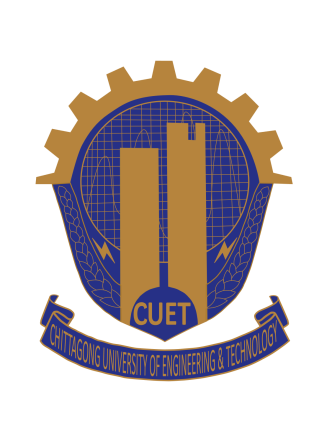
**Chittagong University of Engineering & Technology**



Department of Computer Science & Engineering

**Project Name:**Building a library management system with Shellscript.

Course ID: CSE-336

Course Title: Operating System (Sessional)

|  |  |
| --- | --- |
| Submitted To | Submitted By |
| * Md Shafiul Alam Forhad   Assistant Professor,  Dept. of CSE, CUET   * Hasan Murad   Assistant Professor,  Dept, of CSE, CUET | * Md. Abdulla (2004072) * Towhid Al Faysal (2004073) * Nahin Munkar (2004074) * Shawrin Fouzia (2004075) * Shuvodwip Saha (2004076) |

**Introduction:** In the modern digital age, efficient management of library resources is crucial for academic institutions, public libraries, and private collections. The traditional methods of manually cataloging, tracking, and managing library resources are increasingly being replaced by automated systems that offer higher accuracy, improved user experience, and greater efficiency. This project focuses on the development of a Library Management System (LMS) using Linux shell scripting, aiming to provide a robust, lightweight, and cost-effective solution for managing library operations.

Linux shell scripting is a powerful tool for automating tasks and managing system operations, known for its flexibility, simplicity, and compatibility with various UNIX-based systems. By leveraging the capabilities of shell scripting, this project aims to create a comprehensive LMS that can handle core library functions such as book cataloging, user management and inventory tracking.This report outlines the methodology, implementation, and results of the Library Management System project. The subsequent sections will detail the system design, the scripting techniques employed, the challenges encountered, and the solutions devised. The report concludes with an evaluation of the system's performance and potential areas for future enhancement.

**Objectives:**

* **Develop a User-Friendly Interface**: Design a simple and intuitive command-line interface that allows librarians and users to easily interact with the system.
* **Automate Library Operations**: Implement automation for key library management tasks to reduce manual workload and minimize errors.
* **Ensure Data Integrity and Security**: Develop mechanisms to maintain accurate records and protect sensitive data from unauthorized access.
* **Provide Comprehensive Reporting**: Enable the generation of various reports to assist in library administration and decision-making.

## **Methodology and Approach:**

## Based on the provided excerpts and interactions, the methodology and approach of the "Library Management System" project can be inferred as follows:

## **Methodology:**

## 1. Script-Based Approach: The system is implemented as a shell script, making it lightweight and easily runnable on Unix/Linux systems without the need for compiling or additional runtime environments.

## 2. Interactive User Interface: The system interacts with the user through the command line, prompting for inputs and displaying information. This approach is user-friendly for those comfortable with command-line interfaces.

## 3. Data Storage: The system likely uses a CSV file (`$library\_database\_temp`) as a simple database to store book records. This choice simplifies data manipulation and avoids the complexity of integrating with a database management system.

## 4. Modular Function Design: The system is structured into functions, each handling a specific aspect of library management (e.g., `add\_books`, `find\_books`, `edit\_books`, `remove\_books`, `view\_books`). This modular design improves code readability and maintainability.

## **Approach:**

## 1. Initialization: The script might begin with setting up the environment, such as clearing the terminal screen and initializing variables.

## 2. Main Menu Loop: The core of the system operates in a loop, presenting a menu of actions (add, find, edit, remove, view books) and processing user input until the user decides to exit.

## 3. Input Validation: Before proceeding with actions, the system validates user inputs (e.g., checking for empty book titles) to ensure data integrity.

## 4. Data Manipulation: For operations like adding, editing, or removing books, the system manipulates the CSV file, updating records as necessary.

## 5. Feedback and Confirmation: After each operation, the system provides feedback to the user (e.g., confirmation of added or edited records).

## 6. Clean-up and Exit: Upon exiting, the system performs any necessary clean-up, such as removing temporary files.

## **Results or Findings:**

The development of the Library Management System using Linux shell scripting yielded several noteworthy results. By employing various shell scripting functionalities, we were able to create a robust and efficient system with the following key features and outcomes:

1. **Book Management**:
   * **Add Books**: The system allows users to add new books to the library record efficiently. The implementation ensures that all necessary book details are captured and stored accurately in a structured database file.
   * **View Books**: The system allows users to view the book records and catalogue of the the library.
   * **Find Books**: A search functionality enables users to find books in the database based on various criteria such as title. The search process is swift and returns precise results, enhancing user experience.
   * **Remove Books**: Users can delete books from the library record when they are no longer needed. This feature ensures that the database remains up-to-date and free of redundant entries.
   * **Edit Book**: The system provides an option to edit book details, allowing librarians to update information as needed. This functionality ensures the accuracy and relevance of the database.
2. **User Interface**:
   * The command-line interface (CLI) designed for this system is user-friendly and intuitive. It allows users to navigate through different options seamlessly, making the system accessible to individuals with varying levels of technical expertise.
3. **System Operations**:
   * **Conditional Logic and Control Flow**: The use of conditional statements and control flow mechanisms in the script ensures that the system operates smoothly and handles different scenarios effectively.
   * **Variable Handling**: Efficient management of variables throughout the script helps in maintaining the state and consistency of data.
   * **Functions**: Modular functions implemented within the script promote code reusability and simplify the overall structure of the system.
   * **Error Handling**: Comprehensive error handling mechanisms are in place to manage unexpected inputs and system errors gracefully, thereby enhancing system reliability.
   * **File Operations**: The system utilizes file handling commands to read from and write to the database file, ensuring data persistence and integrity.
4. **System Performance**:
   * The system performs all its functionalities with minimal latency, providing a responsive experience to the users. The efficiency of shell scripting ensures that operations are executed promptly without significant resource overhead.

Overall, the project successfully demonstrates the practicality and effectiveness of using Linux shell scripting to develop a functional Library Management System.

**Conclusion:**

The Library Management System project, developed using Linux shell scripting, achieves its objectives of creating a reliable, efficient, and user-friendly solution for managing library operations. By leveraging the powerful features of shell scripting, the system effectively handles essential tasks such as adding, finding, deleting, and editing books in the library database. The use of conditional logic, control flow, variable handling, functions, error handling, and file operations contributes to the robustness and efficiency of the system.

This project highlights the potential of shell scripting as a viable tool for developing lightweight and cost-effective management systems. The successful implementation of the Library Management System demonstrates the script's ability to automate processes, maintain data integrity, and provide a seamless user experience.

Future enhancements could include adding advanced features such as user authentication, detailed logging, and integration with other library systems or databases. Additionally, the system could be extended to include a graphical user interface (GUI) to cater to users who prefer a visual interaction model.

In conclusion, the Library Management System project serves as a testament to the versatility and capability of Linux shell scripting in developing practical and efficient software solutions for real-world applications.