

University of Tabuk

College of Computers & Information Technology

Department of Computer Science

Advanced Web Technology (CIT1303)

University Library Management System

Final PHP Project

Group Members

[Member1 Name]

[Member 2 Name]

[Member 3 Name]

[Member 4 Name]

[Member 5 Name]

Project Description

We created a web application called the "University Library Management System". This system handles the core operations of a library, including inventory management, user authentication, and book lending. It allows an Admin to manage the entire book catalog, track current loans, and manage all user accounts (students and staff). Students can log in to view their active loans and check due dates. Non-logged-in Guests can search and view the public catalog for book availability. The application is built using a secure PHP backend and a responsive front-end design.

Database Structure

We used MySQL to store all system data. There are three essential tables that link together using Foreign Keys to maintain data integrity:

- **users Table:** Stores student and admin information, including ID, username, password (hashed), and user_type.
- **books Table:** Holds the library inventory, including book titles, authors, ISBN, total_copies, and available_copies.
- **loans Table:** Tracks all lending transactions, linking a specific user_id to a book_id, along with loan_date, due_date, and return_date.

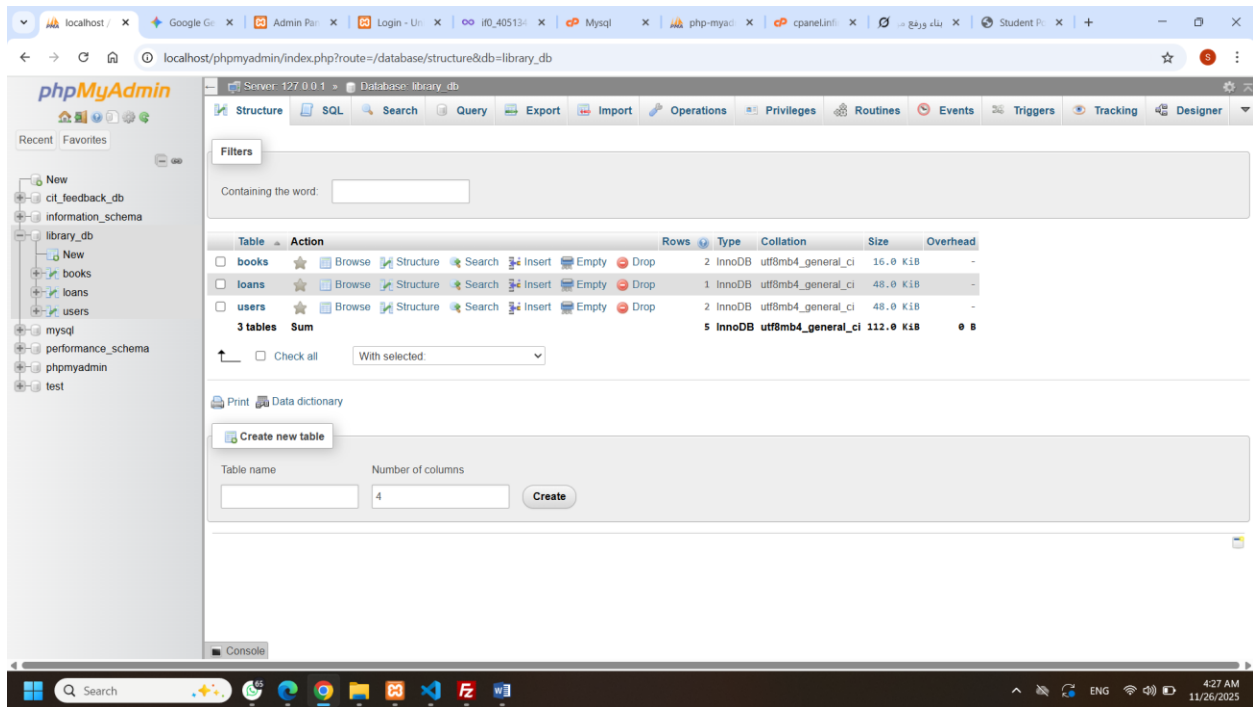


Figure 1: Database structure showing all tables and relationships.

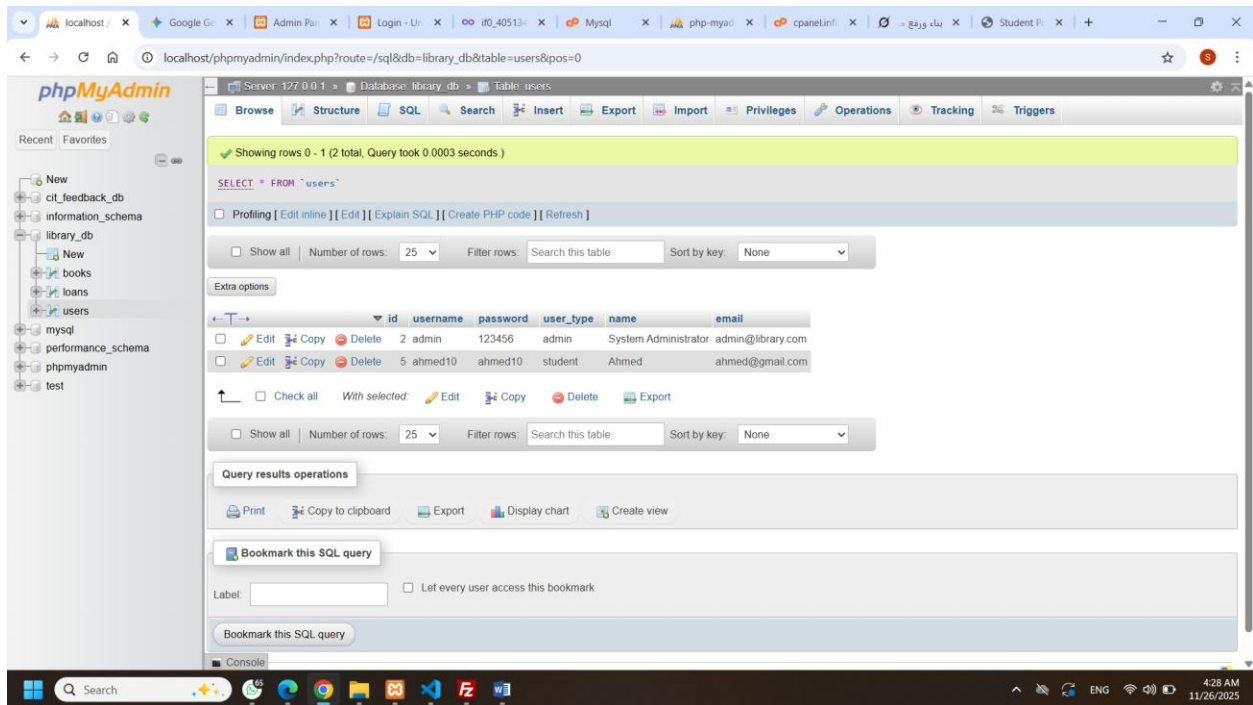


Figure 2: Structure view of the users table.

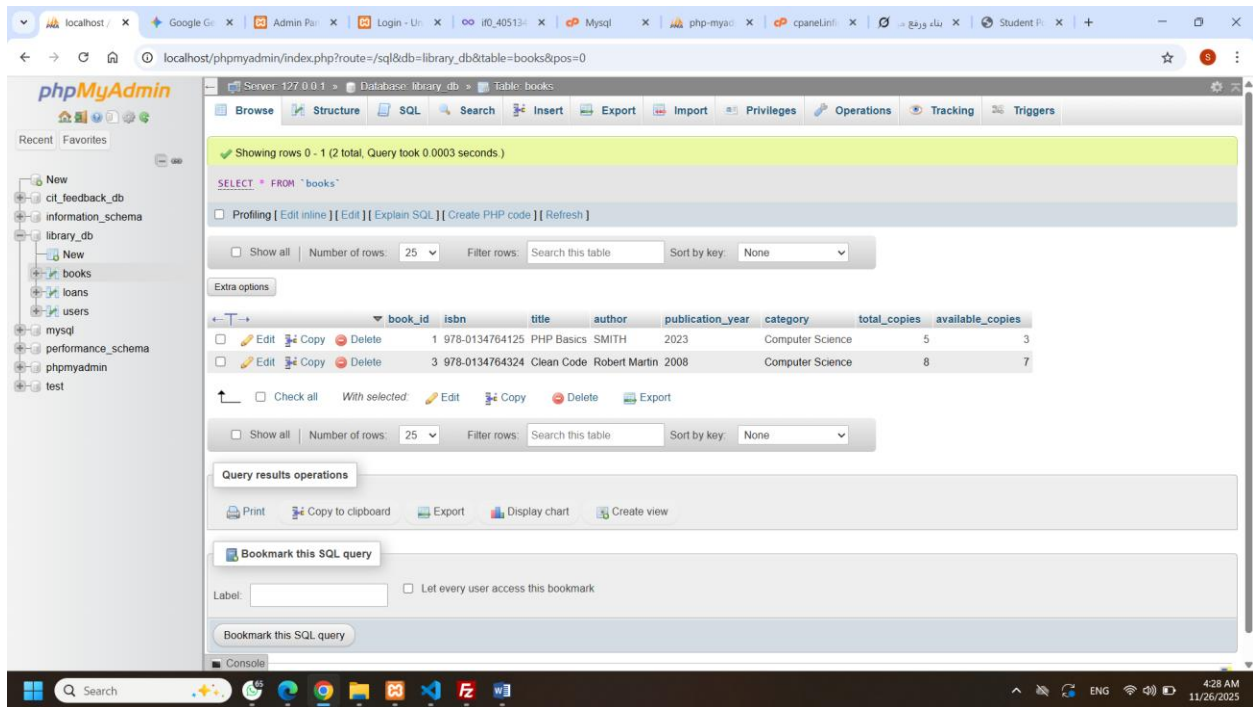


Figure 3: Structure view of the books table.

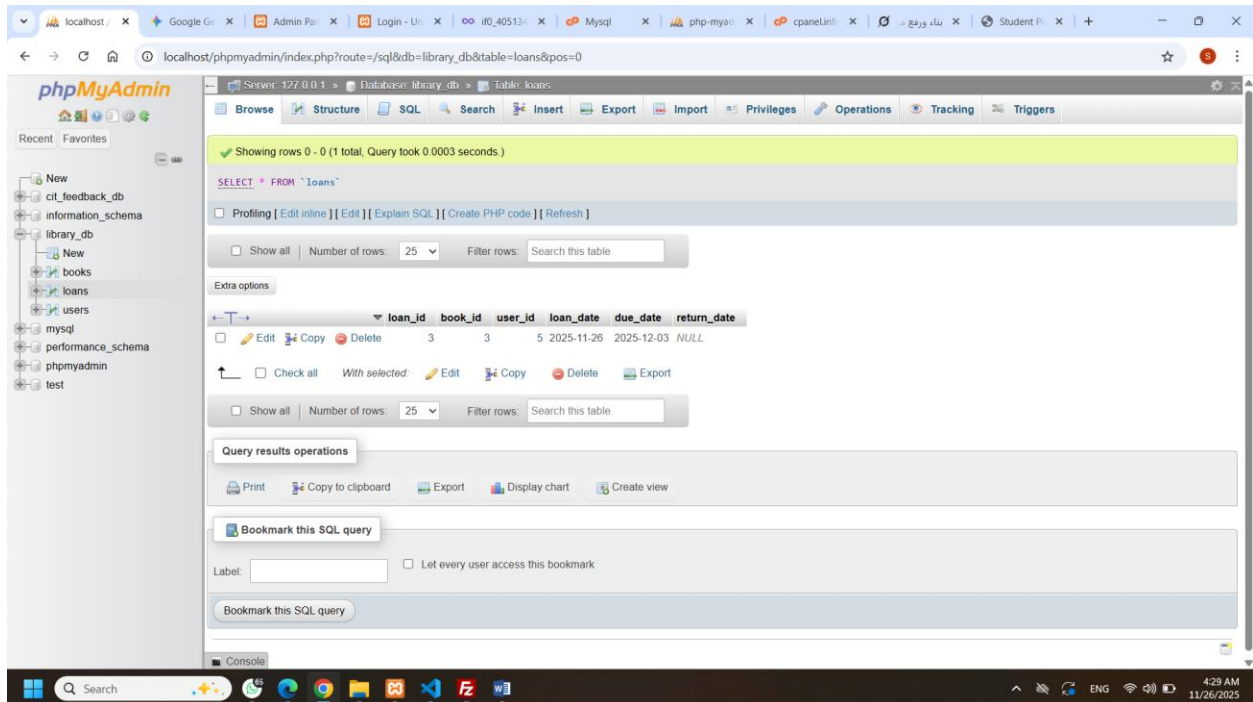


Figure 4: Structure view of the loans table.

System Pages and Features

The system is split into protected portals and public access areas.

Public Access

The system uses the Login Page to authenticate users and the Public Catalog Page (catalog.php) to allow guests to view the entire inventory.

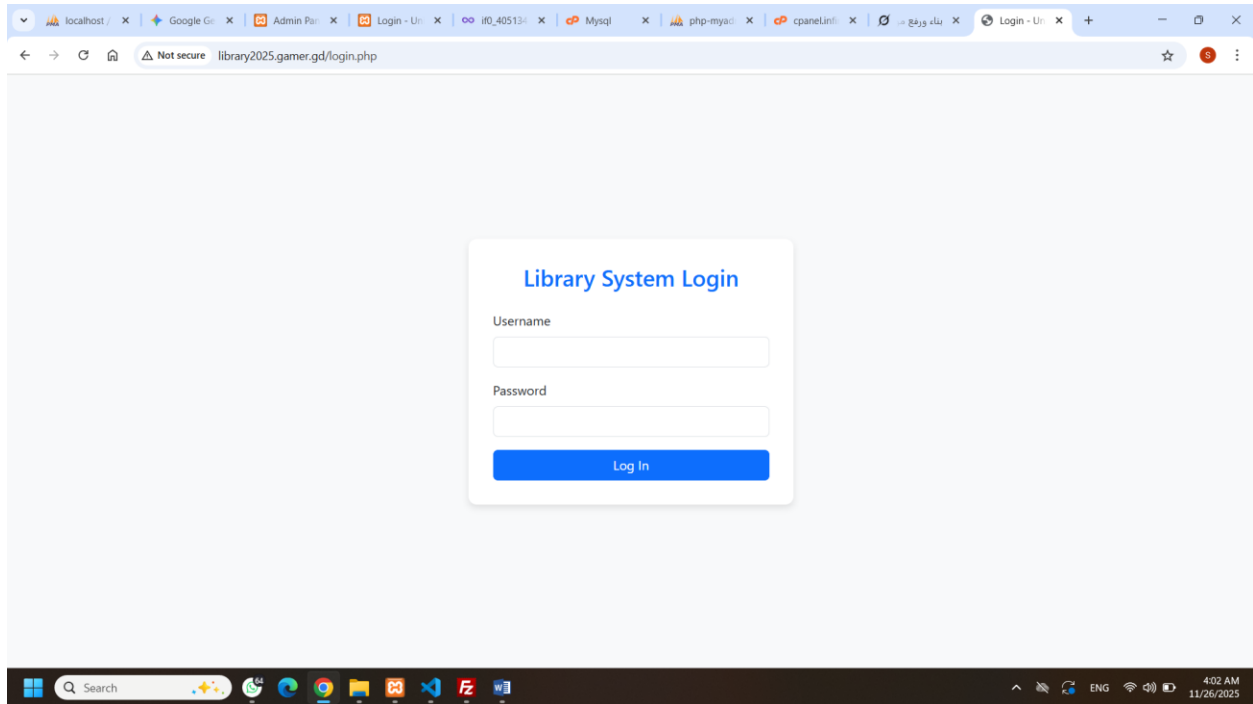


Figure 5: Login page for students and admin.

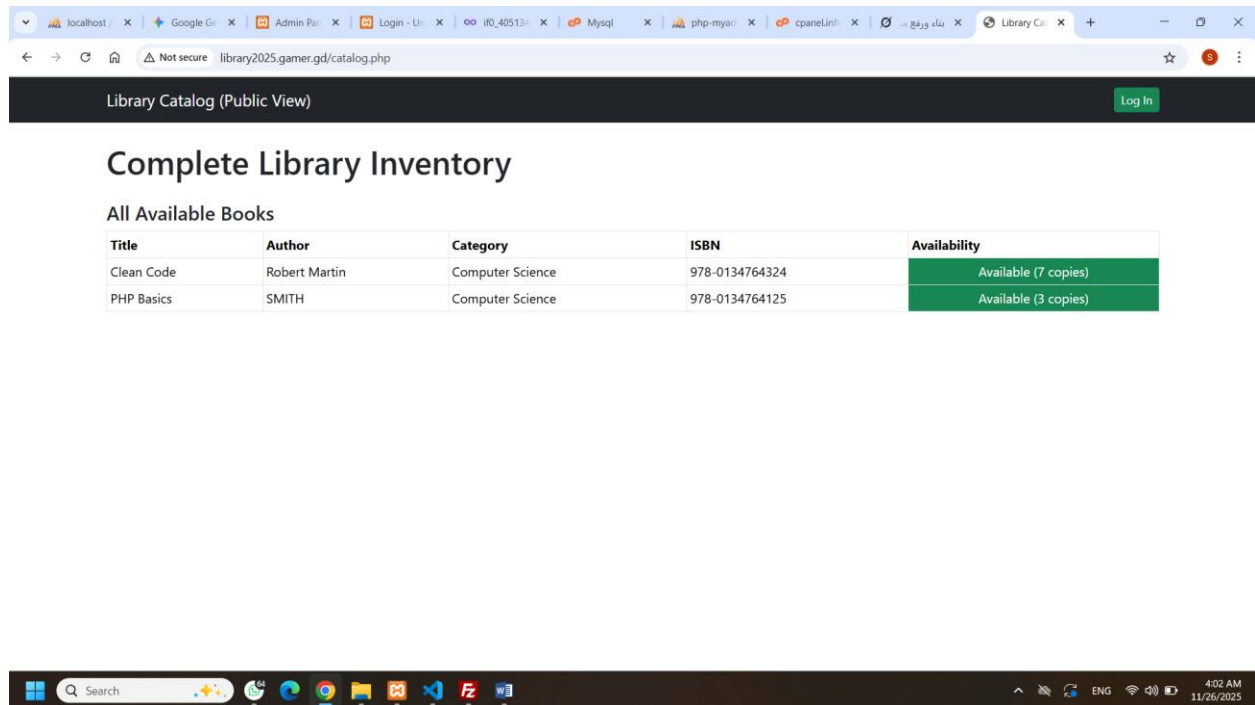


Figure 6: Public Catalog page showing all books and their availability (Guest View).

Admin Portal

The Admin controls all CRUD operations:

- Admin Dashboard: Shows key statistics (Total Books, Current Loans, Total Users).
- Manage Books: Allows the admin to Add (C), View (R), Edit (U), and Delete (D) book records.
- Manage Loans: Allows the admin to perform Check Out (C) and Check In (U) transactions.
- Manage Users: Allows the admin to Add (C), View (R), Edit (U), and Delete (D) user accounts.

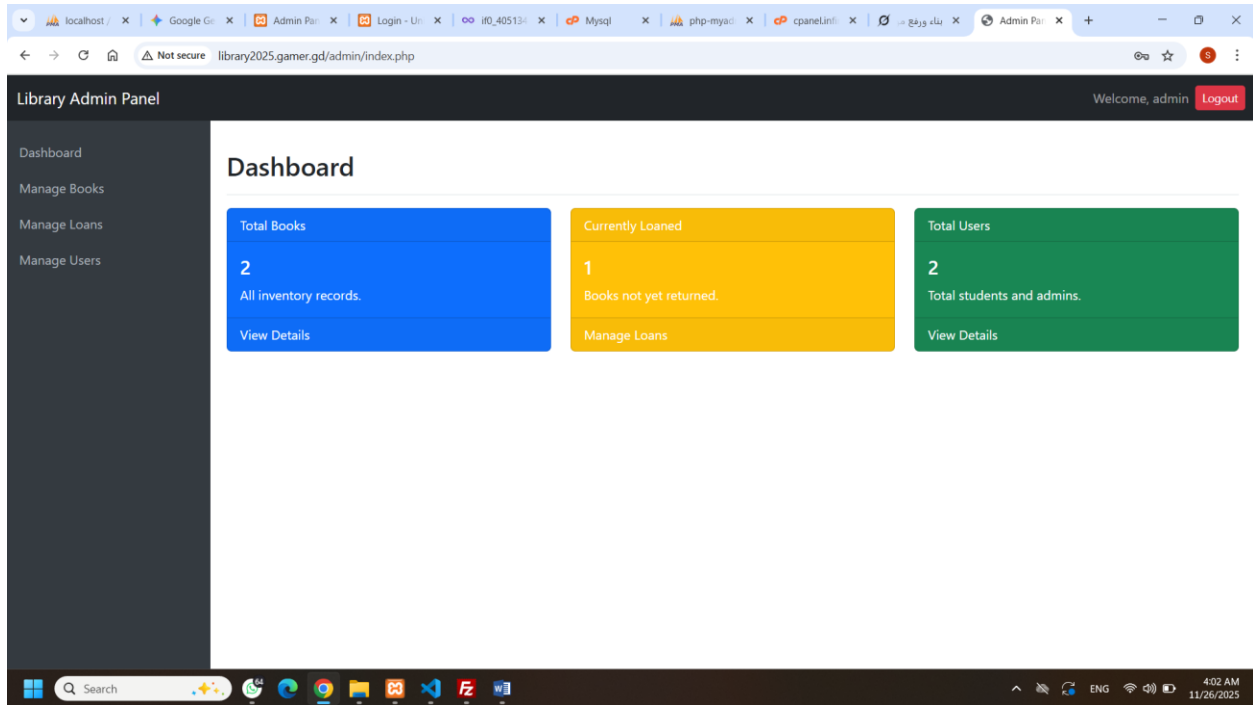


Figure 7: Admin Dashboard with summary statistics.

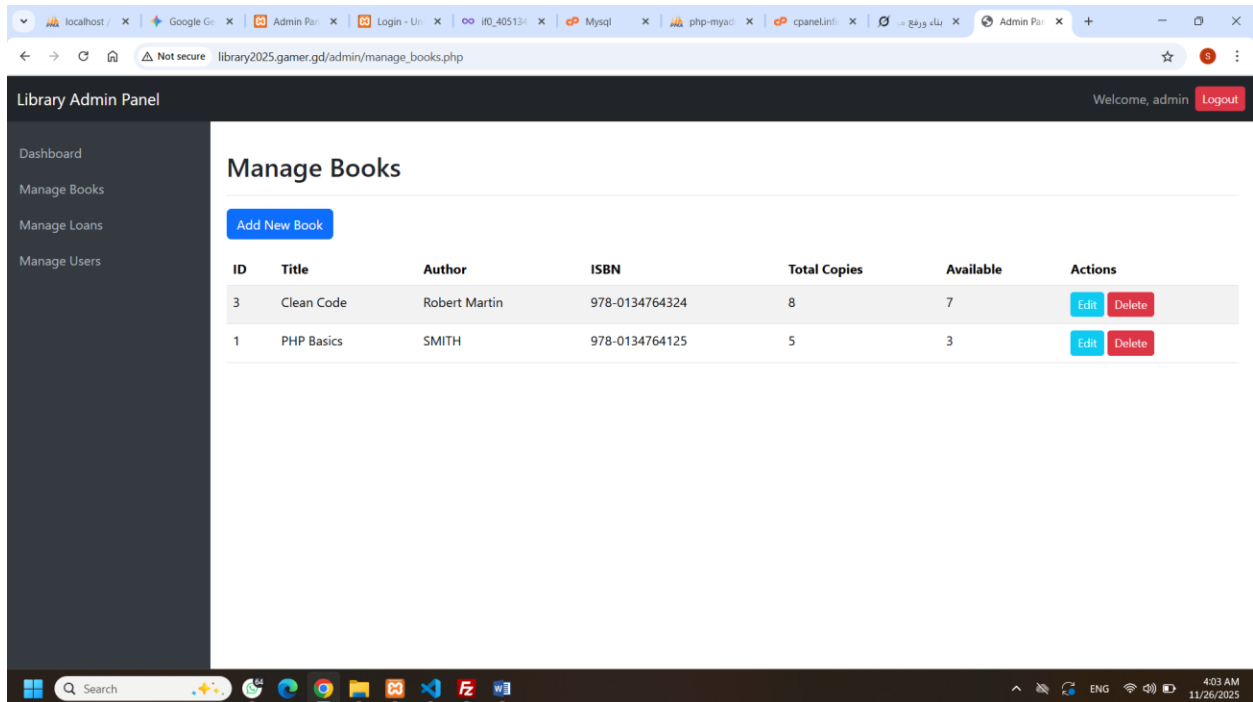


Figure 8: Manage Books page showing CRUD functionality.

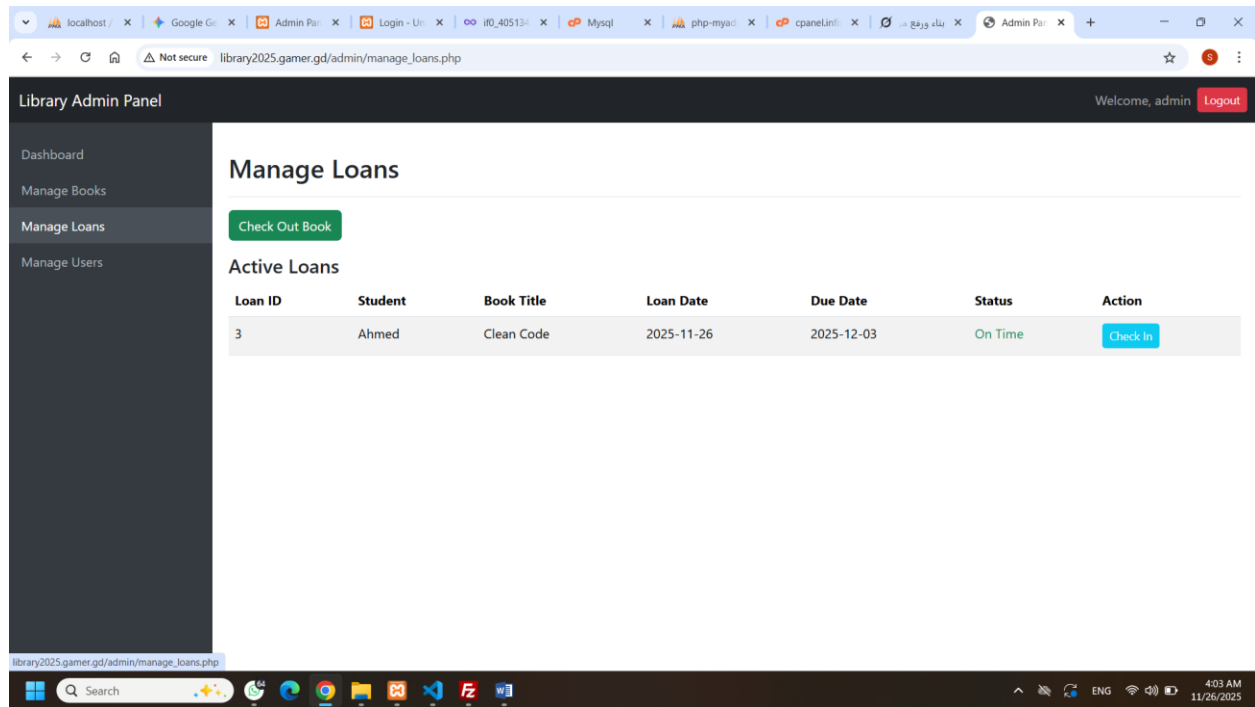


Figure 9: Manage Loans page showing active transactions and Check In option.

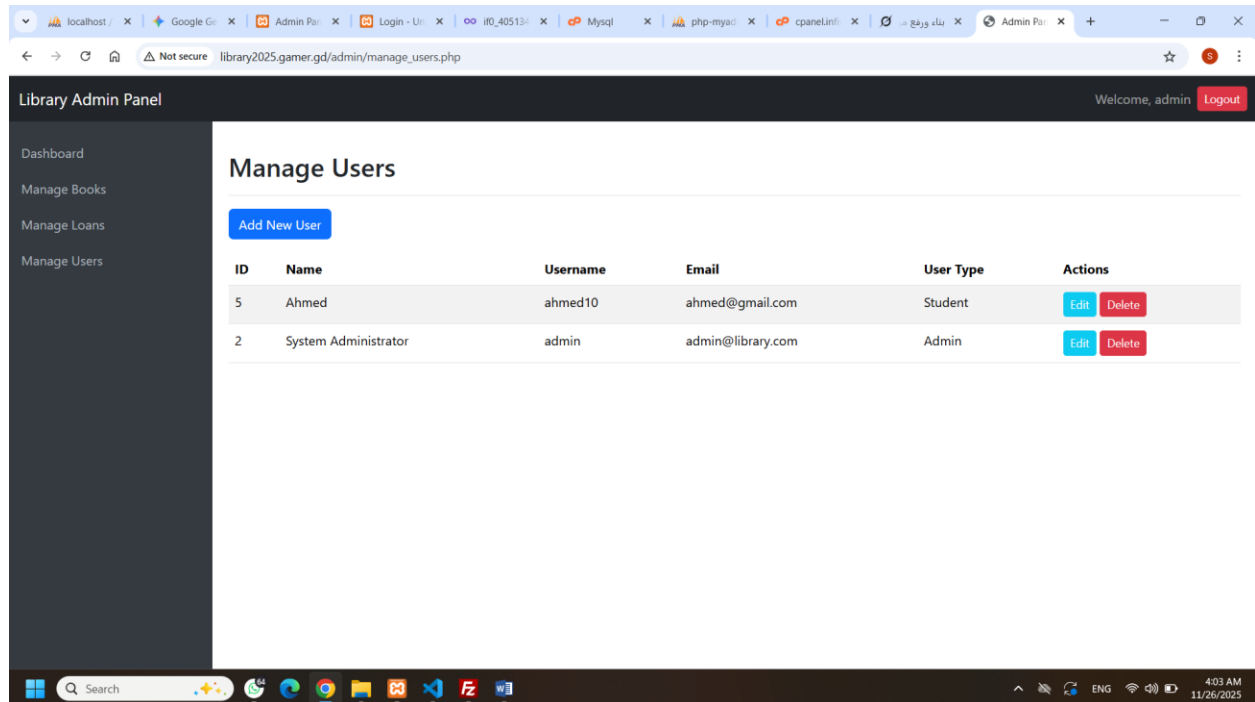


Figure 10: Manage Users page showing user accounts and management tools.

Student Portal

- **Student Portal:** Allows the student to View (R) their active loans, check due dates, and see if any books are overdue.

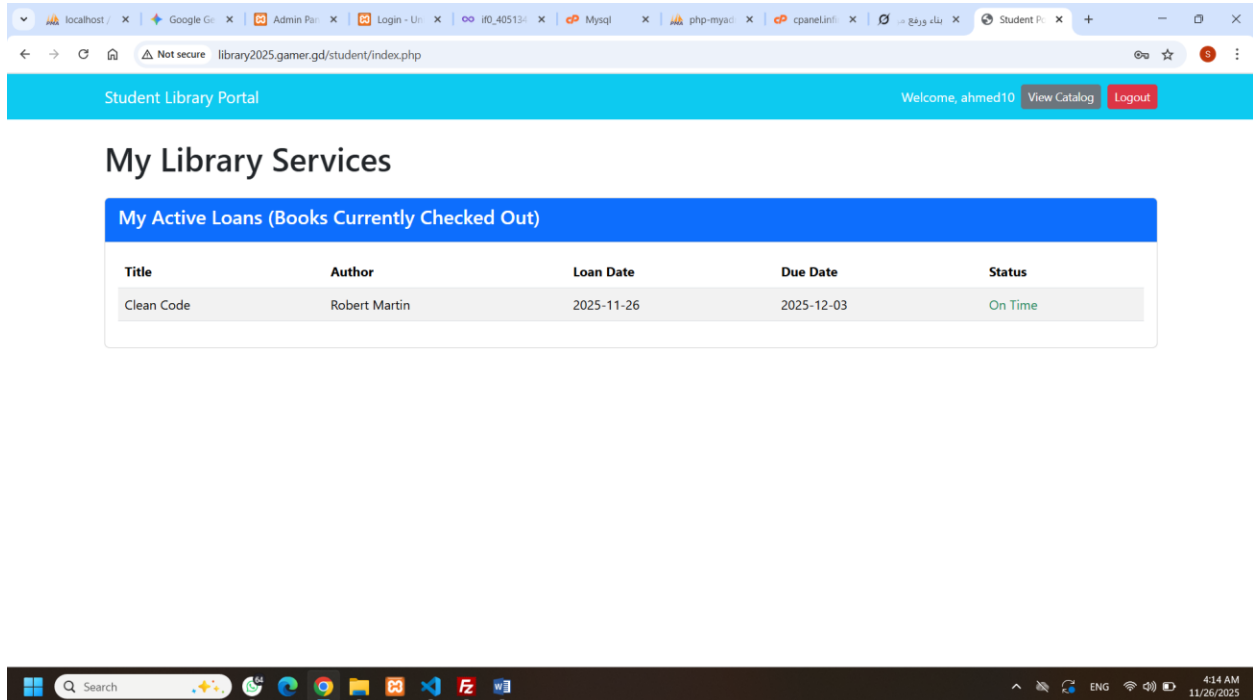


Figure 11: Student Portal showing the list of active loans.

CRUD Operations Summary

The system implements complete Create, Read, Update, and Delete operations:

- **Create (C):** Adding new books and users (manage_books.php, manage_users.php), and creating a new loan transaction (Check Out in manage_loans.php).
- **Read (R):** Displaying all data in lists and reports (manage_books.php, manage_loans.php), and showing user-specific data (My Active Loans).
- **Update (U):** Editing book and user details, and performing a Check In transaction (manage_loans.php) which updates the loan status and the book's availability count.
- **Delete (D):** Removing book records and user accounts (manage_books.php, manage_users.php).

Challenges and Solutions

During development, we successfully overcame critical challenges related to security and database integrity:

1. **Authorization:** We ensured that only users with admin status could access the /admin/ folder by checking the PHP Session (\$_SESSION['user_type']) on every management page.
2. **Referential Integrity (Foreign Keys):** Deleting a user who had active loans, or deleting a book that was currently checked out, would cause a foreign key error. We solved this by implementing checks in the Delete logic to prevent the action if there is an active dependency (i.e., the user must return the book first).
3. **Cross-Page Design:** We ensured a consistent design across the Admin portal by using layout_admin_top.php and layout_admin_bottom.php as template files.

Conclusion

The result is a professional and fully operational Library Management System. We applied core concepts learned in the Advanced Web Technology course, including secure database interaction (PDO), authorization control (PHP Sessions), robust data validation, and responsive design (Bootstrap). The project demonstrates a strong understanding of web application development principles and is ready for institutional use.