### **Virtual Library Management System (VLMS) Project Overview**

**Objective:** Develop a console-based application that simulates the core functionalities of a library management system, enabling operations such as book browsing, borrowing, and returning for users, alongside administrative controls for managing the library's catalog.

### **Technical Requirements**

1. **Version Control**: Utilize Git for source code management. The project should be hosted on a platform like GitHub to facilitate collaboration and version tracking.
2. **Testing**: Implement unit tests for individual components and integration tests for the system using C++ testing frameworks such as Google Test.
3. **Documentation**: Each team member is responsible for documenting their code and contributing to the project report, which will detail the system's design, implementation strategy, testing protocols, and usage instructions.
4. **Data Structures**:
   * **Book Management**: Employ a Binary Search Tree (BST) for storing and efficiently searching for book information.
   * **User Management**: Use a Hash Map for fast access to user accounts.
   * **Checkout Queue**: Implement a Queue to manage the FIFO borrowing system.

### **Project Components and Responsibilities**

* **User Interface & Navigation**: Design a clear console interface for user interaction, including separate menus for logged-out users, standard users, and administrators.
* **Book Management**: Implement functionalities to add, remove, update, and search for books.
* **User Account Management**: Create a secure login and registration system, handling user data responsibly.
* **Data Persistence**: Ensure the system can save and retrieve library and user data between sessions.

### **Feature Details**

**Pre-Login Menu**

Welcome to the Virtual Library Management System  
------------------------------------------------  
1. Login  
2. Register  
3. Exit

Please select an option:

**User Menu (After Login)**

Welcome [Username], you're logged in as a [Role].  
--------------------------------  
1. Search for Books  
2. Borrow a Book  
3. Return a Book  
4. View Borrowed Books  
5. Update Profile  
6. Logout

***Admin Menu***

Admin Dashboard  
---------------  
1. Add a Book  
2. Remove a Book  
3. Update Book Information  
4. View All Loans  
5. Add/Remove User (Admins)  
6. Logout

### **Deliverables**

1. **Source Code**: Fully documented and managed using Git, including regular commits and meaningful commit messages.
2. **Testing Suite**: A set of automated tests for critical functionalities.
3. **Project Report**: A comprehensive document covering system design, implementation details, testing strategy, user manual, and team member contributions.

### **Implementation Notes**

* **Security Considerations**: While in-depth security practices like password hashing may be beyond this project's scope, emphasize their importance in the documentation.
* **Data Storage**: Use simple file handling (e.g., text or JSON files) for storing user and book data. This simplifies the project's complexity while teaching fundamental data management concepts.
* **Role Management**: Ensure the system distinguishes between standard users and administrators, granting appropriate access and functionalities based on the user's role.

For this Week 4 Assignment, submit a rough Project Plan, that shows the Tasks and assignments.  This should be in a Gantt Chart format.  Use Microsoft Project or [ProjectLibreLinks to an external site.](https://www.projectlibre.com/" \t "_blank) to do so