = $_POST['password'];
    // Simple admin login query (basic)
    $query = "SELECT * FROM users
            WHERE username='$username'
            AND password='$password'
            AND role='admin'";
    $result = mysqli_query($conn, $query);
    if (mysqli_num_rows($result) == 1) {
        $row = mysqli_fetch_assoc($result);
        // Session create
        $_SESSION['user_id'] = $row['user_id'];
        // Redirect to dashboard
        header("Location: dashboard.php");
        exit();
    } else {
        echo "<p style='color:red;'>Invalid Username or Password</p>";
    }
}
?>
<!DOCTYPE html>
<html>
```

```
<head>
    <title>Admin Login Portal</title>
    <link rel="stylesheet" href="style.css">
</head>
<body>
<h2>Admin Login Portal</h2>
<br>
<form method="POST">
    <input type="text" name="username" placeholder="Username" required>
    <br><br>
    <input type="password" name="password" placeholder="Password" required>
    <br><br>
    <div class="btn-row">
        <button type="submit" name="login">Login</button>
        <button type="button" onclick="window.location.href='register.php'">
            Register New Admin
        </button>
    </div>
</form>
</body>
</html>
```

7. logout.php

Destroys the active user session securely.
Logs the admin out and redirects to the login page.

```
<?php
session_start();
session_destroy();
header("Location: login.php");
```

```
exit();
?>
```

8. register.php

Registers a new admin user in the system.
Validates input and stores admin data in database.

```
<?php
include "db.php";
if (isset($_POST['register'])) {
    $first_name = $_POST['first_name'];
    $middle_name = $_POST['middle_name'];
    $last_name = $_POST['last_name'];
    $dob = $_POST['dob'];
    $cnic = $_POST['cnic'];
    $address = $_POST['address'];
    $username = $_POST['username'];
    $email = $_POST['email'];
    $password = $_POST['password'];
    $confirm_password = $_POST['confirm_password'];
    // 1️⃣ Confirm password check
    if ($password != $confirm_password) {
        echo "<p style='color:red;'>Passwords do not match</p>";
    } else {
        // 2️⃣ Email already exists check
        $check_email = mysqli_query($conn,
            "SELECT * FROM users WHERE email='$email'");
        if (mysqli_num_rows($check_email) > 0) {
            echo "<p style='color:red;'>Email already exists</p>";
        } else {
```

```
        // 3. Insert admin record
        $query = "INSERT INTO users
        (first_name, middle_name, last_name, dob, cnic, address, username, email,
password, role)
        VALUES
        ('$first_name', '$middle_name', '$last_name', '$dob', '$cnic', '$address',
'$username', '$email', '$password', 'admin')";

        if (mysqli_query($conn, $query)) {
            echo "<p style='color:green;'>Admin Registered Successfully</p>";
        } else {
            echo "<p style='color:red;'>Registration Failed</p>";
        }     }    }
}
?>
<!DOCTYPE html>
<html>
<head>
   <title>Admin Registration</title>
   <link rel="stylesheet" href="style.css">
</head>
<body>
<h2>Admin Registration Form</h2>
<form method="POST">
   <input type="text" name="first_name" placeholder="First Name"
required><br><br>
   <input type="text" name="middle_name" placeholder="Middle Name"><br><br>
   <input type="text" name="last_name" placeholder="Last Name"
required><br><br>
   <input type="date" name="dob" required><br><br>
```

```
    <input type="text" name="cnic" placeholder="CNIC" required><br><br>
    <textarea name="address" placeholder="Address" required></textarea><br><br>
    <input type="email" name="email" placeholder="Email" required><br><br>
    <input type="text" name="username" placeholder="Username" required><br><br>
    <input type="password" name="password" placeholder="Password" required><br><br>
    <input type="password" name="confirm_password" placeholder="Confirm Password" required><br><br>
    <div class="btn-row">
        <button type="submit" name="register">Register Admin</button>
        <button type="button" onclick="window.location.href='login.php'">
            Back to Login
        </button>
    </div>
</form>
</body>
</html>
```

9. style.css

Provides styling for all pages of the project.
Ensures consistent layout, buttons, forms, and tables.

```
body {
    font-family: Arial, Helvetica, sans-serif;
    background-color: #f2f2f2;
    margin: 0;
    padding: 0;
}
h2 {
    text-align: center;
    color: #333;
```

```
    margin-top: 20px;
}
p {
    text-align: center;
    font-size: 14px;
}
/* ====================
   FORMS (Login, Register, Add, Issue)
==================== */
form {
    width: 320px;
    margin: 25px auto;
    background-color: #ffffff;
    padding: 20px;
    border-radius: 8px;
    border: 1px solid #ccc;
}
input,
select,
textarea {
    width: 95%;
    padding: 8px;
    margin-bottom: 12px;
    border-radius: 6px;
    border: 1px solid #aaa;
    font-size: 14px;
}
textarea {
    resize: none;
```

```
}
/* ====================
   BUTTONS
==================== */
button {
    padding: 8px 14px;
    background-color: #007BFF;
    color: #ffffff;
    border: none;
    border-radius: 6px;
    cursor: pointer;
    font-size: 14px;
}
button:hover {
    background-color: #0056b3;
}
/* Button row (Add + Back etc.) */
.btn-row {
    text-align: center;
}
.btn-row button {
    margin: 0 10px;
}
/* Back button container */
.back-btn {
    text-align: center;
    margin-top: 15px;
}
```

```
/* ====================
   DASHBOARD MENU
==================== */
ul {
    list-style-type: none;
    padding: 0;
    width: 320px;
    margin: 30px auto;      }
ul li {
    margin: 10px 0;
}
ul li a {
    display: block;
    padding: 12px;
    background-color: #007BFF;
    color: #ffffff;
    text-decoration: none;
    text-align: center;
    border-radius: 6px;
    font-weight: bold;
}
ul li a:hover {
    background-color: #0056b3;
}
/* ====================
   TABLES (View Books, Issued Books)
==================== */
table {
    width: 85%;
```

```
    margin: 20px auto;
    border-collapse: collapse;
    background-color: #ffffff;    }
th,
td {
    border: 1px solid #999;
    padding: 10px;
    text-align: center;
    font-size: 14px;
}
th {
    background-color: #ddd;
    font-weight: bold;
}
/* ======================
   ACTION LINKS (Return)
====================== */
table a {
    color: #007BFF;
    text-decoration: none;
    font-weight: bold;
}
table a:hover {
    text-decoration: underline;
}
/* ======================
   SUCCESS & ERROR TEXT
====================== */
.success {
```

```
    color: green;
    text-align: center;
    font-weight: bold;
}
.error {
    color: red;
    text-align: center;
    font-weight: bold;
}
```

6. SQL QUERIES USED

Admin Login Query

```
SELECT * FROM users
WHERE username='$username'
AND password='$password'
AND role='admin';
```

Purpose is to validates admin credentials and allows access to the dashboard.

Insert New Book

```
INSERT INTO books (title, author, isbn, quantity)
VALUES ('$title', '$author', '$isbn', '$quantity');
```

Purpose is to adds a new book to the library database.

View All Books

```
SELECT * FROM books;
```

Purpose is to displays all books available in the library.

Issue Book

INSERT INTO issued_books (book_id, student_name, roll_no, department, issue_date) VALUES ('$book_id', '$student_name', '$roll_no', '$department', '$issue_date');

Purpose stores issued book details.

Update Book Quantity (Issue)

UPDATE books SET quantity = quantity - 1 WHERE book_id = $book_id;

Return Book

DELETE FROM issued_books WHERE issue_id = $issue_id;

Update Quantity (Return)

UPDATE books SET quantity = quantity + 1 WHERE book_id = $book_id;

9. SECURITY FEATURES

- Session handling for login
- Admin-only access
- Validation checks before issuing books

10. LIMITATIONS

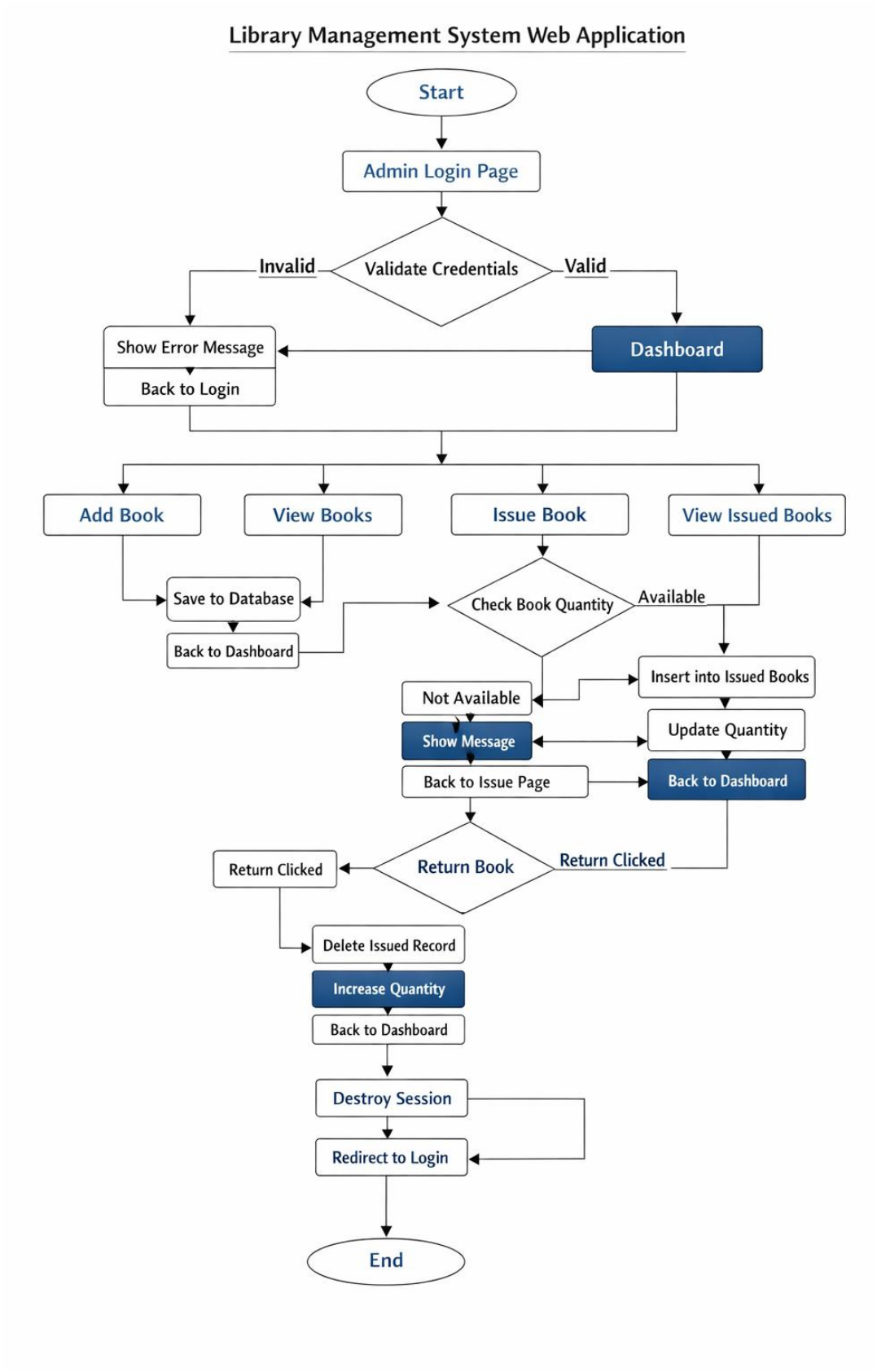
- No student login system
- Passwords are not encrypted
- No search or filtering

11. FUTURE ENHANCEMENTS

- Student login portal
- Password hashing
- Book search functionality
- Fine calculation

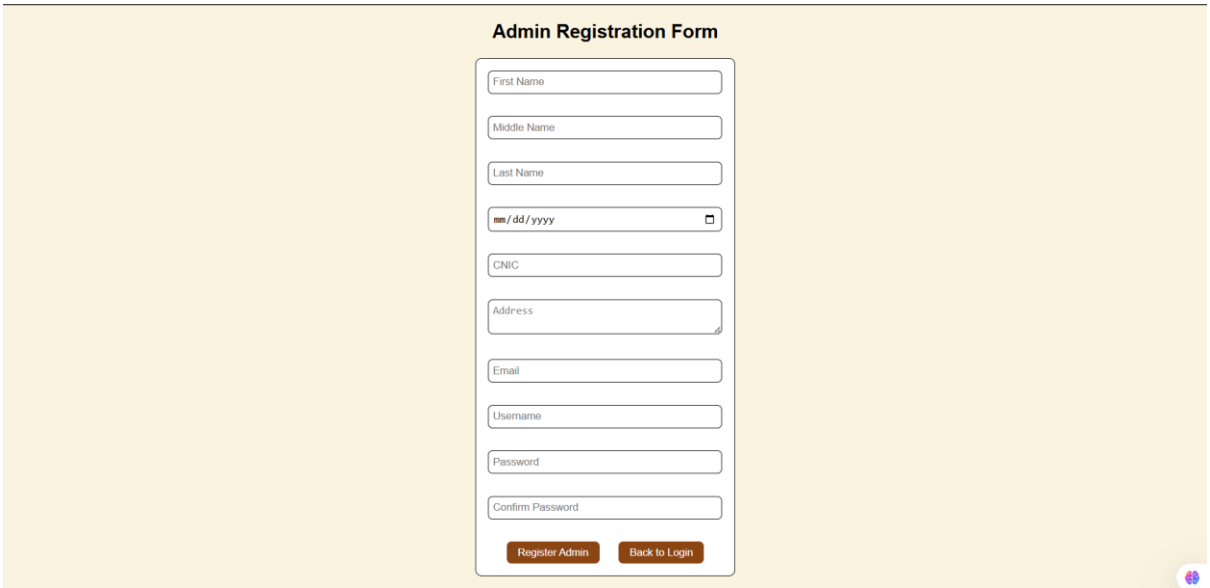
  Report generation

APPLICATION FLOW CHART & SCREENSHOTS



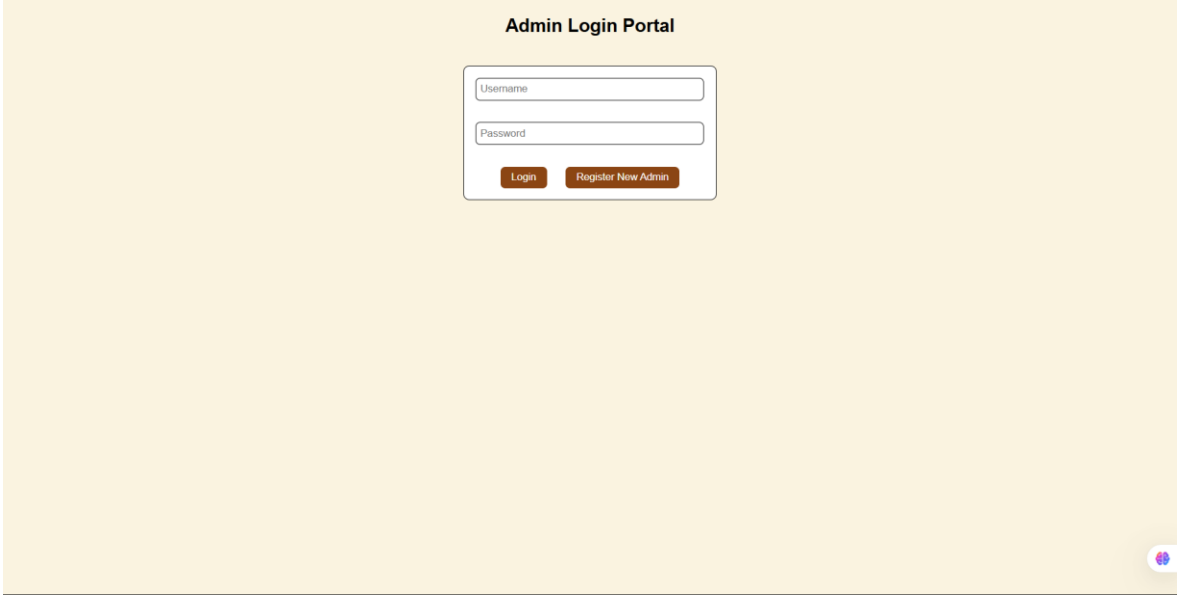
Admin Registration Page:

This page is for the new registration of the admins. They have to enter their details to register in the system



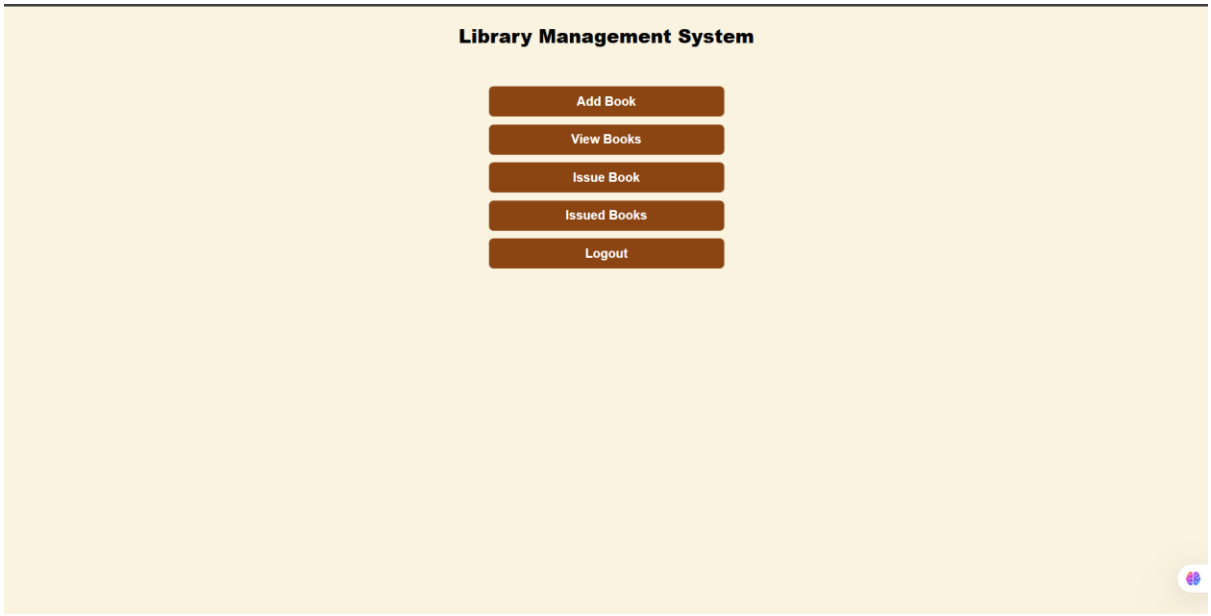
Login Page

The Login Page is used for user authentication. The user enters login credentials, which are validated through the database. If the credentials are correct, the user is redirected to the Dashboard Page. In case of invalid credentials, an error message is displayed.



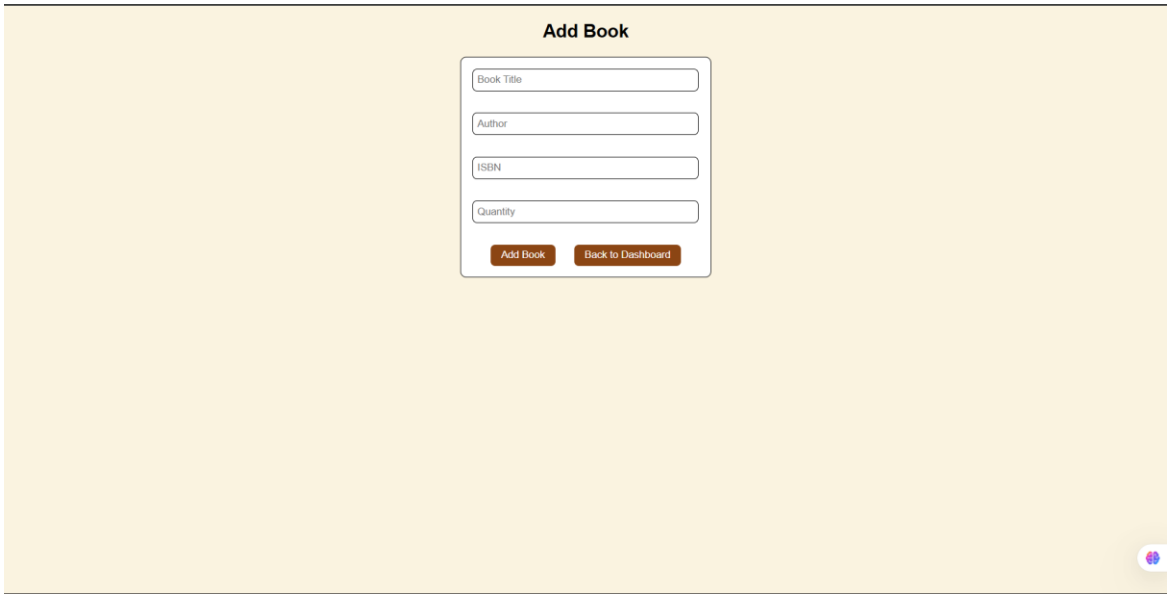
Dashboard Page

The Dashboard Page acts as the main interface of the system after successful login. It provides access to different modules and functional pages such as data entry, viewing records, and system management. From here, the user can navigate to other pages easily.



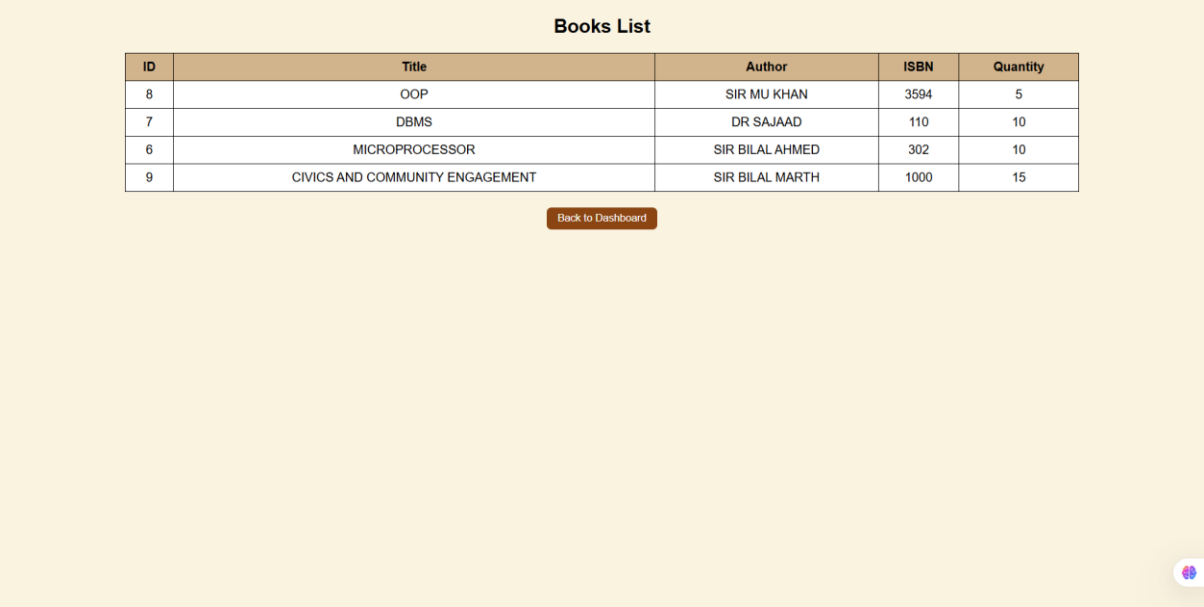
Add Book Page

This page allows the admin to add new books to the library.Book details such as title, author, ISBN, and quantity are stored in the database.The system confirms successful insertion after submission



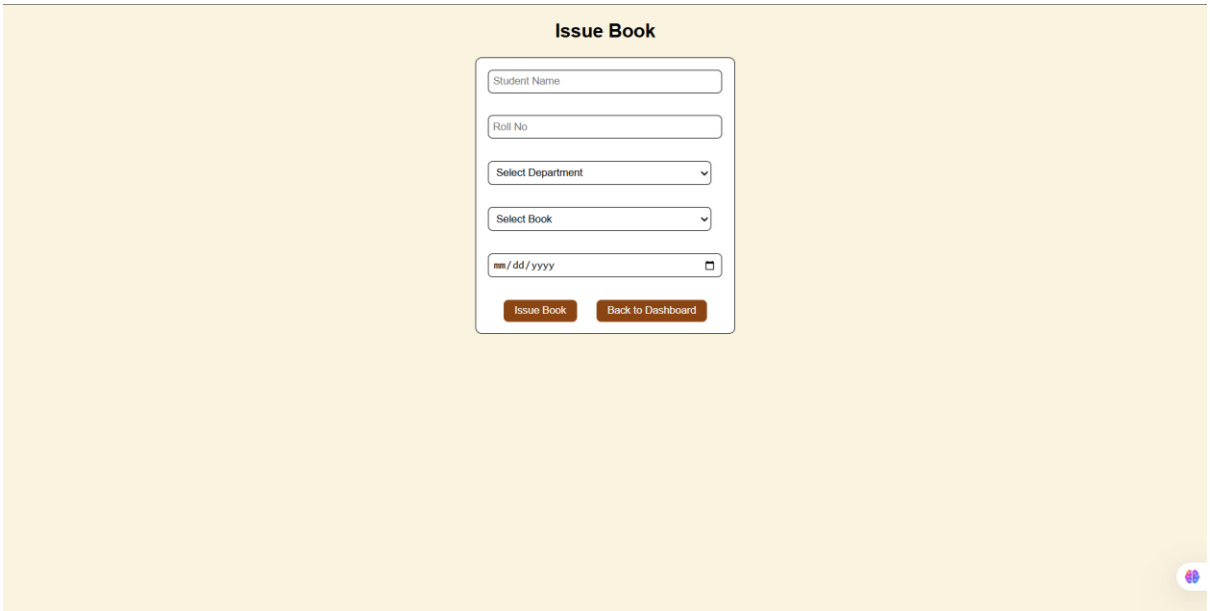
View Books Page

This page displays all available books in tabular format. It retrieves book records directly from the database.The admin can view book details including quantity and ISBN



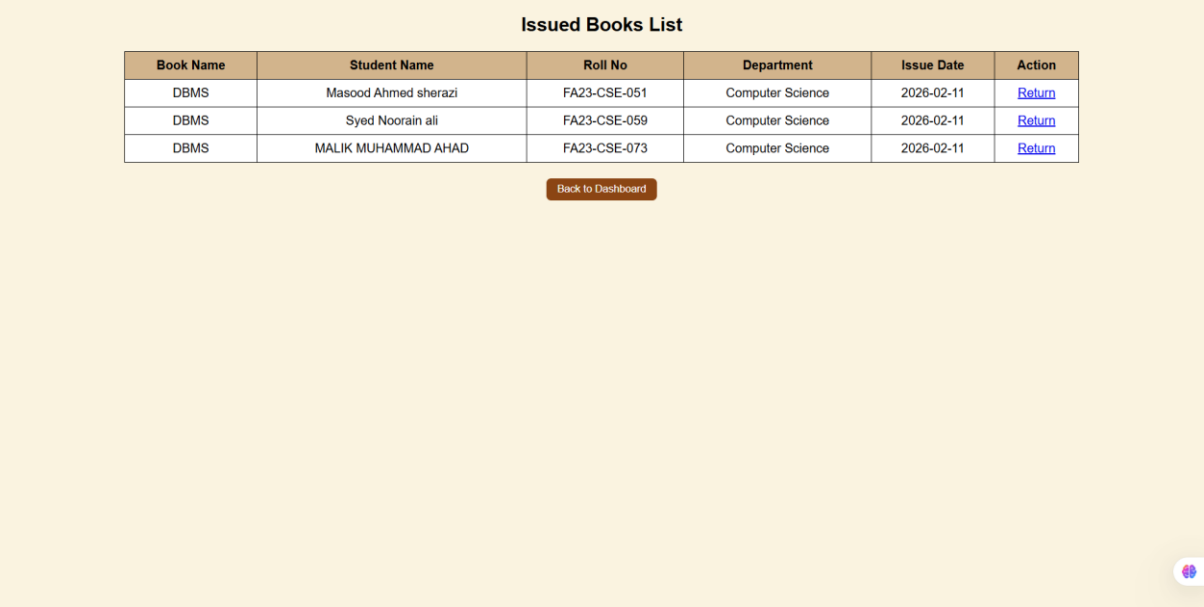
Issue Book Page

This page allows the admin to issue a book to a student.The system checks book availability before issuing.After issuing, the book quantity is automatically reduced.



Issued Books Page

This page shows all books currently issued to students.It displays student details along with issue date.The admin can return a book, which updates the quantity in the database



←THE END→