

MINI PROJECT REPORT ON LIBRARY BOOK MANAGEMENT SYSTEM

AMRITA VISHWA VIDYAPEETHAM,

NAGERCOIL

SCHOOL OF COMPUTING

23CSE101-

COMPUTATIONAL PROBLEM SOLVING

SUBMITTED BY :

NAME : S.MENUKA

ROLL NO : NC.SC.U4CSE25151

SEMESTER : 1

COURSE : B. TECH CSE-B

Index

Sl.No.	Chapter Name	Page No.
1.	Introduction	3
2.	Problem Statement	4
3.	Objectives	5
4.	Python Libraries used in the project	6
5.	Modules of project	7
6.	Code	9
7.	Output Screenshots	12
8.	Application of the project	14
9.	Limitations of the project	16
10.	Bibliography	18
11.	GitHub link of the project	18

INTRODUCTION

-A library plays a crucial role in any educational institution. Managing books manually using registers is time-consuming and often leads errors.

- The Library Book Management System helps store book details, manage availability, issue/return books, and search for books easily.

- This program introduces Python basics such as lists, dictionaries, conditional statements, loops, and file handling, demonstrating how Python can solve everyday problems.

PROBLEM STATEMENT

-Libraries often depend on manual registers to record book details, issue status, and availability.

-This method has several drawbacks:

****Difficult to update or retrieve book information***

****High chance of human errors***

****Time-consuming***

****No proper record of issued books***

****Hard to search books manually***

- Hence, a simple Python-based automated system is required to store books, check availability, issue/return books, and save records for future use.

OBJECTIVE

- To automate the process of recording and managing library books***
- To reduce manual errors in book entry***
- To check book availability easily***
- To store book data permanently in text files***
- To help beginners learn Python through a real-world application***
- To create a simple, user-friendly mini project***

PYTHON LIBRARIES USED IN THIS PROJECT

This project uses only Python's built-in libraries:

-I/O Functions

input()

print()

-File Handling

open()

.write()

-Data Structures

List

Dictionary

-String Handling

.strip()

.lower()

MODULES OF PROJECT

1. Book Data Module

Stores all book names and availability

Acts as the main database

2. Add Book Module

Allows the librarian to add new books

Updates dictionary dynamically

3. Issue Book Module

Marks a book as “Issued”

Checks if the book is already issued

4. Return Book Module

Changes the status of the book to “Available”

5. Search Book Module

Helps locate a book by name

Shows current status

6. File Handling Module

Saves all book details to library_records.txt

Ensures permanent storage

7. Main Menu Module

Controls all operations

CODE

```
books = {"Python Basics": "Available", "Data Structures": "Available", "AI  
Fundamentals": "Available"}
```

```
def add_book():  
    name = input("Enter book name to add: ").strip()  
    if name in books:  
        print("Book already exists!")  
    else:  
        books[name] = "Available"  
        print("Book added successfully!")
```

```
def issue_book():  
    name = input("Enter book name to issue: ").strip()  
    if name in books:  
        if books[name] == "Available":  
            books[name] = "Issued"  
            print("Book issued successfully!")  
        else:  
            print("Book is already issued.")  
    else:  
        print("Book not found.")
```

```
def return_book():  
    name = input("Enter book name to return: ").strip()  
    if name in books:  
        books[name] = "Available"  
        print("Book returned successfully!")  
    else:  
        print("Book not found.")
```

```
def search_book():  
    name = input("Enter book name to search: ").strip()  
    if name in books:  
        print(f"{name} - {books[name]}")  
    else:  
        print("Book not found.")
```

```
def save_to_file():
```



```
with open("library_records.txt", "w") as file:
    file.write("Library Book Records\n")
    file.write("-" * 30 + "\n")
    for book, status in books.items():
        file.write(f"{book}: {status}\n")
    print("Data saved to file successfully.")
```

```
while True:
    print("\n==== Library Book Management System =====")
    print("1. Add Book")
    print("2. Issue Book")
    print("3. Return Book")
    print("4. Search Book")
    print("5. Save Records")
    print("6. Exit")
```

```
choice = input("Enter your choice (1-6): ")
```

```
if choice == '1':
    add_book()
elif choice == '2':
    issue_book()
elif choice == '3':
    return_book()
elif choice == '4':
    search_book()
elif choice == '5':
    save_to_file()
elif choice == '6':
    print("Exiting Program...")
    break
else:
    print("Invalid choice! Please try again.")
```

OUTPUT SCREENSHOTS

```
===== Library Book Management System =====
```

1. Add Book
2. Issue Book
3. Return Book
4. Search Book
5. Save Records
6. Exit

```
Enter your choice (1-6): 1
```

```
Enter book name to add: Machine Learning Basics
```

```
Book added successfully!
```

```
===== Library Book Management System =====
```

1. Add Book
2. Issue Book
3. Return Book
4. Search Book
5. Save Records
6. Exit

```
Enter your choice (1-6): 2
```

```
Enter book name to issue: Python Basics
```

```
Book issued successfully!
```

```
===== Library Book Management System =====
```

1. Add Book
2. Issue Book
3. Return Book
4. Search Book
5. Save Records
6. Exit

```
Enter your choice (1-6): 3
```

```
Enter book name to return: Python Basics
```

```
Book returned successfully!
```

```
===== Library Book Management System =====
```

1. Add Book
2. Issue Book
3. Return Book
4. Search Book
5. Save Records
6. Exit

```
Enter your choice (1-6): 4
```

```
Enter book name to search: Data Structure
```

===== Library Book Management System =====

1. Add Book
2. Issue Book
3. Return Book
4. Search Book
5. Save Records
6. Exit

Enter your choice (1-6): 5

Data saved to file successfully.

===== Library Book Management System =====

1. Add Book
2. Issue Book
3. Return Book
4. Search Book
5. Save Records
6. Exit

Enter your choice (1-6): 6

Exiting Program...

APPLICATIONS OF THE PROJECT

- The Library Book Management System has several practical applications in different environments.***
- It can be used in schools and colleges to maintain basic book records and help librarians manage daily operations.***
- Small private libraries and local reading clubs can also use this system to organize their collections without needing a complex setup.***
- It is useful for book rental shops to keep track of the books available for rent and their details.***
- Individuals can use it for personal book tracking, helping them maintain records of the books they own or lend to others.***
- With further development, the project can be extended into a mobile or web-based application, making it easier to access and manage books from anywhere.***

LIMITATIONS OF THE PROJECT

- Although the system is functional, it has some limitations.***
- Currently, it has no database support, so data is not stored permanently.***
- It also cannot track issue or return history, making it difficult to monitor past transactions.***
- The lack of a Graphical User Interface (GUI) makes it less user-friendly, especially for beginners.***
- Books cannot be automatically categorized, so all organization work must be done manually.***
- There are no login or security features, which means unauthorized users can access or modify records.***
- The system supports only a single user at a time, limiting its use in busy environments.***

BIBLIOGRAPHY

- [***https://docs.python.org***](https://docs.python.org)
- [***https://www.w3schools.com/python***](https://www.w3schools.com/python)