BUDGE BUDGE INSTITUTE OF TECHNOLOGY

PROJECT REPORT ON

"LIBRARY MANAGEMENT SYSTEM"

CERTIFICATE

This is to certify that **MD SHAHWAZ** has successfully completed the project under the guidance of our HOD"

Mr Siddartha Pradhan" our lecture" **Mr Subsankar Mukherjee.**

CONTENT

INTRODUCTION

The objective of this project is to develop a simple yet efficient Library Management System using Python.This system aims to automate the standard procedures of a traditional library, such as book entry, issuing, returning, and display.It serves as a foundation for more advanced systems that may include user authentication, database integration, and graphical user interfaces.

Software Requirements

Python 3.x (for development and execution)

A text editor or IDE like Visual Studio Code, PyCharm, or Sublime Text

Command Line Interface (CLI) for running the script

Optional: Git for version control, and a virtual environment for managing dependencies

Modules Used

input(): Used for capturing user input

list: To maintain the collection of books

dictionary: For storing issued book-user relationships

functions: For encapsulating repetitive tasks

class: To represent the Library as an object-oriented entity

System Description

The Library Management System is a console-based Python application that handles basic operations such as:

Displaying all available books in the library

Adding new books to the collection

Issuing books to a user, and tracking which user has which book

Returning previously issued books back to the library

When the application runs, it presents a simple menu that the user interacts with via number-based choices. Each feature is encapsulated within methods of the Library class, promoting clean code and reuse.

Advantages

Simple and intuitive interface

No external dependencies required, making it highly portable

Easy to enhance with new features due to modular design

Provides a solid foundation for transitioning into more advanced systems

Useful for small-scale libraries in schools or departments.

Limitations

No Graphical User Interface (GUI), which may limit ease of use for non-technical users.

Data is not persistent once the program ends, all book records and issued information are lost.

Data is not persistent once the program ends, all book records and issued Information are lost

Multi-user access is not supported

No user authentication, meaning no distinction between admin and regular users.

Future Enhancements

This system can be significantly improved with the following enhancements:

Database Integration: Use SQLite or MySQL for persistent storage of books and users

Graphical Interface: Create a GUI using Tkinter, PyQt, or build a web app using Flask/Django

User Authentication: Add login/registration for admins and users with role-based access

Due Date Management: Include issue dates, due dates, and fine calculations for overdue returns

Book Search Feature: Allow users to search for books by title, author, or category

Reporting System: Generate reports of books Issued, returned, and overdue

Conclusion

In conclusion, the Library Management System project effectively demonstrates how Python can be used to build a simple, functional system that automates library operations. It encourages best practices like object-oriented programming and modular code design.

Although the current system is basic, it serves as a solid starting point for more advanced, full-featured applications.

With added features and integration, it can evolve into a complete library management solution for schools, colleges, or small organizations.

CODING

labrary_project.py > Library > __init__

```python
class Library:
    def __init__(self, book_list):
        self.books = book_list
        self.issued_books = {}

    def display_books(self):
        print("\nAvailable Books:")
        for book in self.books:
            print(f"- {book}")

    def add_book(self, book):
        self.books.append(book)
        print(f'"{book}" has been added to the library.')

    def issue_book(self, book, user):
        if book in self.books:
            self.issued_books[book] = user
            self.books.remove(book)
            print(f'"{book}" has been issued to {user}.')
        else:
            print(f'Sorry, "{book}" is not available.')

    def return_book(self, book):
        if book in self.issued_books:
            self.books.append(book)
            del self.issued_books[book]
            print(f'"{book}" has been returned.')
        else:
            print(f'"{book}" was not issued.')

```

CODING

```
def main():
    lib = Library(["C Programming", "DBMS", "Python Basics", "Data Structures"])
    while True:
        print("\n-- Library Menu --")
        print("1. Display Book")
        print("2. Add Book")
        print("3. Issue Book")
        print("4. Return Book")
        print("5. Exit")
        choice = input("Enter your choice: ")

        if choice == '1':
            lib.display_books()
        elif choice == '2':
            book = input("Enter the book name to add: ")
            lib.add_book(book)
        elif choice == '3':
            book = input("Enter the book name to issue: ")
            user = input("Enter your name: ")
            lib.issue_book(book, user)
        elif choice == '4':
            book = input("Enter the book name to return: ")
            lib.return_book(book)
        elif choice == '5':
            print("Thank you for using the Library Management system!")
            break
        else:
            print("Invalid choice. Try again.")

if __name__ == "__main__":
    main()
```

OUTPUT

```
-- Library Menu --
1. Display Book
2. Add Book
3. Issue Book
4. Return Book
5. Exit
Enter your choice: 1

Available Books:
- C Programming
- DBMS
- Python Basics
- Data Structures

-- Library Menu --
1. Display Book
2. Add Book
3. Issue Book
4. Return Book
5. Exit
Enter your choice: 2
Enter the book name to add: COMPUTER NETWORK
"COMPUTER NETWORK" has been added to the library.
```

OUTPUT

```
-- Library Menu --
1. Display Book
2. Add Book
3. Issue Book
4. Return Book
5. Exit
Enter your choice: 3
Enter the book name to issue: C Programming
Enter your name: SHAHWAZ
"C Programming" has been issued to SHAHWAZ.

-- Library Menu --
1. Display Book
2. Add Book
3. Issue Book
4. Return Book
5. Exit
Enter your choice: 4
Enter the book name to returned: C Programming
"C Programming" has been returned.

-- Library Menu --
1. Display Book
2. Add Book
3. Issue Book
4. Return Book
5. Exit
Enter your choice: 5
Thank you for using the Library Management system!
```

BIBLIOGRAPHY

Python Official Documentation

Python Software Foundation. (n.d.). The Python Language Reference. Retrieved from: **https://docs.python.org/3/**

W3Schools Python Tutorial

W3Schools. (n.d.). Python Tutorial.

Retrieved from: https://www.w3schools.com/python/

GeeksforGeeks - Python Programming

GeeksforGeeks. (n.d.). Python Programming Language. Retrieved from: https://www.geeksforgeeks.org/python-programming-language/

Real Python : Real Python. (n.d.). Learn Python Programming. Retrieved from: https://realpython.com/

Tutorials Point-Library Management System Project

TutorialsPoint. (n.d.). Library Management System in Python. Retrieved from:

https://www.tutorialspoint.com/python_projects_library_mana gement system.htm.

Stack Overflow : Stack Overflow. (n.d.). Community discussions and problem-solving related to Python and object-oriented design.

Retrieved from: https://stackoverflow.com/