A

Project Report

on

Build a Library Management System for our Departmental Library

Master of Science in (Computer Science)

दक्षिण बिहार केंद्रीय विश्वविद्यालय

Central University of South Bihar

Under The Supervision of

Dr. Nemi Chandra Rathore

Assistant Professor

Department of Computer Science

Submitted By:

Abhishek Kumar (CUSB2202312001)

Sandeep Kumar Chauhan (CUSB2202312025)

Department of Computer Science

School Of Mathematics, Statistics & Computer Science

Central University of South Bihar, Gaya

December, 2023

दक्षिण बिहार केंद्रीय विश्वविद्यालय
Central University of South Bihar

CANDIDATE'S DECLARATION

We hereby certify that the work which is being presented in the project entitled **"Build a Library Management System for our Departmental Library"** in partial fulfillment of the requirements for the completion of the Mini-Project of the **Master of Science in Computer Science** submitted in the Department of Computer Science of Central University of South Bihar, Gaya, is an original work carried out during the period of November, 2023, under the supervision of **Dr. Nemi Chandra Rathore (Assistant Professor),** Department of Computer Science, Central University of South Bihar, Gaya.

The matter presented in the project has not been submitted by us for the award of any other project of this or any other places.

Abhishek Kumar (CUSB2202312001)

Sandeep Kumar Chauhan (CUSB2202312025)

This is to certify that the above statement made by the candidates is correct to the best of my knowledge.

Dr. Nemi Chandra Rathore

(Assistant Professor)

ACKNOWLEDGEMENT

We take this occasion to thank God, almighty for blessing us with his grace and takingour endeavor to a successful culmination. We extend our sincere and heartfelt thanks to our esteemed guide,

Dr. **Nemi Chandra Rathore** for providing us with the right guidance and advice at the crucial junctures and for showing us the right way. We extend our sincere thanks to our respected guide **Mr. Purnendu Prabhat,** for Guidance and support to do the project work.

We would like to thank the other faculty members also, at this occasion. Last but not the least; we would like to thank to our team members & friends for the support and encouragement they have given us during the Project Report of our work.

Thank You,
Sincerely,

Abhishek Kumar (M.Sc. CS)

Sandeep Kumar Chauhan (M.Sc. CS)

ABSTRACT

Task management is a ubiquitous challenge faced by individuals, often complicated by the absence of accessible and user-friendly solutions. Traditional methods for organizing tasks are often cumbersome and lack essential features, leading to inefficiencies and disorganization in personal and professional settings. The absence of a streamlined task management tool creates a need for a solution that is both simple and effective. The Library Management System App in Python, developed with Tkinter, addresses the existing problem by offering a straightforward and intuitive task management platform. The application allows users to effortlessly add, view, edit, and delete tasks through a user-friendly graphical interface. The proposed solution aims to enhance the user experience by providing a tool that is not only functional but also adaptable for future feature enhancements. The Library Management System App leverages Python as the primary programming language and Tkinter as the GUI toolkit. These technologies are chosen for their simplicity, readability, and the ability to create a visually appealing interface. The application development process also involves an integrated development environment (IDE), version control with Git, and potential documentation tools for clarity and collaboration. The Library Management System App successfully achieves its objectives, offering users a simple yet effective platform for managing tasks. Through the implementation of core functionalities, including error handling and scalability, the application provides a seamless experience. User feedback and potential future testing will contribute to refining the app, ensuring its usability and relevance. In conclusion, the Library Management System App presents a practical solution to the existing problem of inefficient task management. Its user-friendly design and adaptability lay the groundwork for future enhancements, such as task prioritization and due dates. The conclusion reflects the application's current utility and highlights its potential for growth and improvement based on evolving user needs and technological advancements.

Keyword: *Library Management System App, User-friendly, Task management, Reminders*

Table of Contents

List of Figures

Chapter 1
Introduction

1. Introduction

The Library Management System App is a Python-based application developed using the Tkinter library. This report provides an overview of the application's features, design, and functionality. The app allows users to manage tasks through a user-friendly graphical interface, incorporating essential task management operations such as adding, viewing, editing, and deleting tasks. The purpose of this project was to design and implement a Library Management System App, a tool that helps users organize and manage their tasks efficiently. The app allows users to create, edit, and delete tasks. The project aimed to provide a simple and user-friendly interface for task management. The objectives of the project are clearly outlined, indicating the specific goals the team aimed to achieve. These objectives serve as a roadmap for the reader, detailing the intended outcomes of the Library Management System App. The introduction highlights the key features and functionalities of the Library Management System App. This section gives the reader a glimpse into the capabilities of the application. The overarching purpose of the report is communicated, emphasizing that it serves as a documentation of the project's development process, challenges faced, and the resulting Library Management System App. The report is positioned as a valuable resource for understanding the project's evolution.

1.1 Formulation of Problem

The Library Management System addresses the challenge of efficiently managing and organizing a library's resources, including books and patron information. The system aims to streamline the process of book management, borrowing, and returning through a graphical user interface (GUI). However, several key issues and challenges prompt the need for the development and enhancement of this system:

Manual Data Management:

The current system relies on manual data entry for adding books and patrons, which can be time-consuming and prone to errors. This manual process may hinder the overall efficiency of library operations.

Limited Data Persistence:

The absence of database integration limits the system's ability to persistently store data. This poses a challenge as the system does not retain information between sessions, potentially leading to data loss.

Lack of Error Handling:

The system lacks robust error-handling mechanisms. This can result in issues such as attempting to borrow non-existent books or returning books that have not been borrowed. Improved error handling is crucial for a more reliable user experience.

Insufficient User Feedback:

The system currently provides basic feedback through message boxes. However, more sophisticated and informative feedback mechanisms, such as notifications or status messages, are essential for user guidance and clarity.

Security Concerns:

The absence of user authentication and authorization mechanisms raises security concerns in a multi-user environment. Without proper access controls, unauthorized users may manipulate the library's data, compromising its integrity.

Basic User Interface:

The GUI, while functional, lacks certain features that could enhance user experience. Incorporating advanced features such as search functionality, sorting options, and a visually appealing layout can contribute to a more user-friendly interface.

Scalability and Customization:As libraries vary in size and complexity, the current system may not fully meet the diverse needs of different libraries. Considerations for scalability and customization are necessary to adapt the system to various library requirements.

1.2 Tools and Technology Used

The following technologies and tools were used in the development of the Library Management System App:

I. **Python:**

Python serves as the primary programming language for developing the application. Its simplicity, readability, and extensive standard libraries make it an ideal choice for rapid application development.

II. **Tkinter:**

Tkinter is the standard GUI (Graphical User Interface) toolkit that comes with Python. It provides the necessary components for creating the application's graphical interface, including frames, labels, buttons, entry fields, and list boxes.

III. **Integrated Development Environment (IDE):**

An integrated development environment, such as PyCharm, Visual Studio Code, or IDLE, is used for coding, debugging, and testing the Python scripts that constitute the Library Management System App.

IV. **MySQL Connector:**

MySQL Connector is a Python driver for MySQL databases. It enables Python programs to communicate with MySQL servers.

The script uses MySQL Connector to connect to a MySQL database. It performs operations like inserting, updating, and retrieving tasks from the MySQL database.

V. **MySQL Database:**

MySQL is a popular open-source relational database management system (RDBMS).

The script connects to a MySQL database named " Library Management System " and interacts with a table named "Task." It performs operations like inserting, updating, and retrieving tasks from this database.

VI. **GitHub (Optional):**

GitHub or a similar platform can be utilized for hosting the project's source code, enabling collaborative development and version control. It also facilitates easier sharing and distribution of the application.

Chapter 2

2. Features & Functionality

Graphical User Interface (GUI):

The application provides a graphical interface using Tkinter, making it user-friendly. Users interact with the application through buttons, an entry widget, and a listbox to manage their Library Management System.

1. Book Class:

Represents a book in the library.

Attributes:

title: Title of the book.

author: Author of the book.

available_copies: Number of available copies of the book.

2. Patron Class:

Represents a library patron.

Attributes:

name: Name of the patron.

3. Library Class:

Manages the collection of books, patrons, and transactions.

Methods:

add_book(book): Adds a book to the library.

add_patron(patron): Adds a patron to the library.

borrow_book(patron, book): Handles the process of borrowing a book.

return_book(patron, book): Handles the process of returning a book.

display_books(): Displays the list of books in the library.

display_transactions(): Displays the transaction history.

4. Library Management App Class:

Implements the Tkinter GUI application.

Widgets:

Entry fields for book title, author, available copies, patron name.

Buttons for adding a book, borrowing a book, and returning a book.

5. User Interface:

The Tkinter GUI provides a user-friendly interface for interacting with the library

management system.

Users can add books to the library by providing title, author, and available copies.

Patrons can borrow and return books by entering their names and the book titles.

Retrieve from Database:

Tasks are retrieved from the database on application startup.

The application retrieves tasks stored in the MySQL database and populates the in-memory list and the listbox with these tasks when the program starts.

Database Integration:

The application uses a MySQL database to store tasks.

Tasks are inserted, updated, and retrieved from the MySQL database using SQL queries and the MySQL Connector library.

Error Handling:

The application handles errors gracefully.

The script uses try-except blocks to catch exceptions, such as TclError, providing informative error messages to the user.

Exit:

The "Exit" button gracefully closes the application

Improvements and Customization

- **Database Integration:**

  Currently, the system stores data in memory. Consider integrating a database for persistent data storage.

- **Error Handling:**

  Enhance the application by adding error handling mechanisms for cases such as non-existent books or patrons.

- **User Feedback**:

  Implement a more sophisticated feedback mechanism, such as notifications or status messages, to inform users about the success or failure of operations.

- **Security Considerations:**

  Implement user authentication and authorization mechanisms if the system is intended for multi-user access.

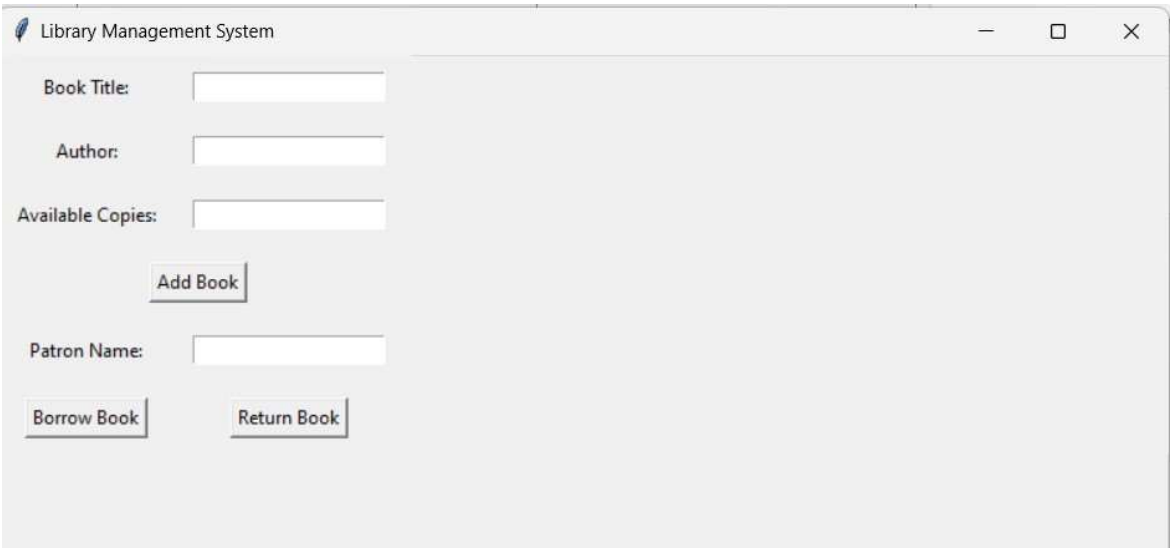
- **UI Enhancements:**

  Improve the user interface with additional features like search functionality, sorting options, and a more visually appealing layout.

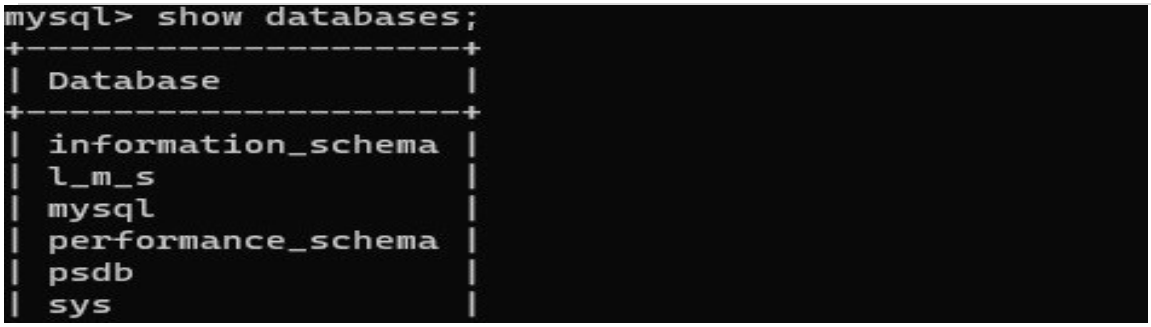
Chapter 3

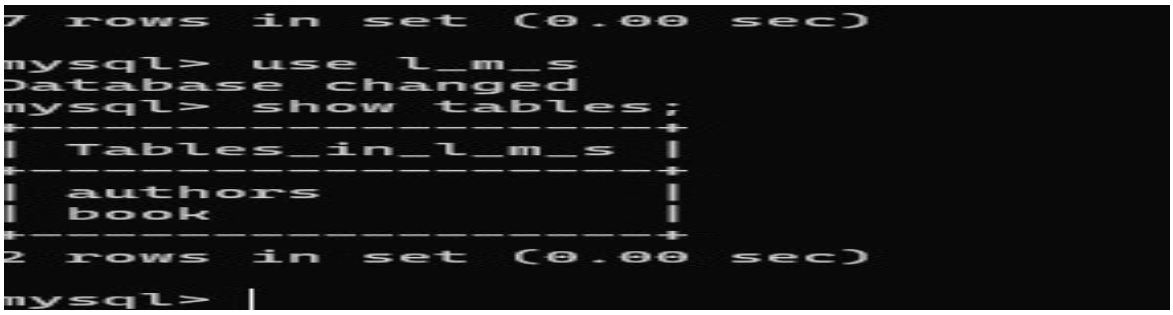
3. Result/Output

App Interface



Database Snapshot





Chapter 4

4. Conclusion and Future Scope

4.1 Conclusion

In conclusion, the Library Management System is a basic yet functional application designed to manage a library's resources through a graphical user interface (GUI) built with Python's Tkinter library. The system allows users to add books, patrons, and perform transactions such as borrowing and returning books. The report provides an overview of the system's components, example usage, and suggestions for improvements and customizations.The system's key components include classes for representing books and patrons, a Library class for managing these entities and transactions, and a Tkinter-based GUI application (LibraryManagementApp class). The user interface features entry fields for book and patron details and buttons for adding books, borrowing, and returning.

Reference

1. Tkinter Documentation: https://docs.python.org/3/library/tkinter.html
2. MySQL Connector Documentation: https://dev.mysql.com/doc/connector-python/en/
3. https://edube.org/learn/pcpp1-4-gui-programming/python-professional-course-series-gui-programming