LIBRARY MANAGEMENT SYSTEM Python Console Application Project Report

1. Cover Page

Project Title: Library Management System

Submitted By: KARANPREET KAUR

REG no. 25BAI10773

Subject: Python Programming

FACULTY : Dr. PREETAM SUMAN

Date of Submission: 23 November 2025

2. Introduction

This is a simple menu-driven Library Management System created using only Python. It allows users to add books, view the book list, borrow, return and search books. All data is permanently saved in a text file (books.txt) using file handling.

3. Problem Statement

In small libraries or personal collections, people use notebooks or memory to track books. This causes problems like:

- Forgetting which book is available or borrowed
- Difficulty in searching a particular book
- Losing records when paper is damaged This project provides an easy digital solution without using any database.

4. Functional Requirements

- Add new book (title + author)
- Display all books with status
- Borrow a book (status → Borrowed)
- Return a book (status → Available)
- Search book by title
- Save and load data automatically

5. Non-Functional Requirements

- Simple and easy-to-use menu
- Works offline
- Fast and lightweight
- No external libraries needed
- Works on Windows, Mac, Linux with Python

6. System Architecture

User → Python Program (library.py) → Text File (books.txt)

7. Design Diagrams (Text-based)

Use Case Diagram Actor: User Use Cases: Add Book, View Books, Borrow Book, Return Book, Search Book, Exit

Workflow Diagram Start → Show Menu → Choose option → Perform action → Save to file → Back to menu → Exit

Sequence Diagram (Add Book) User → Program: Choose 1 → Program: Ask title & author → User: Enter data → Program: Save to books.txt → Program: "Book added"

Class / Component Diagram Only one component: library.py Contains functions: load_books(), save_books(), add_book(), view_books(), etc.

ER Diagram (Storage) Entity: Book Attributes: title (string), author (string), status (Available/Borrowed)

8. Design Decisions & Rationale

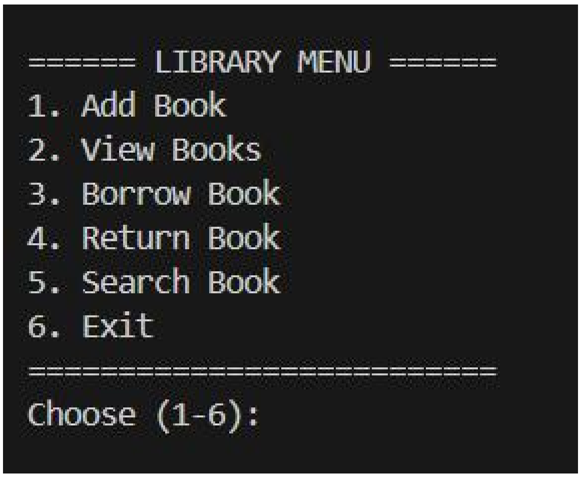
- Used text file instead of database → simple, no installation
- List of dictionaries → easy to handle in Python
- Menu-driven interface → user-friendly for beginners
- Separate functions → clean and reusable code

9. Implementation Details

- Language: Python 3
- Built-in module: os
- Data format: Title|Author|Status (one book per line)
- Main file: library.py
- Storage file: books.txt (auto created)

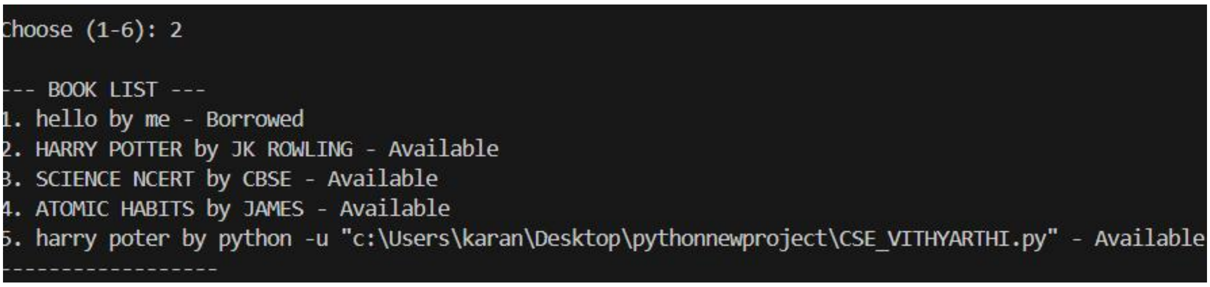
10. Screenshots / Results (Sample Output)

Main Menu



```
====== LIBRARY MENU ======
1. Add Book
2. View Books
3. Borrow Book
4. Return Book
5. Search Book
6. Exit
==========================
Choose (1-6):
```

View Book



```
Choose (1-6): 2

--- BOOK LIST ---
1. hello by me - Borrowed
2. HARRY POTTER by JK ROWLING - Available
3. SCIENCE NCERT by CBSE - Available
4. ATOMIC HABITS by JAMES - Available
5. harry poter by python -u "c:\Users\karan\Desktop\pythonnewproject\CSE_VITHYARTHI.py" - Available
-----------------
```

11. Testing Approach

- Tested all 6 menu options
- Tested with empty file
- Tested invalid inputs (letters instead of numbers)
- Tested borrowing already borrowed book
- All cases passed successfully

12. Challenges Faced

- File not existing on first run → solved using os.path.exists()
- Wrong user input → added try-except
- Data loss on program close → fixed with save_books() function

13. Learnings & Key Takeaways

- File handling (read/write)
- Working with lists and dictionaries
- Menu-driven programming
- Functions and code organization
- Error handling using try-except

14. Future Enhancements

- Add member name who borrowed the book
- Add due date and fine calculation
- Delete book option
- Create GUI using Tkinter
- Password protection for admin

15. References

- Python official documentation (python.org)
- W3Schools Python Tutorial
- GeeksforGeeks – File Handling in Python
- Class notes and self-practice

Done!