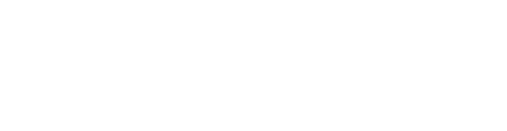
**PROJECT PROPOSAL**

**07/01/2024**



Submitted To: Sir Omer Aftab

Submitted By: Rimsha Shehzadi

Roll no: Fa-2023/BSSE/152

Semester: 4TH

Section: B

**Library Management System**

**Background & Overview**

In today’s fast-paced world, libraries have evolved beyond just physical bookshelves. Manual record-keeping is time-consuming and prone to errors. A **Library Management System (LMS)** offers an automated way to manage users and books, making operations smooth, efficient, and error-free.

This C++ project simulates a **mini-library system**, enabling users to **sign up, log in**, and manage **book-related data** like adding, viewing, deleting, and searching by genre or author.

**Why Build This Project?**

* **Real-world Need**: Libraries require structured data management. Manually managing books and user records is inefficient and outdated.
* **Learning Purpose**: Helps beginners practice:
  + Linked lists (via struct pointers)
  + Basic file/data handling logic (though this one’s in-memory)
  + User authentication logic
  + Menu-driven programming
* **Foundation for Bigger Systems**: This project can grow into a more complex system involving file handling, GUI, or even database integration.

**Objectives and Goals**

| **Objective** | **Description** |
| --- | --- |
| **User Authentication** | Secure sign-up and login system |
| **Book Management** | Add, delete, view all books |
| **Search Functionality** | Search by **genre** or **author** |
| **Memory Management** | Use of dynamic memory and linked lists |
| **Scalability** | Base structure for advanced future improvements |

**Project Explanation**

Imagine you're a **librarian** in a small but busy community library. You’ve got dozens of books and multiple visitors every day, but only paper records to manage everything.

You decide to **build a digital system** in C++ that does the following:

1. **SignUp/Login**:  
   Just like any secure website, new users can register and log in with a password. This ensures that **only authorized users can access** or modify the book records.
2. **AddBooks**:  
   You just received a new book? Great! Add its ID, title, author, and genre with one command, and it’s saved in your system instantly.
3. **ViewBooks**:  
   Any time you want to check your collection, just press a button and the entire list of books appears.
4. **DeleteBooks**:  
   Old book not needed anymore? Delete by entering its ID — it's removed from the list.
5. **SearchFeature**:  
   Want to recommend a horror book to a reader? Or books by a certain author? Search by **genre** or **author** with one command.
6. **LinkedListMagic**:  
   All book records and users are stored using **linked lists**, which means:
   * Dynamic memory allocation
   * No limit on data size
   * Efficient traversal

**💡 Benefits of the Project**

| **Benefit** | **Explanation** |
| --- | --- |
| **Organized Data** | Easily manage books and users |
| **Efficiency** | Reduces manual effort and errors |
| **Secure Access** | Login system protects sensitive data |
| **Good Practice** | Strengthens understanding of core C++ concepts |

**Conclusion**

This Library Management System is more than just a C++ project — it’s a **practical solution** to a common real-life problem, built with **core programming logic** and offering plenty of room for future development.

Whether you're a student, a hobbyist, or preparing for software development, this project gives you a **real taste of structured problem-solving and system design**.