Table of content

1. Introduction

The Library Management System (LMS) is an innovative solution designed to automate and streamline the management of library resources. It replaces traditional manual processes with a user-friendly platform that facilitates efficient book cataloging, user registration, borrowing, returning, and searching for books. The system enhances accuracy, minimizes human error, and integrates advanced features such as undo functionality, cart-based borrowing, and secure login mechanisms. The goal of this project is to modernize library management practices, ensuring scalability and adaptability to meet the evolving needs of libraries.

2. Background Study

Libraries have historically relied on manual methods for managing resources, which often led to inaccuracies, inefficiencies, and difficulties in tracking books and transactions. Automated Library Management Systems (LMS) were introduced to address these challenges, streamlining processes like cataloging, borrowing, and returning books while reducing human error.

Existing LMS platforms, such as those developed by Stuti Mongia, A133uz, and F4H1M4HMED, demonstrate various approaches to library automation. However, these systems often lack critical features like undo functionality, cart-based borrowing, and dynamic book management. This project builds upon these foundations, introducing improvements to enhance usability, security, and operational efficiency.

3. Problem Statement

Manual library management is plagued by inefficiencies, errors, and difficulties in tracking books. Existing automated solutions often lack essential features, such as:

- Undo functionality to reverse accidental actions.
- Cart-based borrowing for selecting multiple books at once.
- Flexible transaction handling and real-time updates on book availability.

These limitations highlight the need for a comprehensive LMS that addresses these gaps while introducing innovative features to improve user experience and operational efficiency.

4. Problem Solution

The proposed LMS offers a user-friendly and efficient solution with the following features:

- Easy search and borrowing (similar to online shopping).
- Secure login mechanisms for staff and users.
- Real-time updates on book availability.
- Undo functionality to correct mistakes.
- Cart-based borrowing for selecting multiple books.
- Genre-based filtering for easier navigation.

By automating tasks and integrating these features, the system reduces errors, saves time, and ensures smooth library operations.

5. Objectives

The primary objectives of this project are:

- Design a user-friendly LMS to simplify library operations.
- Implement core features such as user registration, book search, borrowing, and returning.
- Integrate advanced functionalities like undo actions and cart-based borrowing.
- Ensure scalability to accommodate growing libraries.
- Provide efficient book tracking to minimize misplacement and ensure accurate availability updates.

6. Methodology

6.1 Tools and Techniques

- Programming Language: Python
- IDE: Visual Studio Code
- AI-Assisted Tools: GitHub Copilot & ChatGPT
- Diagramming Tools: Lucidchart/Draw.io
- Code Editor: Cursor AI

6.2 Functional and Non-Functional Requirements

Functional Requirements:

- User registration, login, and account management.
- Book search, borrowing, and returning with real-time updates.
- Cart-based borrowing for multiple books.

- Undo functionality to reverse actions.
- Staff capabilities to add, remove, or update book details.

Non-Functional Requirements:

- User-friendly interface.
- Scalability to handle large numbers of users and books.
- Data accuracy and security.

6.3 System Requirements

Software

- A basic computer with at least 4GB RAM and any processor capable of running a Python interpreter

Hardware

- Python 3.6 or above installed, along with a basic code editor or IDE such as IDLE, VS Code, or PyCharm.

6.4 Flowchart Description

The flowchart illustrates the complete process flow of the Library Management System, covering all types of users including registered users, guests, and administrators. The system initiates from the Start node and provides the user with three primary options: Login, Register, or Continue as Guest.

If the user chooses to register, they must enter their name, password, and card details. A condition checks whether they are willing to pay a monthly fee—if not, they are redirected to the start. Otherwise, their information is verified and access is granted. Alternatively, if the user continues as a guest, they must pay per book and provide temporary credentials before accessing book options.

In the Login path, users and admins are separated. Regular users must enter valid credentials; if correct, they can view books, add them to cart, and borrow or return books. Admins follow a similar login process but have access to manage members and books, including adding or removing both.

6.5 Flowchart Diagram

7. Results

- Undo functionality for reversing unintended actions.
- Genre-based categorization for easier navigation.
- Real-time updates on book availability.

Table No.1: Survey Analysis

Code Feature	Cart	Undo	Account	Genre	Add Books	Return Books	Update Available Books	User Entry
Project 1	✗	✗	✗	✗	☑	☑	✗	☑
Project 2	✗	✗	☑	✗	✗	☑	✗	✗
Project 3	✗	✗	✗	✗	☑	☑	☑	☑
Proposed project	☑	☑	☑	☑	☑	☑	☑	☑

8. Conclusion

The LMS successfully modernizes library management by automating processes, reducing errors, and enhancing user experience. Key features like cart-based borrowing, undo functionality, and secure logins ensure efficiency and reliability. The system's scalability and modular design make it adaptable for libraries of all sizes, positioning it as a comprehensive solution for the digital age.

9. Acknowledgement

I would like to express my deepest gratitude to Mr. Aqib Rauf for his invaluable guidance, support, and encouragement throughout the development of this project. His expertise and insightful suggestions greatly contributed to the successful completion of my work.

I am also thankful to my classmates and friends for their cooperation and moral support, as well as to my family for their continuous encouragement.

This project would not have been possible without the help and motivation provided by all those mentioned above.

10. References

1. Stuti Mongia's Library Management System. Source Code. Retrieved from https://github.com/stutimongia2024/Library-Management-System/blob/main/source_code.py

2. A133uz's Simple Library Management System. Source Code. Retrieved from https://github.com/A133uz/simple-library-management-system/blob/main/library.py

3. F4H1M4HMED's Library Management System. Source Code. Retrieved from https://github.com/F4H1M4HMED/Library-Management-System/blob/main/main.py

4. Python Official Documentation - Libraries for Data Structures and Automation. Retrieved from https://docs.python.org

5. Visual Studio Code IDE - Official Documentation and Extensions. Retrieved from https://code.visualstudio.com/docs