Library Management System (LMS)

Overview

The Library Management System (LMS) is a console-based application developed in C++ to manage the operations of a library. It follows Object-Oriented Programming (OOP) principles such as abstraction, inheritance, polymorphism, and encapsulation to facilitate book borrowing, returns, reservations, fine calculations, and user management.

Features

- **User Authentication:** Users can log in with an ID and password.
- **Role-based Access Control:** Three types of users:
  - **Students**: Can borrow up to 3 books, must return within 15 days, incur fines for late returns.
  - **Faculty**: Can borrow up to 5 books, must return within 30 days, no fines but overdue books restrict further borrowing.
  - **Librarian**: Manages users and books, cannot borrow books.
- **Book Management:** Books have attributes such as title, author, publisher, ISBN, and availability status.
- **Borrowing and Returning Books:** Users can check out and return books, with automated fine calculations for students.
- **Book Reservations:** If a book is currently issued, users can reserve it for future borrowing.
- **Fine Calculation:** Students incur fines of 10 rupees per day for overdue books.
- **Data Persistence:** User and book data are stored in CSV files to maintain state between program executions.

Class Structure

1. `User` (Base Class)

Defines common functionalities for all users.

- `see_history(id)`: Displays borrowing history.
- `reserve_book(id, bookname)`: Reserves a book.
- `unreserve_book(id, bookname)`: Cancels a reservation.
- `issue_book(id, bookname, type_user, V)`: Issues a book.

- `return_book(id, isbncode)`: Returns a borrowed book and calculates fines if applicable.
- `calc_fine(id, type_user)`: Computes overdue fines.
- `clear_fine_amount(id, type_user)`: Resets fine amount.
- `logout()`: Logs the user out.

2. `Student` (Derived from `User`)

- `display_student_menu(id)`: Provides options for students to borrow, return, and view books.

3. `Faculty` (Derived from `User`)

- `display_facu_menu(id)`: Provides options for faculty members.

4. `Librarian` (Derived from `User`)

- `display_lib_menu(id)`: Provides options to manage books and users.
- `see_all_users(id)`: Displays all registered users.
- `see_all_books(id)`: Displays all books in the library.
- `add_user(id)`: Adds a new user.
- `update_user(id)`: Updates user details.
- `delete_user(id)`: Removes a user from the system.

5. `Book`

Represents a book in the library.

- `Book_request(id, bookname, type_user)`: Allows a user to borrow a book.
- `Show_duedate(isbn_no)`: Displays due date for a borrowed book.

6. `Library`

Handles library-related operations.

- `display_menu()`: Displays the main menu.
- `login()`: Authenticates users.
- `add_book(id)`: Adds a book to the system.
- `update_book(id)`: Modifies book details.
- `delete_book(id)`: Removes a book from the system.

7. `Account`

Manages user account data.

- `Account_add(id)`: Registers a new user.
- `Account_update(id)`: Updates user details.
- `Account_delete(id)`: Removes a user account.
- `Account_clear_history(id)`: Clears a user's borrow history.

Borrowing & Return Rules

- **Students**: Can borrow up to 3 books for 15 days. Overdue books incur a fine of 10 rupees per day.
- **Faculty**: Can borrow up to 5 books for 30 days. No fines, but overdue books prevent further borrowing.
- **Librarians**: Cannot borrow books but manage library resources.
- **Books can only be borrowed if available.**

Fine Payment & Restrictions

- Users with unpaid fines cannot borrow new books.
- Users can clear fines through a menu option.
- Once cleared, borrowing privileges are restored.

File Storage

Data is stored in CSV files:

- `all_users_data.csv`: Stores user details and fines.
- `all_books_data.csv`: Stores book information.
- `issued_books_data.csv`: Tracks currently borrowed books.
- `borrow_history.csv`: Maintains borrowing records.

How to Run the Program

1. Compile the program using a C++ compiler:
   g++ main.cpp -o library
2. Run the executable:
   ./library.exe
3. Follow on-screen prompts to log in and interact with the system.

Conclusion

This Library Management System effectively manages book lending, reservations, and fines while enforcing borrowing rules based on user roles. It demonstrates key OOP principles and ensures data persistence through file handling.