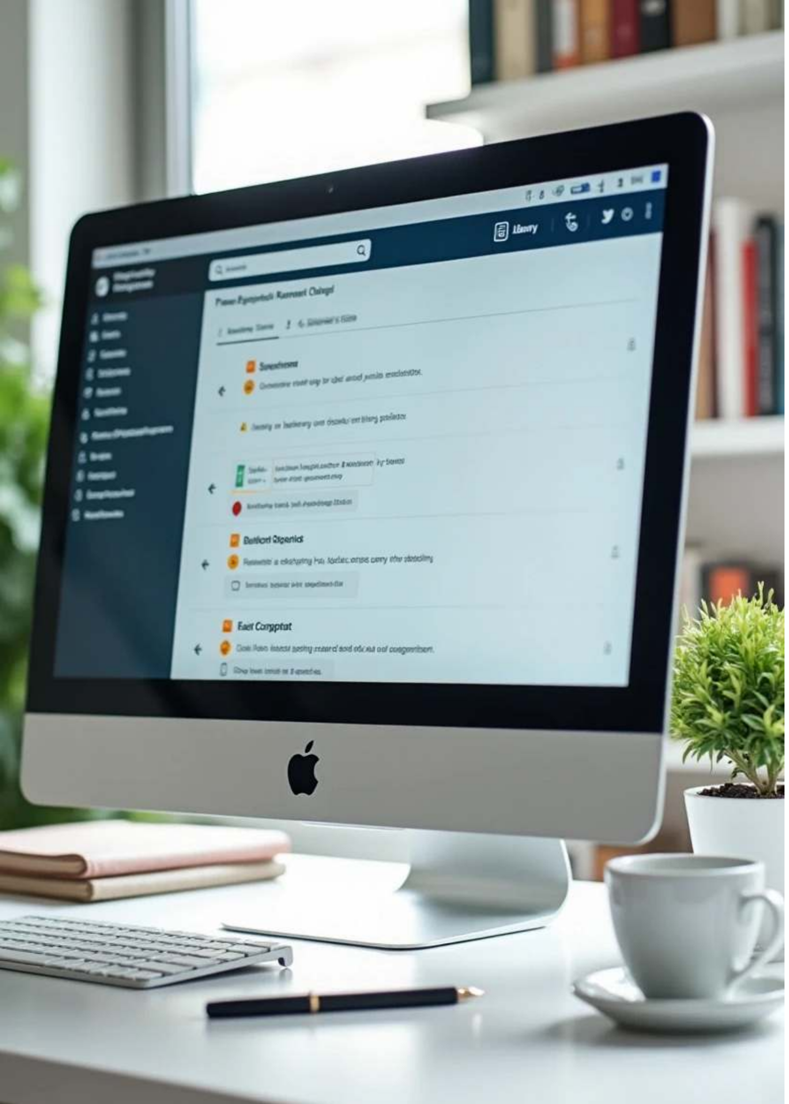


Library Management System Overview

This presentation introduces a simple text-based Library Management System developed in Python. It covers the system's objectives, the necessary software requirements, the steps for implementation, and potential enhancements for future improvements.

Library Management System Overview

1

Goal

Create a straightforward Library Management System in Python to manage books.

2

Purpose

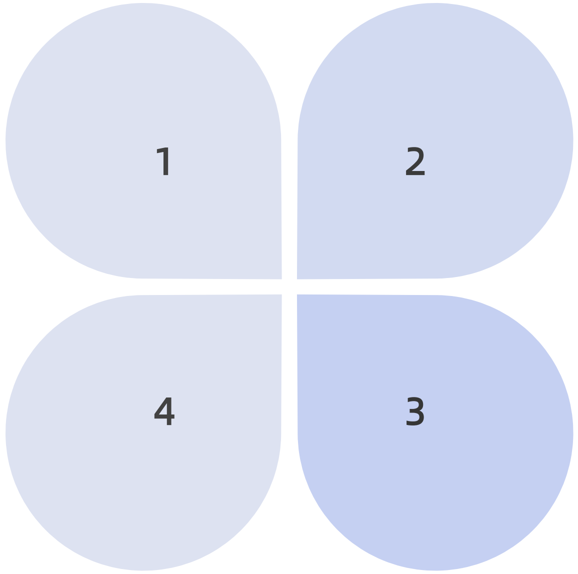
The system will allow users to add, view, issue, return, and search for books.

3

File Name

The project's main file will be named library_management.py.

Software Requirements



Python Version

Ensure you have Python 3.8 or newer installed on your system.

Text Editor/IDE

Use any compatible text editor or IDE like VS Code, PyCharm, or Sublime Text.

Operating System

Compatible with Windows, macOS, or Linux operating systems.

Terminal/Command Prompt

Required to run the program interactively and view outcomes.

Implementation Steps for Library Management System

1. **Install Python**

   Add Python to system PATH and verify installation.

2. **Create Project Folder**

   Set up a folder for your project.

3. **Save Code File**

   Save the code as library_management.py in the project folder.

4. **Navigate to Project Folder**

   Use Command Prompt or Terminal to access the folder.

5. **Run the Program**

   Execute the program to see the main menu.

Menu Options

Option 1: Add Book

Allows users to insert a new book by providing its ID, title, and author.

Option 2: Display All Books

Displays all added books, both available and issued.

Option 3: Issue Book

Users can mark a book as issued by entering its ID.

Option 4: Return Book

Facilitates the return of previously issued books by ID.

Option 5: Search Book

Users can search for books by title or author using keywords.

Option 6: Exit

Safely closes the program.

Testing Example

This testing example illustrates the functionality and usability of the program in managing books. Users can add, display, issue, return, and search for books efficiently.

1. **Sample Run**

   The program begins with a menu for users to select actions on books.

2. **Adding a Book**

   Inputting details like Book ID, Title, and Author.

3. **Displaying Books**

   Use Option 2 to see a complete list of books.

4. **Issuing and Returning**

   Options 3 and 4 to manage book status.

5. **Searching**

   Use Option 5 to find books based on specific keywords.

Common Errors & Fixes

SyntaxError

Replace smart quotes with straight quotes for code accuracy.

EOFError

Ensure valid inputs are provided without pressing Enter unnecessarily.

Program Closes Instantly

Run the program from the terminal for better handling.

Unicode Emojis Issues

Remove emojis if terminal lacks support, to prevent issues.

Optional Enhancements

Saving Books Permanently

Implement data persistence using modules like pickle or json for file storage.

Login System

Enhancing security with an admin login feature could be beneficial.

Database Integration

Utilize sqlite3 for efficient long-term data management and retrieval.

Graphical User Interface (GUI)

Transition to a GUI using libraries such as tkinter or PyQt for improved user experience.