# Admissions - Ghulam Ishaq Khan Institute of Engineering Sciences and Technology - GIKI

**Notebook Application (C++)**

Group Members:

Muhammad Ghazali Khan -2024380

Muhammad Shariq Usman -2024465

Sohaib Bin Tausif -2024595

Instructor:

Dr. Zubair Ahmed.

**Introduction**

In the digital age, when it comes to keeping all personal and professional information under control, there are not too many options. Manual data manipulation tends to be accompanied by errors, repetitions and loss of data.

* Profile creation, viewing, and deletion for individual users.
* Page management to add, edit, and delete notebook pages.
* Categorized expense tracking (Family, Medical, Work, Leisure, and Salary).
* Login history tracking using stack-based storage.
* Calculation of total income, expenses, and net savings.

Many students have noticed that while using the LMS that the current notebook feature on offer is quite lacking and clunky to use. Which is why we choose to develop a better offering to help improve the experience of students interacting with GIKI’s domestically developed LMS system

**Problem Statement:**

To design and implement a Multi-User Notebook Program using C++ that demonstrates the use of fundamental data structures for managing users, profiles, pages, and categorized notes.

**Objectives**

The objective of this project is to design and implement a console-based multi-user notebook management application in C++ that demonstrates the use of data structures such as Linked Lists, Stacks, and Queues for managing user accounts, profiles, pages, and financial categories.

**Tools and Concepts used:**

* Language: C++
* Data Structures: Singly Linked List, Doubly Linked List, Stack
* Programming Concepts: Classes, Pointers, Encapsulation, Loops, and Menu-driven structure.

**System Overview**

The program allows multiple users to create accounts, log in, and manage their own profile, pages, and financial records. It consists of modules for user management, profile handling, page management, category tracking, and login history.

**Data Structures Used**

• Singly Linked List – Used in UserList and CategoryList  
• Doubly Linked List – Used in PageList and ProfileList  
• Stack – Used in LoginHistory to store login order

**Implementation of core data structures and algorithms concepts:**

This project demonstrates multiple data structure concepts effectively:

**Linked Lists:** Used for sequential data management in user accounts, profiles, and notes.

**Stacks:** Used in the login history to maintain a Last-In-First-Out (LIFO) structure.

**Deletion Algorithms:** Implemented for iterating through and managing nodes dynamically.

**Dynamic Memory:** Allocation: Each new user, profile field, or page is created and linked using pointers.

**Switch Case Control and Nested Menus**: Allow efficient user interaction and navigation between modules.

**Key Features Implemented:**

- User Authentication (Signup/Login)  
- Profile Management (Create, View, Delete)  
- Page Management (Add, Delete, Display Pages)  
- Category Management (Family, Medical, Work, Leisure, Salary)  
- Login History Tracking  
- Expense and Savings Calculation

**Challenges Faced:**

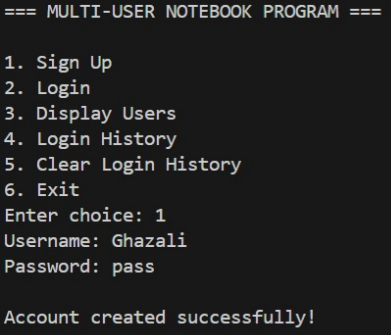
- Managing multiple linked lists and ensuring proper node connections.  
- Handling pointer operations safely to avoid segmentation faults.  
- Ensuring modular and reusable class design.  
- Balancing user interaction with backend data manipulation.

**Solutions:**

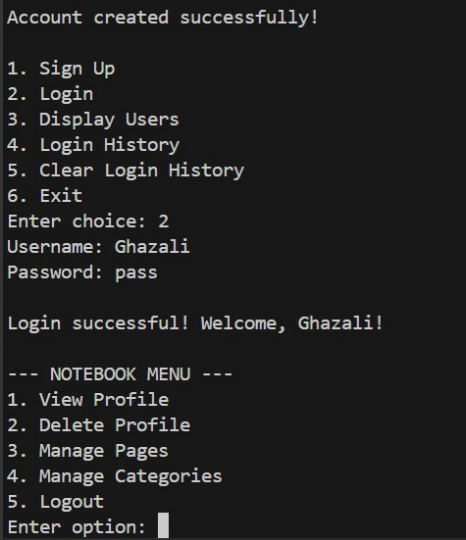
- Used structured class design with clear responsibility segregation.  
- Implemented robust input validation and error handling.  
- Applied dynamic memory deallocation to prevent memory leaks.

**Sample Outputs**

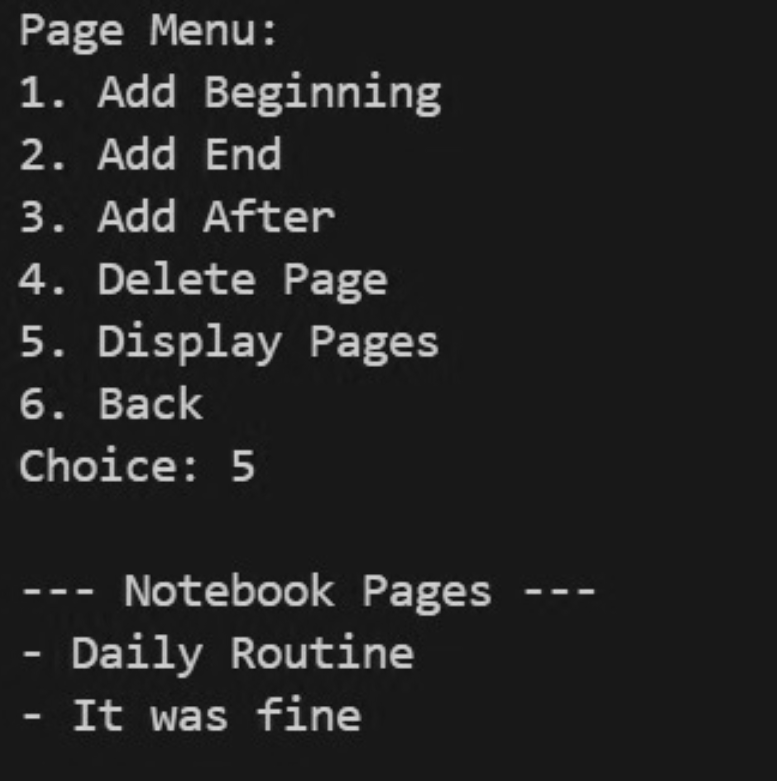
Login/Signup Menu (Choice = 1):



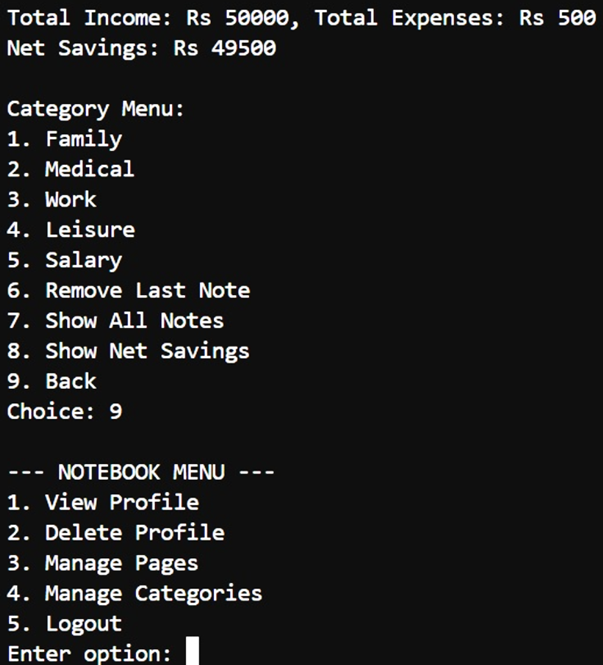
Login/Signup Menu (Choice = 2):



Page Menu:



Category Menu:



**Conclusion**

The Multi-User Notebook Project effectively showcases the practical use of data structures like linked lists and stacks. It demonstrates modular design principles, efficient data handling, and user-oriented interaction through console-based menus.

**Future Improvements**

- Implement file handling to store data permanently.  
- Add password encryption for improved security.  
- Integrate a GUI interface for better user experience.  
- Expand notebook functionality with sorting and searching algorithms.