University of Barishal

EDGE: BU-CSE Digital Skills Training Program

Project Report on

Library Management System

Course: Basic Programming with Python (BPP)

Presented by

Md. Nazmul Islam

Roll: 03-008-26

Submitted to

Md. Rashid Al Asif

Assistant Professor,

Department of CSE,

University of Barishal

Date: 16 February, 2025

Table of Contents

Abstract

This project is a simple Library Management System developed using Python. The system allows users to add, search, edit, view, and delete book records. The purpose of this project is to create a basic system for managing library books efficiently. The report discusses its functionalities, challenges faced, and potential improvements. The system is designed with a text-based interface and serves as an introductory project for learning Python programming concepts.

Introduction

Libraries require an organized system to manage books efficiently. This Library Management System (LMS) allows users to add, search, edit, view, and delete books, providing a structured way to handle library records. The project was created as part of the EDGE digital skills program at Barishal University.

The LMS aims to help library administrators maintain a record of books in a structured manner. Instead of manually recording book details, this system allows automation of adding and retrieving books. The system is developed in Python, offering a lightweight and user-friendly interface. While currently limited to basic operations, it can be expanded with additional functionalities such as borrowing and returning books.

System Requirements

This Library Management System is a simple Python-based program requiring minimal system resources. The following components are needed:

- Programming Language : Python
- Development Environment : PyCharm Community Edition
- Libraries Used : Standard Python libraries (no external dependencies)
- Operating System : Windows/Linux/Mac
- Database : None (Data stored in memory)

Since the system is text-based, it does not require a graphical user interface. It is lightweight and can run on any system with Python installed.

Features and Functionalities

The Library Management System provides the following functionalities:

1. Add Books
   - Allows users to input book details such as title, author, category, and year of publication.
2. Search Books
   - Enables users to search for books using keywords in the title or author's name. The system retrieves relevant books from the stored records.

3. Edit Books
  - Users can modify existing book details if corrections or updates are needed.
4. View Books
  - Displays all books currently stored in the system in an organized manner.
5. Delete Books
  - Users can remove a book from the system by specifying its details.

These features ensure basic book management functions are available for library users.

Implementation Details

The system follows a simple menu-driven approach, allowing users to navigate through different options. The main components include:

1. Main Menu : Provides options such as "Add Book", "Search Book", "Edit Book", "View Books", and "Delete Book".
2. Data Storage : Books are stored in a list within the program. Since there is no database, book data is lost once the program exits.
3. Function Calls : Each menu option triggers a function, executing the selected task. The functions handle user inputs and perform relevant actions.
4. User Interaction : The system continuously prompts users for input and validates their responses. If invalid inputs are detected, error messages are displayed.

Challenges and Solutions

As a beginner in Python programming, I encountered multiple challenges during this project. Some of the key difficulties and their solutions are outlined below:

1. Logic Implementation
  - Challenge: Writing logic for adding, searching, and editing books.
  - Solution: Breaking the problem into smaller steps and implementing each function separately before integrating them.

2. Error Handling
  - Challenge: Ensuring the program does not crash due to invalid user input.
  - Solution: Using try-except blocks and input validation techniques to handle

errors smoothly.

3. Data Persistence
   - Challenge: Data was stored only in memory, leading to loss when the program was closed.
   - Solution: Future plans include implementing file handling or database integration.

Future Improvements

While the system currently meets basic requirements, several enhancements can improve its functionality:

- Unique ID Generation:
  Assigning a unique identifier to each book for easier retrieval.

- Book Borrowing and Returning:
  Introducing a lending system where users can borrow books and return them later.

- Database Integration:
  Using a database like SQLite to store book records permanently.

- Graphical User Interface (GUI):
  Developing a user-friendly interface using Tkinter or PyQt for better usability.

Conclusion

This Library Management System provided hands-on experience in Python programming. It introduced core programming concepts, including data handling, user input validation, and function structuring.

Although the system is currently simple, it serves as a foundation for further enhancements. Implementing database support, unique book identification, and borrowing features can significantly improve its functionality.

Through this project, I improved my coding skills and gained a better understanding

of how to develop structured programs in Python.

References

1. Python Documentation - https://docs.python.org/3/
2. Online Python Tutorials - https://realpython.com/
3. Tkinter GUI Programming - https://docs.python.org/3/library/tkinter.html