Week 5-Task 1

PROJECT
LIBRARY MANAGEMENT STSTEM

PREPARED BY

Sana Fatima

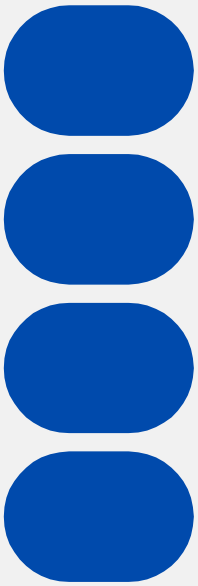
sana.faati@gmail.com

PREPARED TO

A&D Tech

ad.techinnov25@gmail.com

TABLE OF CONTENTS



Problem Statement

Implementing the Program

- Create a project and named it as LibraryManagementSystem.
- Create three different classes to make code workable(Book class, Library class and User class) where user class is optional.
- Implement functionality to:
  1. Add new books to the library.
  2. Check out books to users.

3. Return books and update their status.
4. Search for books by title, author, or ISBN.
5. Use exception handling to manage common issues, such as trying to borrow a
6. book that is already checked out.

SOLUTION OVER VIEW

The solution overview for this program is:

The code is a C++ program that creates a Library Management System. The program includes two main classes:

`Library` class and `Book` class (these classes contain the main functionality).

Whereas ,User class is optional but I tried best to implement this class and make it fully functional, this code also implement saving and retrieving library data from a file or database (simple file handling).

The main.cpp file contains the primary interface for interacting with the user, presenting a menu and handling user input. The Library and User classes manage the core functionalities like adding books, checking out/returning books, and saving/loading data.

CHALLENGES AND RESOLUTIONS

.

Challenge: Handling complexity

- **Problem:** As the project contain multiple classes the main problem is to handle the complexity.
- **Resolution:** By reviewing code over and over.

Challenge: Saving and Loading Data to/from Files

- **Problem:** The project requires saving and loading both library and user data to/from text files. Ensuring that the data is saved and loaded correctly, especially when dealing with objects, requires careful implementation of file I/O operations.
- **Resolution:** The WriteToFile and ReadBookFromFile methods in the Library class, along with WriteUserToFile and ReadUserFromFile in the User class, handle file operations. These methods ensure that all relevant data, such as the list of books and their statuses, as well as user information, are correctly saved and loaded.

Challenge: Handling Input Validation and User Experience

- **Problem:** The program must handle user input gracefully, ensuring that invalid input does not crash the system or cause unexpected behavior.
- **Resolution:** Input validation is implemented using loops and checks, such as the while (!(cin >> choice) || choice < 1 || choice > 8) loop, which ensures that the user only enters valid menu choices.

INSTRUCTIONS

The `main` function displays a menu with three options:

- **Compile the Code:** Use a C++ compiler to compile the program files(Library.h, User.h, Book.h and main.cpp.
- **Run the Executable:** After compilation, run the program.
- **Interact with the Menu:** The program will present a menu with options for managing the library.
- Enter the corresponding number for the desired action.
- **File Operations:** The program saves library data to output.txt and user data to user.txt.
- These files are automatically created and managed by the program.
- **Exit the Program:** Choose the option 8. Exit from the menu to safely exit the program