

“Library Management System based on Shell Scripting”

Subject – Operating System

SUBMITTED BY

Name of Student	Roll No.
Riya Kharade	24101C2001
Shivani Choudhari	24101C2002
Kirtana Singh	23101C0065

Supervisor

Mr. Mahesh Khandke



Vidyalankar Institute of Technology
(AY 2024-25)

CERTIFICATE

This is to certify that the Mini Project entitled **“Library Management System”** is a Bonafide work of **Riya Kharade (24101C2001), Shivani Choudhari (24101C2002), Kirtana Singh (23101C0065).**

Mr. Mahesh Khandke

Supervisor

Contents

Abstract	ii
Acknowledgments	iii
List of Abbreviations	iv
List of Figures	v
List of Tables	vi
List of Symbols	vii
1 Introduction	1
1.1 Problem Statement & Objectives	
2 Literature Survey	2
2.1 Survey of Existing/Similar System	
3 Proposed System	3
3.1 Introduction	
3.2 Architecture/ Framework	
3.3 Algorithm and Process Design	
3.4 Details of Hardware & Software	
3.4 Experiment and Results	
3.5 Conclusion and Future work	
4 References	15

Abstract

In today's digital world, automation plays a key role in improving the efficiency and accuracy of daily operations. One such area where automation can be highly beneficial is library management. Traditionally, managing books in a library involves keeping manual records, which can be time-consuming and error prone. To overcome these challenges, we have developed a Library Management System using Bash scripting. This system helps in handling books easily by supporting key operations such as insertion, deletion, updating, searching, displaying, and issuing of books—all through a command-line interface.

This project uses a simple text file (`library_data.txt`) to store all the book records. Each line in this file represents one book and contains details like the Book ID, Book Name, Author Name, Publisher, Publisher Year, and Number of Copies, separated by colons (:). This format makes it easy to store and retrieve data using basic Linux commands. The Book ID is used as a unique identifier to avoid duplication and ensure accurate updates or deletions. This design supports a lightweight, easy-to-maintain solution, especially useful in small libraries, personal collections, or educational labs.

The user interacts with the system through a menu-driven interface. When the user chooses to insert a book, the system checks if the Book ID already exists using the `grep` command. If not, the new book is added to the file. The update function allows users to modify one or more fields of an existing book entry without overwriting the entire record. Similarly, the delete function removes a book from the database, while the search feature displays book details based on Book ID input. All these operations are handled efficiently using Bash utilities like `grep`, `cut`, and `sed`.

Another important part of the system is the book issuance module. When a book is issued, the number of available copies is reduced by one. This prevents the book from being issued again when no copies are left. It ensures accurate tracking of book stock. The system also includes a feature to display all books in a neatly formatted table. Additionally, the user can view the list of books sorted alphabetically by book name using the `sort` command. This improves the organization and visibility of the book records.

The simplicity and efficiency of this system lie in the use of core Bash scripting features such as variables, conditionals, loops, functions, and text processing tools. Despite being lightweight and easy to implement, it performs all the necessary operations required for basic library management. The program ensures data consistency, reduces the need for manual record-keeping, and eliminates common human errors, making library tasks faster and more accurate.

In conclusion, the Library Management System using Bash demonstrates the power of shell scripting in real-world applications. It provides an organized, error-free, and automated method for managing books in a digital format. This project can be further improved by adding features like user authentication, fine calculation, issuing history, return management, and even graphical interfaces using tools like `dialog` or web-based extensions. Overall, it serves as a great example of how open-source scripting can be used to create practical and user-friendly solutions for day-to-day problems.

1. Introduction

1.1 Problem Statement & Objectives

Libraries are essential for providing access to knowledge, but managing them manually can be tiring and inefficient. Tasks like keeping track of available books, updating records, and issuing or returning books can take a lot of time and effort when done on paper or spreadsheets. Manual systems are not only slow, but they are also more likely to have mistakes like incorrect entries or misplaced books. This creates confusion, delays, and frustration for both library staff and users.

In today's digital world, automation is the key to making processes smooth and fast. Even a small library can benefit from using a basic digital solution that reduces paperwork and organizes information properly. The goal of this project is to build a Library Management System using simple Bash scripting, which can be used directly in the Linux terminal. This makes it light, easy to install, and perfect for environments with limited resources or no access to big database software.

The system is built to handle all the main functions of a library:

- Insert Book – Allows new books to be added with complete details such as Book ID, Book Name, Author, Publisher, Year, and Copies.
- Update Book – Modifies existing book data if anything changes (e.g., new copies added or updated info).
- Delete Book – Removes the book entry from the database if the book is no longer in stock.
- Search Book by ID – Quickly finds a book and its details using its unique Book ID.
- Display All Books – Shows all books in a clean table format, with all their details.
- Sort Books (A–Z) – Lists books in alphabetical order to make browsing easier.
- Issue Book – Issues a book to a reader and decreases the number of available copies.

All records are stored in a text file, which makes the system very easy to understand, maintain, and even transfer to other machines. This is very useful for school libraries, college departments, or personal book collections.

By using this system, librarians can save time, reduce mistakes, and make the library experience better for everyone. It also gives a clear view of book availability and ensures that book records are always up to date.

2. Literature Survey

2.1 Survey of Existing/Similar System

Traditional library management systems have evolved from manual record-keeping to digital solutions. The existing systems can be categorized into three types:

1. Manual Systems

Libraries using registers or spreadsheets to record book details, issues, and returns. These systems are prone to errors, time-consuming, and difficult to maintain.

2. Software-Based Systems

Libraries use custom software or database applications to manage book records. Examples include Microsoft Access-based systems or standalone desktop applications. While effective, they require installation, regular updates, and sometimes licensing costs.

3. Cloud-Based Library Management

Advanced systems like Koha, Evergreen, or commercial software store book records online and allow remote access. These provide high efficiency but require internet access and may involve costs.

Comparison with Our System

Our Bash script-based Library Management System is designed to be:

- **Simple and lightweight**
 - No need for databases or internet access.
- **Cost-effective**
 - It runs on any Linux-based system without additional software.
- **Efficient**
 - Allows quick book searches, updates, and issuing using command-line operations.
- **Easy to maintain**
 - The data is stored in a structured text file, making it accessible and modifiable.

This system is best suited for small libraries, personal collections, or educational institutions that need a basic yet effective book management solution

3. Proposed System

3.1 Introduction

The proposed Library Management System is a command-line based tool designed using Bash scripting to streamline the process of managing books in a library. It replaces traditional paper-based or manual systems with a digital solution that is simple, fast, and efficient. The system allows users to insert, update, delete, search, and issue books, while storing all records in a structured format within a text file. Its lightweight architecture makes it suitable for small libraries or personal use, without the need for complex databases. This system aims to minimize human error, save time, and provide quick access to book information in an organized way.

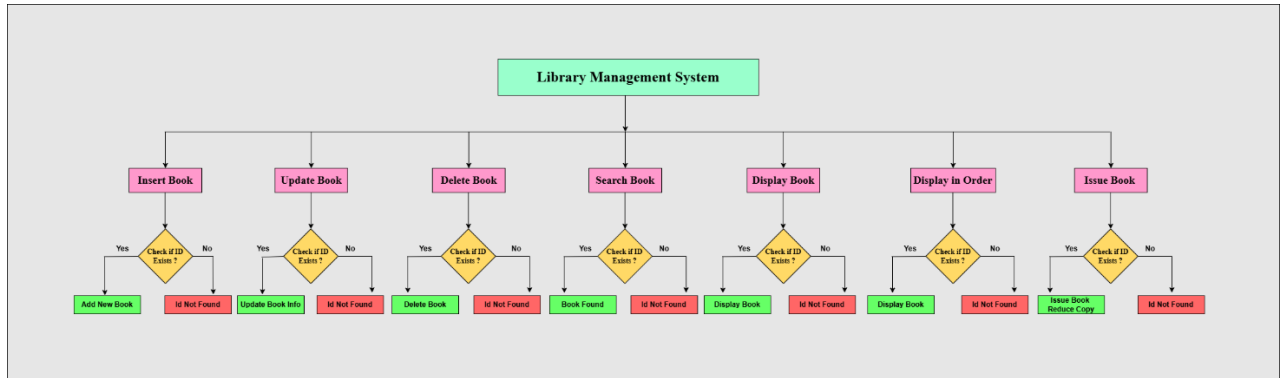
3.2 Architecture/ Framework

The architecture of the Library Management System is designed to be simple yet effective, using Bash scripting to handle all operations. The core idea is to maintain all book-related data in a plain text file (`library_data.txt`) which acts as a mini-database. Each line in the file represents a book record, with fields separated by colons (:), including Book ID, Book Name, Author Name, Publisher, Year, and Number of Copies.

The system works through a menu-driven interface, where the user is prompted to choose actions such as inserting a book, updating details, searching, issuing, or deleting records. When the user selects an option, the corresponding function is executed. Each function uses Unix tools like `grep`, `cut`, `sed`, `sort`, and `read` to manipulate and search data from the file.

The structure follows a modular approach, where each task (insert, update, delete, etc.) is written as a separate function for better readability and maintenance. This framework ensures that data is processed quickly and accurately without the need for external libraries or software.

Since the script runs in a terminal environment, it is lightweight and does not consume much system resources. This makes it ideal for small-scale systems like personal libraries, school libraries, or institutions with limited hardware.



3.3 Algorithm and Process Design

1) Insert a Book

1. Start
2. Enter Book ID, Name, Author, Publisher, Year, and Copies
3. Check if Book ID already exists
 Yes → Show message: "Book already exists"
 No → Save book details
4. Display "Book inserted successfully"
5. End

2) Update a Book

1. Start
2. Enter Book ID to update
3. Check if Book ID exists
 No → Show message: "Book not found"
 Yes → Enter new details (leave blank to keep existing data)
4. Update the book record
5. Display "Book updated successfully"
6. End

3) Delete a Book

1. Start
2. Enter Book ID to delete
3. Check if Book ID exists
 No → Show message: "Book not found"
 Yes → Remove book from the system
4. Display "Book deleted successfully"
5. End

4) Search for a Book

1. Start
2. Enter Book ID to search
3. Check if Book ID exists
 No → Show message: "Book not found"
 Yes → Show book details
4. End

5) Display All Books

1. Start
2. Retrieve and list all books
3. Show data in a tabular format
4. End

6) Display Books in Order (A-Z by Name)

1. Start
2. Retrieve and sort all books by name
3. Show sorted list
4. End

7) Issue a Book

1. Start
2. Enter Book ID to issue
3. Check if Book ID exists
 No → Show message: "Book not found"
 Yes → Check if copies are available
 No → Show message: "No copies left"
 Yes → Reduce copies count
4. Display "Book issued successfully"
5. End

3.4 Details of Hardware & Software

Sr.no	Name of Resources	Specification	Quantity
1.	Computer system	Laptop i5 Processor 11 th Generation	1
2.	Ms Word	Microsoft Word 2021 MSO 64 bit	-
3.	Internet	4G, Wi-Fi	-
4.	Book	Operating System By-Vijay Patil	1

3.4 Experiment and Results

Code

```
#!/bin/bash
# Define file paths for storing data
DATA_FILE="library_data.txt"

# Function to display menu
display_menu() {
    echo "Library Management System"
    echo "1. Insert a book"
    echo "2. Update a book"
    echo "3. Delete a book"
    echo "4. Search for a book by ID"
    echo "5. Display all books"
    echo "6. Display all books in order (A-Z)"
    echo "7. Issue a book"
    echo "8. Quit"
}

# Function to insert a book
insert_book() {
    echo "Insert a new book Details...."
    echo -n "Book ID: "
    read book_id
    echo -n "Book Name: "
    read book_name
    echo -n "Author Name: "
    read author_name
    echo -n "Publisher: "
    read publisher
    echo -n "Publisher Year: "
    read publisher_year
    echo -n "Copies: "
    read copies
    # Check if the book ID already exists
    if grep -q "^$book_id:" "$DATA_FILE"; then
        echo "Book with the same ID already exists..."
    else
        echo
        "$book_id:$book_name:$author_name:$publisher:$publisher_year:$copies"
        >> "$DATA_FILE"
        echo "Book inserted successfully..."
    fi
}
```

```

    fi
}

# Function to update a book by ID
update_book() {
    echo "Update a book by ID:"
    echo -n "Enter Book ID to Update: "
    read book_id

    if grep -q "^$book_id:" "$DATA_FILE"; then
        echo "Enter updated book information:"
        echo -n "Book Name (Enter to keep current value): "
        read new_book_name
        echo -n "Author Name (Enter to keep current value): "
        read new_author_name
        echo -n "Publisher (Enter to keep current value): "
        read new_publisher
        echo -n "Publisher Year (Enter to keep current value): "
        read new_publisher_year
        echo -n "Copies (Enter to keep current value): "
        read new_copies
        # Read the existing book entry
        existing_entry=$(grep "^$book_id:" "$DATA_FILE")
        # Extract the existing field values
        current_book_name=$(echo "$existing_entry" | cut -d ':' -f 2)
        current_author_name=$(echo "$existing_entry" | cut -d ':' -f 3)
        current_publisher=$(echo "$existing_entry" | cut -d ':' -f 4)
        current_publisher_year=$(echo "$existing_entry" | cut -d ':' -f 5)
        current_copies=$(echo "$existing_entry" | cut -d ':' -f 6)
        # Update fields with new values or keep the current value if no input is
        # provided
        updated_entry="$book_id:${new_book_name:-
$current_book_name}:${new_author_name:-
$current_author_name}:${new_publisher:-
$current_publisher}:${new_publisher_year:-
$current_publisher_year}:${new_copies:-$current_copies}"

        # Update the book entry
        sed -i "s/^$book_id:.*$/$updated_entry/" "$DATA_FILE"
        echo "Book updated successfully."
    else
        echo "Book with ID $book_id not found."
    fi
}

```

```

}
# Function to delete a book by ID
delete_book() {
    echo "Delete a book by ID:"
    echo -n "Enter Book ID to delete: "
    read book_id

    if grep -q "^$book_id:" "$DATA_FILE"; then
        sed -i "/^$book_id:/d" "$DATA_FILE"
        echo "Book deleted successfully."
    else
        echo "Book with ID $book_id not found."
    fi
}

# Function to search for a book by ID
search_book() {
    echo "Search for a book by ID:"
    echo -n "Enter Book ID to search: "
    read book_id

    if grep -q "^$book_id:" "$DATA_FILE"; then
        grep "^$book_id:" "$DATA_FILE"
    else
        echo "Book with ID $book_id not found."
    fi
}

# Function to display all books in tabular format
display_all_books() {
    echo "List of all books in tabular format:"
    echo "-----"
    printf "| %-6s | %-30s | %-20s | %-20s | %-6s | %-6s |\n" "ID" "Book
Name" "Author Name" "Publisher" "Year" "Copies"
    echo "-----"
    while IFS=: read -r book_id book_name author_name publisher
publisher_year copies; do
        printf "| %-6s | %-30s | %-20s | %-20s | %-6s | %-6s |\n" "$book_id"
"$book_name" "$author_name" "$publisher" "$publisher_year" "$copies"
    done < "$DATA_FILE"
    echo "-----"
}

```

```

# Function to display all books in order (A-Z)
display_books_in_order() {
    echo "List of books in alphabetical order:"
    echo "-----"
    printf "| %-6s | %-30s | %-20s | %-20s | %-6s | %-6s |\n" "ID" "Book
Name" "Author Name" "Publisher" "Year" "Copies"
    echo "-----"
    sort -t ':' -k 2,2 "$DATA_FILE" | while IFS=: read -r book_id book_name
author_name publisher publisher_year copies; do
        printf "| %-6s | %-30s | %-20s | %-20s | %-6s | %-6s |\n" "$book_id"
"$book_name" "$author_name" "$publisher" "$publisher_year" "$copies"
    done
    echo "-----"
}

# Function to issue a book
issue_book() {
    echo "Issue a book by ID:"
    echo -n "Enter Book ID to issue: "
    read book_id

    if grep -q "^$book_id:" "$DATA_FILE"; then
        current_copies=$(grep "^$book_id:" "$DATA_FILE" | cut -d ':' -f 6)
        if [ "$current_copies" -gt 0 ]; then
            sed -i
"s/^$book_id:(.):(.):(.):(.):(.*)/$book_id:\1:\2:\3:\4:$((current_copies -
1))/" "$DATA_FILE"
            echo "Book with ID $book_id issued successfully."
        else
            echo "No available copies of the book with ID $book_id."
        fi
    else
        echo "Book with ID $book_id not found."
    fi
}

# Main loop
while true; do
    display_menu
    echo -n "Enter your choice: "
    read choice
    case $choice in
        1) insert_book ;;

```

```
2) update_book ;;
3) delete_book ;;
4) search_book ;;
5) display_all_books ;;
6) display_books_in_order ;;
7) issue_book ;;
8) echo "Goodbye!"; exit 0 ;;
*) echo "Invalid choice. Please try again." ;;
esac
done
```

Result

```
Library Management System
1. Insert a book
2. Update a book
3. Delete a book
4. Search for a book by ID
5. Display all books
6. Display all books in order (A-Z)
7. Issue a book
8. Quit
Enter your choice: 1
Insert a new book Details....
Book ID: 5
Book Name: the midnight library
Author Name: matt haig
Publisher: techhills
Publisher Year: 1990
Copies: 270
Book inserted successfully...
```

Library Management System

1. Insert a book
2. Update a book
3. Delete a book
4. Search for a book by ID
5. Display all books
6. Display all books in order (A-Z)
7. Issue a book
8. Quit

Enter your choice: 5

List of all books in tabular format:

ID	Book Name	Author Name	Publisher	Year	Copies
1	wigns of fire	A.P.J.Kalam	techhills	1990	290
2	Mann mai hai vishwaas	IPS Vishwas nangare-patil	techhills	2018	56
3	TO GOOD TO BE	Prajakta Koli	xyz	2024	200
4	harry potter	j.k.rowling	harry	1998	200
5	the midnight library	matt haig	techhills	1990	270

Library Management System

1. Insert a book
2. Update a book
3. Delete a book
4. Search for a book by ID
5. Display all books
6. Display all books in order (A-Z)
7. Issue a book
8. Quit

Enter your choice: 2

Update a book by ID:

Enter Book ID to Update: 1

Enter updated book information:

Book Name (Enter to keep current value): wigns of fire

Author Name (Enter to keep current value): A.P.J.Kalam

Publisher (Enter to keep current value): techhills

Publisher Year (Enter to keep current value): 1990

Copies (Enter to keep current value): 290

Book updated successfully.

1. Insert a book
2. Update a book
3. Delete a book
4. Search for a book by ID
5. Display all books
6. Display all books in order (A-Z)
7. Issue a book
8. Quit

Enter your choice: 3

Delete a book by ID:

Enter Book ID to delete: 4

Book deleted successfully.


```
Library Management System
1. Insert a book
2. Update a book
3. Delete a book
4. Search for a book by ID
5. Display all books
6. Display all books in order (A-Z)
7. Issue a book
8. Quit
Enter your choice: 6
List of books in alphabetical order:
-----
| ID | Book Name | Author Name | Publisher | Year | Copies |
-----
| 4 | harry potter | j.k.rowling | harry | 1998 | 200 |
| 2 | Mann mai hai vishwaas | IPS Vishwas nangare-patil | techhills | 2018 | 56 |
| 5 | the midnight library | matt haig | techhills | 1990 | 270 |
| 3 | TO GOOD TO BE | Prajakta Koli | xyz | 2024 | 200 |
| 1 | wigns of fire | A.P.J.Kalam | techhills | 1990 | 290 |
-----
```

```
Library Management System
1. Insert a book
2. Update a book
3. Delete a book
4. Search for a book by ID
5. Display all books
6. Display all books in order (A-Z)
7. Issue a book
8. Quit
Enter your choice: 4
Search for a book by ID:
Enter Book ID to search: 3
3:TO GOOD TO BE:Prajakta Koli:xyz:2024:200
```

```
Library Management System
1. Insert a book
2. Update a book
3. Delete a book
4. Search for a book by ID
5. Display all books
6. Display all books in order (A-Z)
7. Issue a book
8. Quit
Enter your choice: 5
List of all books in tabular format:
-----
| ID | Book Name | Author Name | Publisher | Year | Copies |
-----
| 1 | wigns of fire | A.P.J.Kalam | techhills | 1990 | 290 |
| 2 | Mann mai hai vishwaas | IPS Vishwas nangare-patil | techhills | 2018 | 56 |
| 3 | TO GOOD TO BE | Prajakta Koli | xyz | 2024 | 200 |
| 5 | the midnight library | matt haig | techhills | 1990 | 270 |
-----
```

File Management

```
my.ini .project librar: X student_recc pr3e2.py pr3e1.py lis
File Edit View
1:wigns of fire:A.P.J.Kalam:techhills:1990:290
2:Mann mai hai vishwaas:IPS Vishwas nangare-patil:techhills:2018:56
3:TO GOOD TO BE:Prajakta Koli:xyz:2024:200
5:the midnight library:matt haig:techhills:1990:270
```

3.5 Conclusion and Future work

- **Conclusion:**

The Library Management System simplifies book handling by allowing users to add, update, delete, search, and issue books efficiently. The system ensures that book records are well-organized and easily accessible. It helps reduce manual errors and improves the overall management of library resources. By automating these tasks, the system enhances user experience and saves time for both librarians and students.

- **Future Work**

- GUI Integration for better usability.
- User Authentication with roles like Admin and Student.
- Book Return & Fine Calculation for better tracking.
- Cloud-Based Storage for secure data management.
- Recommendation System to suggest books based on user history.

4. References

1. Book Name is Operating System, by-Vijay Patil
2. <https://www.geeksforgeeks.org/introduction-of-operating-system-set-1/>
3. <https://www.jdoodle.com/test-bash-shell-script-online>