



Term: Spring 2025 **Subject:** Computer Science & Engineering (CSE) **Number:** 412
Course Title: Database Management (CSE 412)

Database Project Manual

Project Title: Shelfwise: A smart system to manage library shelves and books wisely.

Team Name: Project Group 4

Team Members' Names and Student IDs:

Sri Sai Varun Bondalapati, Samanvith Reddy Charlapally, Abhijeet Reddy Pailla,
Srinivas Potla

Team Contributions:

Name	Role / Responsibility	% Contribution
Sri Sai Varun Bondalapati	Handled core backend, database integration, and overall project coordination.	100
Abhijeet Reddy Pailla	Managed database dump and supported	100
Srinivas Potla	Developed the transaction module and tested core workflows.authentication and documentation.	100

Samanvith Reddy Charlapally	Worked on frontend views and book management features.	100
--------------------------------	--	-----

1. Overview of the Application

ShelfWise is a comprehensive and intuitive web-based Library Management System designed to digitize and automate core library operations. The application enables librarians to efficiently add, edit, delete, and manage book records while offering library members the ability to browse available books, search by title or genre, borrow books, and track their borrowing history. Each user type librarian or member experiences a customized dashboard tailored to their needs.

The primary objective of ShelfWise is to address the inefficiencies and errors associated with traditional manual library record-keeping. By centralizing operations through an interactive database and user-friendly interface, the system ensures real-time updates, minimizes human error, and provides transparent access to book availability and transaction records.

ShelfWise significantly improves the workflow for library administrators by automating book inventory management and tracking member borrow/return activities. For members, it simplifies the process of locating and borrowing books, reducing wait times and enhancing user satisfaction.

The target users for this system include educational institutions, corporate organizations, or community libraries seeking a modern, reliable, and easily maintainable platform to manage their library resources efficiently. ShelfWise ultimately promotes better organization, accessibility, and user engagement within any library environment.

2. Database Design Summary

The database schema for ShelfWise is designed to efficiently manage users, books, and borrowing transactions while maintaining data integrity through appropriate constraints and relationships.

The system includes the following main tables:

- Librarians: Stores librarian account information including username and password.
- LibraryMembers: Stores member account information including username and password.
- Books: Stores detailed information about books, such as title, author, ISBN, genre, and available copies.
- Transactions: Records each borrowing event, tracking which member borrowed which book, the borrow and return dates, and any applicable fines.

Key Relationships:

- Each transaction is linked to a library member via a foreign key (username).
- Each transaction references the specific book borrowed via a foreign key (title or book_id depending on schema structure).
- The books table maintains a field for available copies, which is updated as transactions occur.

Primary Keys:

- librarians.username
- librarymembers.username
- books.id
- transactions.id

Foreign Keys:

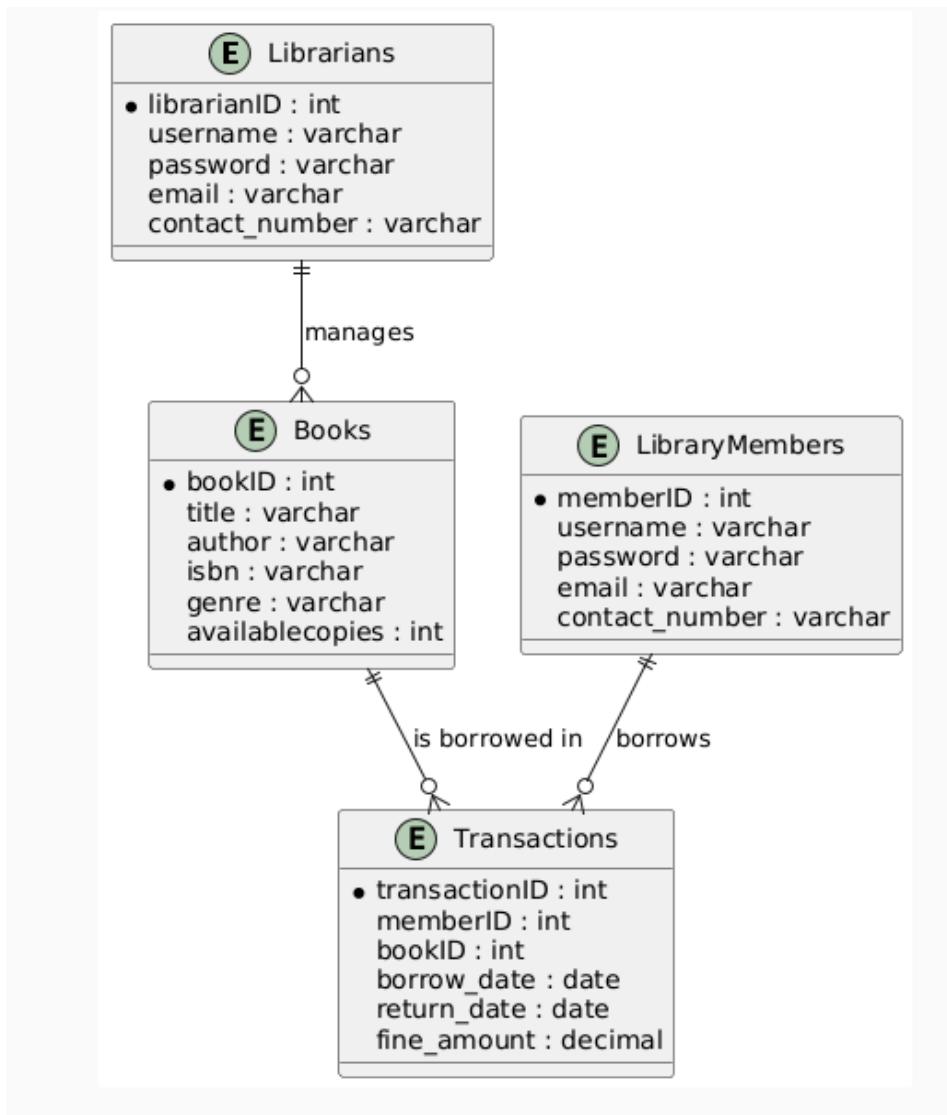
- transactions.username → librarymembers.username

- transactions.book_title → books.title

Constraints:

- Username fields are unique to prevent duplicate accounts.
- Book IDs are auto-incremented and unique.
- Number of copies must be a positive integer.
- Transactions must refer to valid existing users and books (enforced through foreign key constraints).

ER Diagram:



3. Technology Stack

The ShelfWise application is developed using a combination of lightweight, open-source technologies to ensure performance, ease of development, and maintainability.

- Frontend:

The user interface is built using HTML and custom CSS, creating simple and user-friendly web pages for both librarians and members. All styling is embedded directly within the HTML templates.

- Backend:

The server-side logic is implemented using Python Flask, a micro web framework that handles routing, form processing, session tracking, and integration with the database.

- Database System:

ShelfWise uses PostgreSQL, a robust open-source relational database, to store data for users, books, and transactions. The schema is designed with proper normalization and referential integrity using primary and foreign keys.

- APIs / Libraries Used:

- Flask – Web framework for handling application logic and routing
- psycopg2 – PostgreSQL adapter for Python used to perform all database operations

- Other Tools and Platforms:

- GitHub – For version control and public repository hosting
- Visual Studio Code – Primary code editor used during development

This technology stack ensures that the application remains lightweight, easy to deploy, and adaptable for future features or cloud deployment.

4. Setup Instructions

Prerequisites

Before setting up ShelfWise, ensure the following are installed on your machine:

- Python 3.x
- PostgreSQL 9.x or higher
- Git (optional, for cloning the repository)
- Web browser (Chrome, Firefox, or any modern browser)
- Basic knowledge of command-line interfaces

Step 1: Clone the Repository or Copy Files

Copy all project files to your desired directory on your local machine.

Step 2: Set Up the Database

Open PostgreSQL and perform the following:

Create the database:

```
CREATE DATABASE shelfwise_db;
```

Connect to the newly created database:

```
\c shelfwise_db
```

Restore the database structure and initial data using the provided SQL dump:

```
\i databaseDump.sql
```

Step 3: Configure the Application

Open the app.py file in a text editor.

Update the database connection details to match your PostgreSQL setup:

```
conn = psycopg2.connect(
```

```
host="localhost",  
database="shelfwise_db",  
user="your_username",  
password="your_password"  
)
```

Step 4: Install Dependencies

Ensure that Python 3.x is installed on your system.

Install the required Python libraries by running:

```
pip install flask psycopg2
```

Step 5: Run the Application

Start the Flask server by executing:

```
python app.py
```

Open your web browser and navigate to:

<http://127.0.0.1:5000>

You will be redirected to the ShelfWise login page, where you can sign up or log in as a librarian or library member.

Expected Output

- Librarians can log in, add new books, edit or delete books, and view all transaction records.
- Library members can log in, browse available books, borrow books, return books, and view their personal borrowing history.
- All operations are saved live in the PostgreSQL database.

5. Feature List

Login

- Librarians and library members can securely log in using their registered credentials.
- Session management ensures that users access only the features allowed for their role.

Sign-Up

- Separate sign-up options for new librarians and new library members.
- Stores the username and password securely into the database upon registration.

Add Record (Add Book)

- Librarians can add a new book to the database.
- Information fields include Title, Author, ISBN, Genre, and Number of Available Copies.
- Successfully added books are immediately available for viewing and borrowing.

View/Search Records (Books)

- Both librarians and members can view the entire list of available books.
- A search bar allows filtering of books based on partial or full title matches.

Update / Edit Record (Edit Book Information)

- Librarians can edit existing book records to update Title, Author, ISBN, Genre, or the number of Available Copies.
- Updates are reflected immediately in the database after saving changes.

Delete Record (Remove Book)

- Librarians can delete any book record from the system.

- Deletion permanently removes the book information from the database.

Borrow Book (Create Transaction)

- Library members can borrow available books.
- Borrowing creates a new transaction record in the database with the member's ID, the book's ID, and the borrow date.

Return Book (Update Transaction)

- Library members can return books they have borrowed.
- Upon return, the transaction record is updated with the return date, and any applicable fine is calculated if the return is late.

My Transactions (View Borrowed Books)

- Library members can view a list of all books they have borrowed, their borrow dates, return dates (if returned), and fines.

All Transactions (View All Borrow Activities)

- Librarians can view all transactions made by all members.
- This includes records of who borrowed which book, when, and whether it was returned.

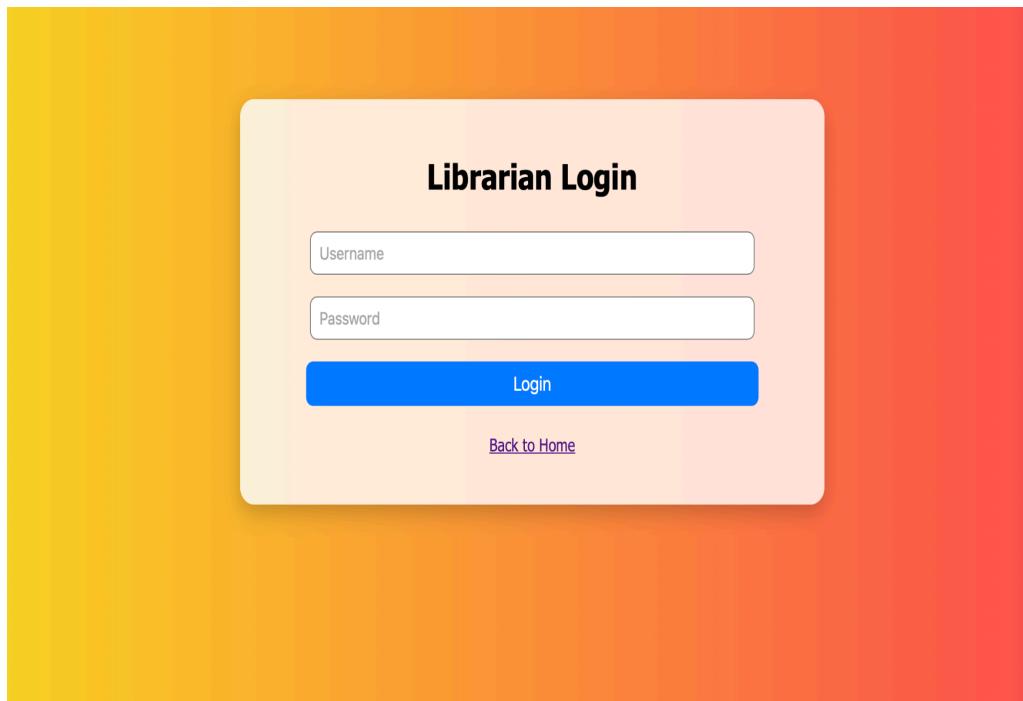
Error Handling

- Invalid operations (like trying to return a book that is not borrowed) are properly handled with redirects or informative error pages.

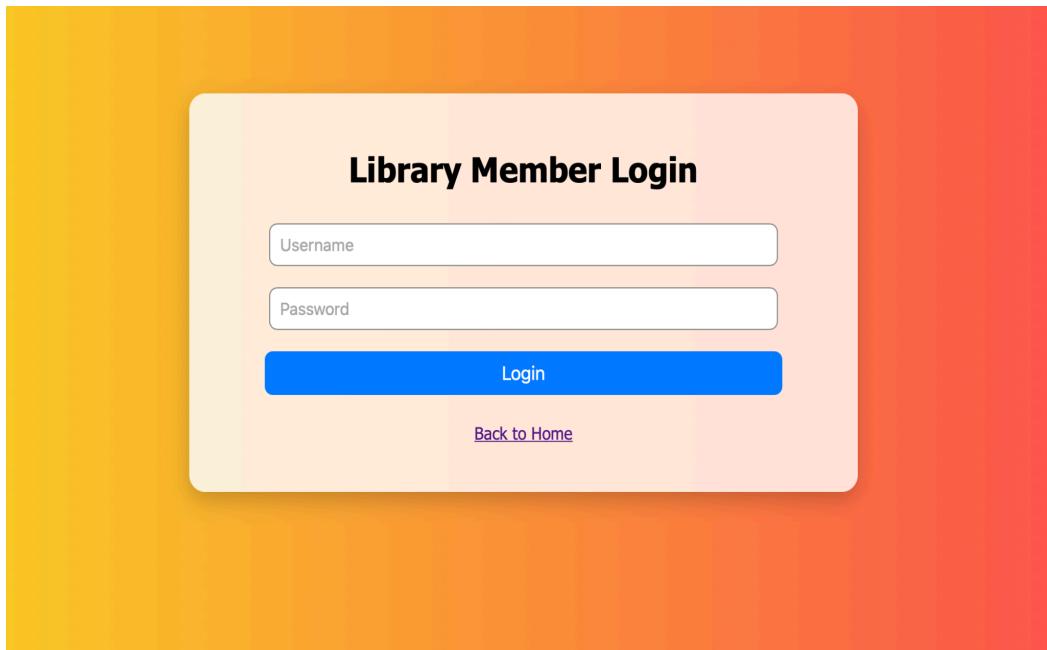
6. Screenshots



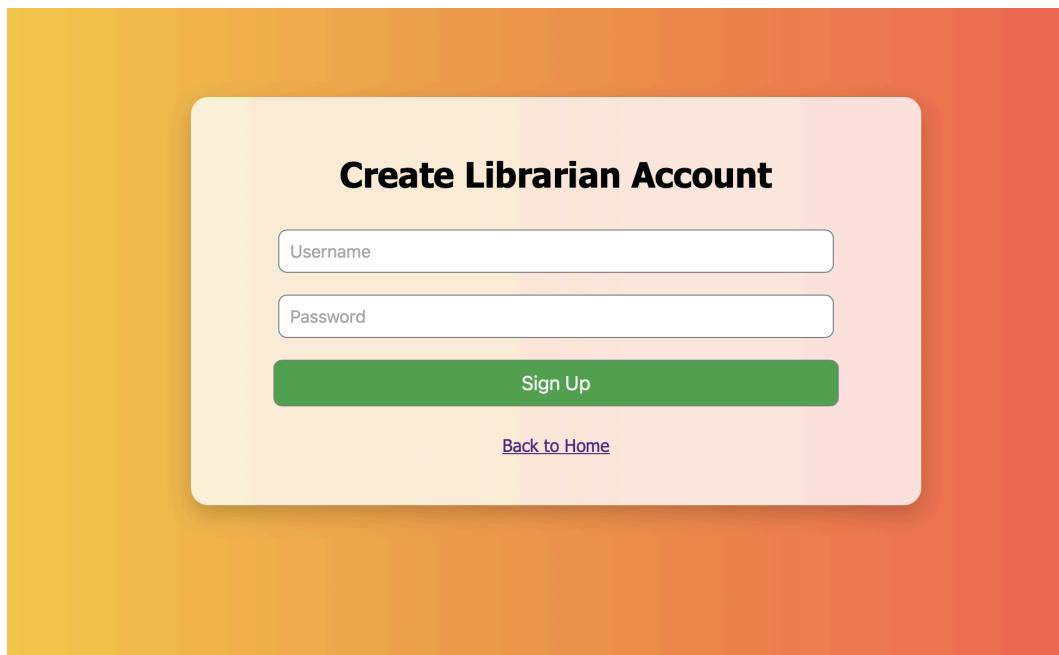
When you first open `http://127.0.0.1:5000` — before logging in.



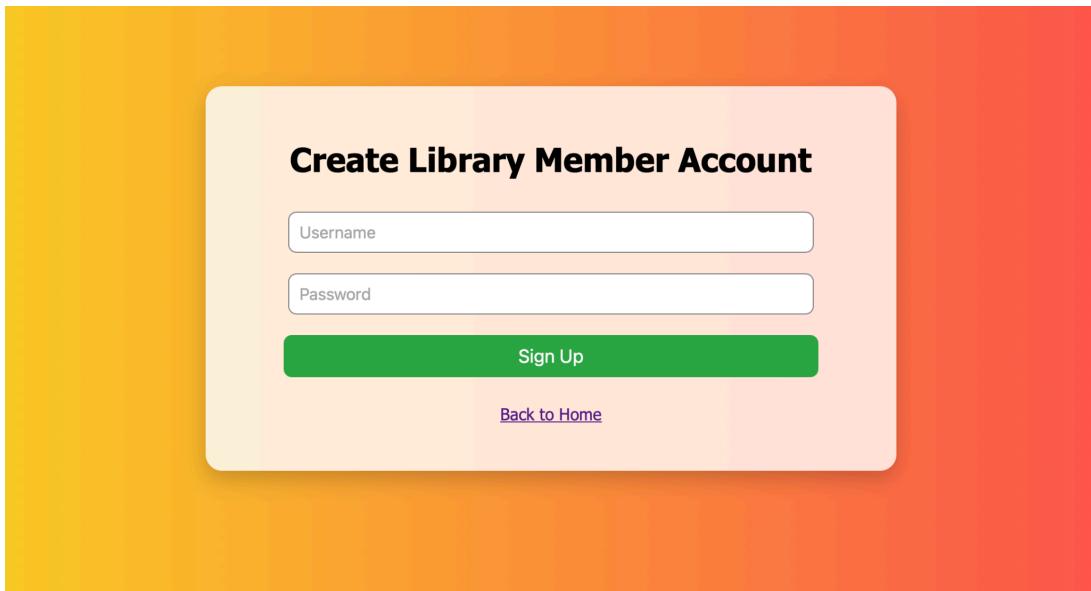
Login page where Librarians can enter their credentials to access ShelfWise.



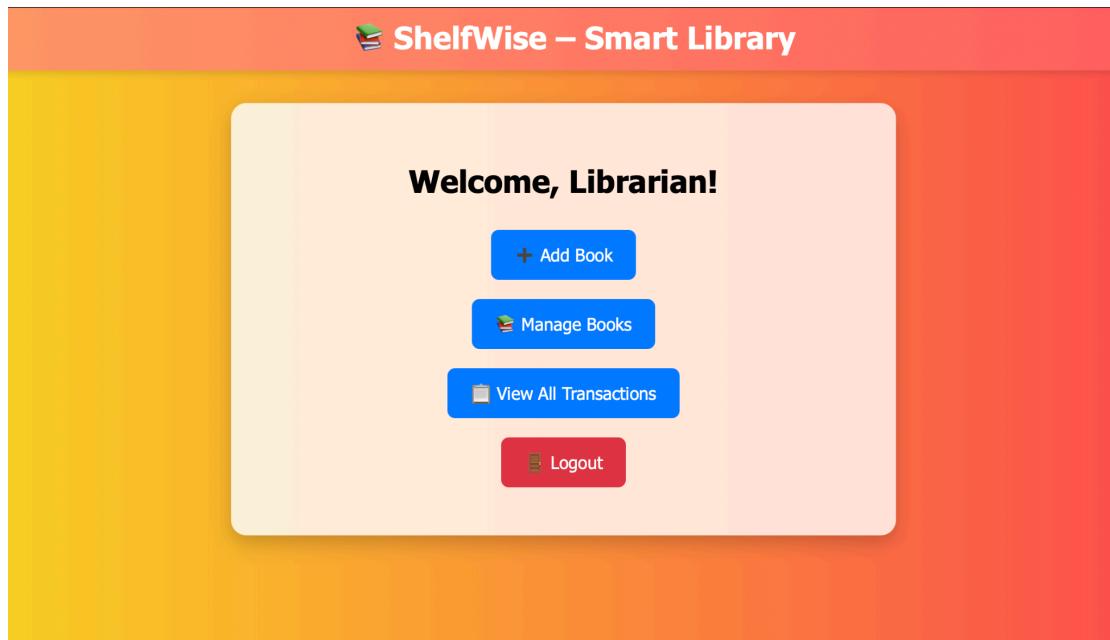
Login page where Library Members can enter their credentials to access ShelfWise.



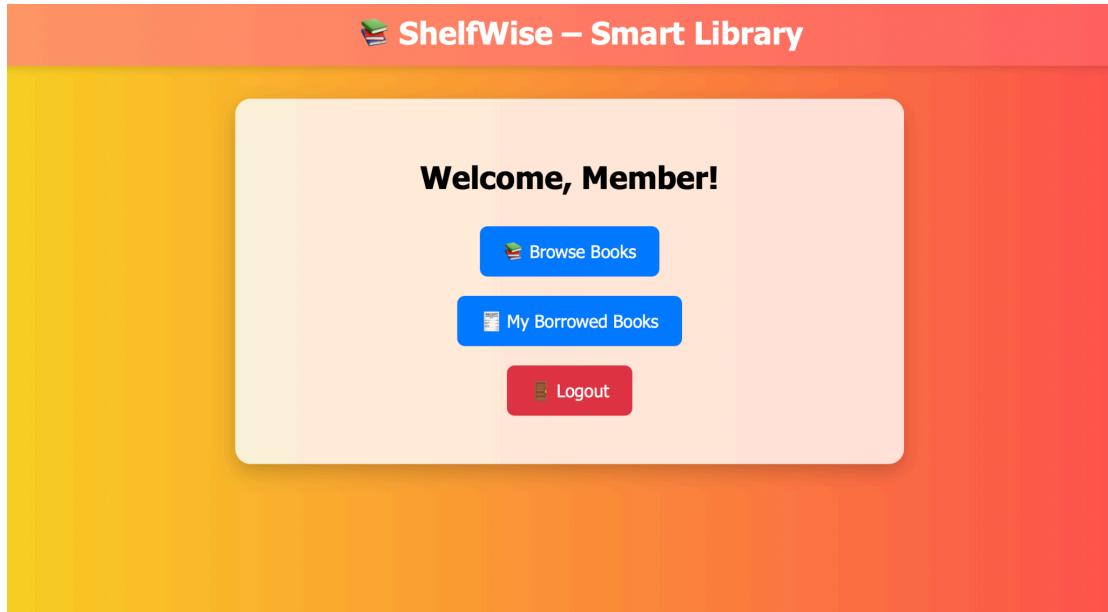
Registration page for new Librarians to create an account.



Registration page for new Library Members to create an account.



Dashboard view after librarian login showing management options.



Dashboard view after member login showing available features.

A screenshot of the ShelfWise Smart Library available books page. The background has a vertical gradient from yellow to red. At the top center is the ShelfWise logo. Below it is a "Back to Dashboard" button. The main section is titled "Available Books" in bold black font. It features a search bar with the placeholder "Search books..." and a "Search" button. Below the search bar, there are five horizontal entries, each representing a book: 1. "Educated" by Tara Westover | Genre: Biography | Copies: 6, with a "Borrow" button. 2. "Where the Crawdads Sing" by Delia Owens | Genre: Mystery | Copies: 8, with a "Borrow" button. 3. "The Alchemist" by Paulo Coelho | Genre: Fantasy | Copies: 4, with a "Borrow" button. 4. "Becoming" by Michelle Obama | Genre: Biography | Copies: 5, with a "Borrow" button. 5. "Dune" by Frank Herbert | Genre: Science Fiction | Copies: 6, with a "Borrow" button.

Page displaying the list of available books for members.

The screenshot shows a web application interface titled "ShelfWise – Smart Library". The main title is at the top left of a red header bar. Below it is a large orange gradient background area containing a white rectangular form. The form has a title "Add New Book" at the top. It contains five input fields: "Title", "Author", "ISBN", "Genre", and "Available Copies". Below these is a green button labeled "Add Book". At the bottom of the form is a link "Back to Dashboard" with a house icon.

Form page where librarians can add a new book to the system.

Available Books	
<input type="text" value="Search books..."/>	<input type="button" value="Search"/>
Educated by Tara Westover Genre: Biography Copies: 6	
<input type="button" value="Edit"/>	<input type="button" value="Delete"/>
<hr/> Where the Crawdads Sing by Delia Owens Genre: Mystery Copies: 8	
<input type="button" value="Edit"/>	<input type="button" value="Delete"/>
<hr/> The Alchemist by Paulo Coelho Genre: Fantasy Copies: 4	
<input type="button" value="Edit"/>	<input type="button" value="Delete"/>
<hr/> Becoming by Michelle Obama Genre: Biography Copies: 5	
<input type="button" value="Edit"/>	<input type="button" value="Delete"/>

Page displaying the list of available books for librarians.

The screenshot shows a web application interface titled "ShelfWise – Smart Library". The main title is at the top left of the header bar. Below it, a large orange gradient background area contains a white rectangular card. The card has a title "Edit Book" in bold black font at the top center. Inside the card, there are five input fields: "Educated", "Tara Westover", "9780399590504", "Biography", and "6". Below these fields is a yellow button labeled "Update Book". At the bottom of the card is a blue "Back to Dashboard" button with a house icon.

Form page where librarians can edit an existing book's details.

The screenshot shows a web application interface titled "ShelfWise – Smart Library". The main title is at the top left of the header bar. Below it, a large orange gradient background area contains a white rectangular card. The card has a "Back to Dashboard" button at the top. Below it is a section titled "My Borrowed Books" in bold black font. It lists three books with their details: "Where the Crawdads Sing" (Borrowed: 2025-04-26 02:09:32.106311 | Not Returned Yet), "The Alchemist" (Borrowed: 2025-04-26 02:08:22.202793 | Returned: 2025-04-26 02:08:29.605311 | Fine: \$0.00), and "Educated" (Borrowed: 2025-04-26 02:06:24.685499 | Returned: 2025-04-26 02:07:59.202886 | Fine: \$0.00). Each book entry is separated by a horizontal line.

Page showing the list of books borrowed by the member.

 ShelfWise – Smart Library[!\[\]\(a48045bf840f60e99d28ce32cf91bb81_img.jpg\) Back to Dashboard](#) All Borrowing Records

Transaction ID	Member Username	Book Title	Borrow Date	Return Date	Fine Amount (\$)
3	Member	Where the Crawdads Sing	2025-04-26 02:09:32.106311	Not Returned	0.00
2	Member	The Alchemist	2025-04-26 02:08:22.202793	2025-04-26 02:08:29.605311	0.00
1	Member	Educated	2025-04-26 02:06:24.685499	2025-04-26 02:07:59.202886	0.00

Page showing all borrowing and returning transactions across all members.

404 - Page Not Found

The page you are looking for doesn't exist!

[Return to Home](#)

Error page displayed for invalid routes or operations.

7. Video Demonstration Link

- <https://youtu.be/QdNpMj6fAhY>

8. GitHub Repository Link

- <https://github.com/Varun27bondala/CSE-412-Group-4-Shelfwise>

References:

- *Flask Documentation. Flask Web Development Documentation.* Retrieved from <https://flask.palletsprojects.com/>
- *psycopg2 Documentation. PostgreSQL Adapter for Python.* Retrieved from <https://www.psycopg.org/docs/>
- *PostgreSQL Official Documentation. The World's Most Advanced Open Source Relational Database.* Retrieved from <https://www.postgresql.org/docs/>
- *GitHub Documentation. GitHub Docs – Understanding the GitHub flow.* Retrieved from <https://docs.github.com/en>
- *W3Schools. HTML, CSS, and SQL Tutorials.* Retrieved from <https://www.w3schools.com/>
- *Stack Overflow Discussions. Community discussions and solutions for Flask, PostgreSQL, and psycopg2 issues.* Retrieved from <https://stackoverflow.com/>