Submitted to

PURBANCHAL UNIVERSITY

Gomendra Multiple College

Birtamode, Jhapa

Project

Library Management System

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF BACHELOR OF COMPUTER APPLICATION

Submitted by

Susmita Dangal

Prajita Thapa

Sarad Chandra Acharya

Year/Semester: II

Acknowledgement

We would like to extend our profound appreciation and unfathomable regards to the Information Technology (IT) department for this commendable guidance, monitoring and constant encouragement through the course of this project. The help and guidance by shall carry is a long way, in the journey in which we are about embark.

We would like to express my sincere gratitude to my teacher, Mr. Jhalnath Chapagain, for his invaluable guidance and support throughout the course of this project. Additionally, I am deeply appreciative of our chief, Dr. Rupak Khanal, and Purbanchal University for providing me with the golden opportunity to undertake this project on the topic of Library Management System. This experience has been instrumental in expanding my knowledge and conducting extensive research, for which I am truly grateful.

We would also like to extend my special thanks to my group members for their cooperation and dedicated efforts towards the success of this project.

Table of content

Introduction

The Library Management System (LMS) is a software application designed to manage the functions of a library efficiently. This project, developed in C++, aims to streamline various library operations such as cataloging books, managing user accounts, tracking borrowed and returned books, and facilitating search functionalities. The LMS provides an organized and user-friendly interface that allows librarians and users to perform their tasks with ease and accuracy.

- **Core Language: C++**

This program ‘Library Management System is an application design for recording and tracking the borrowing, adding, receiving and other facility provided by the library. This program is developed in such a way so that any library can easily use to record the customer’s/ Student’s detail with the little change as per needed. It has been designed in a simple manner so that the user won’t find him/her any difficulties while the system.

The Library Management System (LMS) is built using a modular architecture to ensure scalability and maintainability. The system is developed in C++ due to its performance efficiency and object-oriented features, which are ideal for managing the complex data structures involved in library management.

Project Objective

1) Automation of Library Operations
2) Efficient Catalog Management
3) User Account Management
4) Enhanced Search Functionality
5) Tracking Borrowing and Returning
6) Report Generation

Methodologies used in the project

This project is developed for the efficient storage and maintaining the records of books and also the students using the files and other logics of C++ programing language. The Librarian can proceed to add, delete, modify the books records and the students records easily through this system.

System Feature

1) It reduces the related cost and time
2) The program is simple and the easy to use
3) The user with no prior knowledge can use the system easily
4) This software can run with minimal hardware requirement

System Limitation

1) This is meant for small organization, data usages above 2GB may discover limitations
2) One file, all the information of the database from the program is stored in a single file
3) User must have a least the knowledge of English language
4) Wrong input may crash the system (System restart required)

Hardware Requirement

We have built this product thinking that it will solve the institution’s info keeping problems of the calculation which is related to computer. Our product is based for the single PC. But single PC based product is not sufficient some hardware requirements should be known before running this product so that the user will not face any hardware problems at the run time. As the client’s requirements and the platform in which we are going to develop this product, the hardware requirement can be listed below: -

- PC with 128 MB RAM and above.
- PC with Pentium 4 processor and above.
- PC with at least one GB free space on hard-drive.
- PC should have standard keyboard for input data.
- PC should have any type of monitor for display.

Software Requirement

Only the hardware is not sufficient to run this program. There must be compatible software installed in the PC to conduct the program successfully. Different program requires different software installed to run according to the platform in which the developers have developed it.

As per our program following are software requirements: -

- There must be window environment operating system.
- Front end: Dev C++ or Visual Studio or Turbo C
- Windows 8 (min)

Future strategy

As the semester progress, we will be introduced with new programming language and with be familiarized with new tools so for the future the project will receive a huge upgrade from its CLI interface to GUI interface.

We will be adding receipts print function and also solving the one file database problem using more advance and more reliable SQL database.

Codes

```
#include <iostream>
#include <fstream>
#include <cstring>
#include <conio.h>
using namespace std;
class Book
{
   int id;
   char title[50];
   char author[50];
   bool available;

public:
   Book()
   {
      available = true; // By default, book is available when created
   }

public:
   void createBook()
   {
      cout << "Enter book ID: ";
      cin >> id;
      cin.ignore();
      cout << "Enter book title: ";
      cin.getline(title, 50);
      cout << "Enter book author: ";
      cin.getline(author, 50);
```

```
}

void displayBook() const
{
    cout << "ID: " << id << "\n";
    cout << "Title: " << title << "\n";
    cout << "Author: " << author << "\n";
}

int getId() const
{
    return id;
}
const char *getTitle() const
{
    return title;
}

const char *getAuthor() const
{
    return author;
}

void modifyBook()
{
    cout << "Enter new title: ";
    cin.ignore();
    cin.getline(title, 50);
    cout << "Enter new author: ";
    cin.getline(author, 50);
    cout << "Is the book available? (1 for yes, 0 for no): ";
    cin >> available;
}

bool isAvailable() const
{
    return available;
}

void setAvailable(bool status)
{
    available = status;
}

void saveToFile(ofstream &outFile) const
{
    outFile << id << "\n";
    outFile << title << "\n";
    outFile << author << "\n";
    outFile << available << "\n"; // Save availability status
}

bool loadFromFile(ifstream &inFile)
{
    if (inFile >> id)
    {
        inFile.ignore();
        inFile.getline(title, 50);
        inFile.getline(author, 50);
        inFile >> available;
```

```
            inFile.ignore(); // Ignore the newline after the availability
            return true;
        }
        return false;
    }
};

bool checkPassword(const char *password)
{
    return strcmp(password, "library") == 0;
}

void displayMenu(); // Function declaration
void bookIssue();
void bookDeposit();
void administrativeMenu();
void createBook();
void displayBooks();
void searchBook();
void modifyBook();
void deleteBook();

int main()
{
    char password[20] = "library";
    char inputPassword[20];
    cout << "\n \n  ";
    cout << "Enter the password to access the system: ";
    cin >> inputPassword;
```

```
if (!checkPassword(inputPassword))
{
    cout << "Wrong password. Press any key to exit.";
    _getch();
    return 0;
}

while (true)
{
    displayMenu();
    int choice;
    cin >> choice;

    switch (choice)
    {
    case 1:
        bookIssue();
        break;
    case 2:
        bookDeposit();
        break;
    case 3:
        administrativeMenu();
        break;
    case 4:
        cout << "Exiting the program.\n";
        exit(0);
    default:
        cout << "Invalid choice. Please try again.\n";
    }
    }

    return 0;
}

void displayMenu()
{
    cout << "\n \n";
    cout << "\n ***Welcome to Library Management System***\n";
    cout << "1. Book Issue\n";
    cout << "2. Book Deposit\n";
    cout << "3. Administrative Menu\n";
    cout << "5. Exit\n";
    cout << "Enter your choice: ";
}

void bookIssue()
{
    int bookId;
    cout << "\n \n";
    cout << "Enter book ID to issue: ";
    cin >> bookId;

    // Open the books file for reading and writing
    ifstream inFile("books.txt");
    ofstream outFile("temp.txt");

    if (!inFile || !outFile)
    {
        cout << "Error opening file.\n";
    return;
}

Book book;
bool found = false;
while (book.loadFromFile(inFile))
{
    if (book.getId() == bookId)
    {
        if (book.isAvailable())
        {
            book.setAvailable(false); // Mark book as issued
            cout << "Book issued successfully.\n";
        }
        else
        {
            cout << "Book is already issued.\n";
        }
        found = true;
    }
    book.saveToFile(outFile);
}

inFile.close();
outFile.close();

if (!found)
{
    cout << "Book with ID " << bookId << " not found.\n";
}
```

```
    else
    {
        remove("books.txt");
        rename("temp.txt", "books.txt");
    }
}

void bookDeposit()
{
    int bookId;
    cout << "Enter book ID to deposit: ";
    cin >> bookId;

    ifstream inFile("books.txt");
    ofstream outFile("temp.txt");

    if (!inFile || !outFile)
    {
        cout << "Error opening file.\n";
        return;
    }

    Book book;
    bool found = false;
    while (book.loadFromFile(inFile))
    {
        if (book.getId() == bookId)
        {
            if (!book.isAvailable())
            {
                book.setAvailable(true); // Mark book as available
                cout << "Book deposited successfully.\n";
            }
            else
            {
                cout << "Book is already available.\n";
            }
            found = true;
        }
        book.saveToFile(outFile);
    }

    inFile.close();
    outFile.close();

    if (!found)
    {
        cout << "Book with ID " << bookId << " not found.\n";
    }
    else
    {
        remove("books.txt");
        rename("temp.txt", "books.txt");
    }
}

void administrativeMenu()
{
    while (true)
    {
```

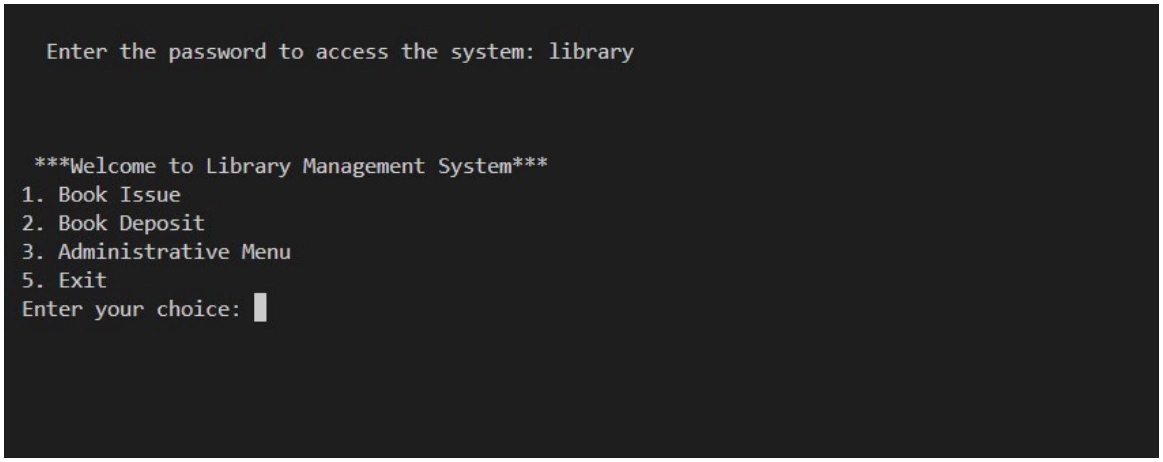
```
cout << "\nAdministrative Menu:\n";
cout << "1. Create Book\n";
cout << "2. Display Books\n";
cout << "3. Search Book\n";
cout << "4. Modify Book\n";
cout << "5. Delete Book\n";
cout << "6. Return to Main Menu\n";
cout << "Enter your choice: ";
int choice;
cin >> choice;

switch (choice)
{
case 1:
   createBook();
   break;
case 2:
   displayBooks();
   break;
case 3:
   searchBook();
   break;
case 4:
   modifyBook();
   break;
case 5:
   deleteBook();
   break;
case 6:
   return;
```

```
        default:
            cout << "Invalid choice. Please try again.\n";
        }
    }
}
void createBook()
{
    Book newBook;
    int newId;
    bool idExists;

    do
    {
        newBook.createBook();
        newId = newBook.getId();
        idExists = false; // Initialize idExists to false for each iteration

        ifstream inFile("books.txt");
        Book existingBook;

        while (existingBook.loadFromFile(inFile))
        {
            if (existingBook.getId() == newId)
            {
                idExists = true;
                break;
            }
        }
        inFile.close();
        if (idExists)
        {
            cout << "A book with ID " << newId << " already exists. Please enter another ID.\n";
        }
        else
        {
            ofstream outFile("books.txt", ios::app);
            newBook.saveToFile(outFile);
            outFile.close();
            cout << "Book created successfully.\n";
        }
    } while (idExists); // Loop until a unique ID is entered
}

void displayBooks()
{
    ifstream inFile("books.txt");
    if (!inFile)
    {
        cout << "No books available.\n";
        return;
    }

    Book book;
    int totalBooks = 0;
    int availableBooks = 0;
    int issuedBooks = 0;

    cout << "\nBooks List:\n";
    while (book.loadFromFile(inFile))
    {
        book.displayBook();
        cout << "--------------------\n";
        totalBooks++;
        if (book.isAvailable())
        {
            availableBooks++;
        }
        else
        {
            issuedBooks++;
        }
    }
    inFile.close();

    cout << "Total books: " << totalBooks << "\n";
    cout << "Available books: " << availableBooks << "\n";
    cout << "Issued books: " << issuedBooks << "\n";
}

void searchBook()
{
    int id;
    cout << "Enter book ID to search: ";
    cin >> id;

    ifstream inFile("books.txt");
    if (!inFile)
    {
        cout << "No books available.\n";
        return;
    }

    Book book;
    bool found = false;
    while (book.loadFromFile(inFile))
    {
        if (book.getId() == id)
        {
            cout << "Book found:\n";
            book.displayBook();
            found = true;
        }
    }
    inFile.close();

    if (!found)
    {
        cout << "Book with ID " << id << " not found.\n";
    }
}

void modifyBook()
{
    int id;
    cout << "Enter book ID to modify: ";
    cin >> id;

    ifstream inFile("books.txt");
```

```
ofstream outFile("temp.txt");

if (!inFile || !outFile)
{
   cout << "Error opening file.\n";
   return;
}

Book book;
bool found = false;
while (book.loadFromFile(inFile))
{
   if (book.getId() == id)
   {
      book.modifyBook();
      cout << "Book modified successfully.\n";
      found = true;
   }
   book.saveToFile(outFile);
}

inFile.close();
outFile.close();

remove("books.txt");
rename("temp.txt", "books.txt");

if (!found)
{
   cout << "Book with ID " << id << " not found.\n";
```

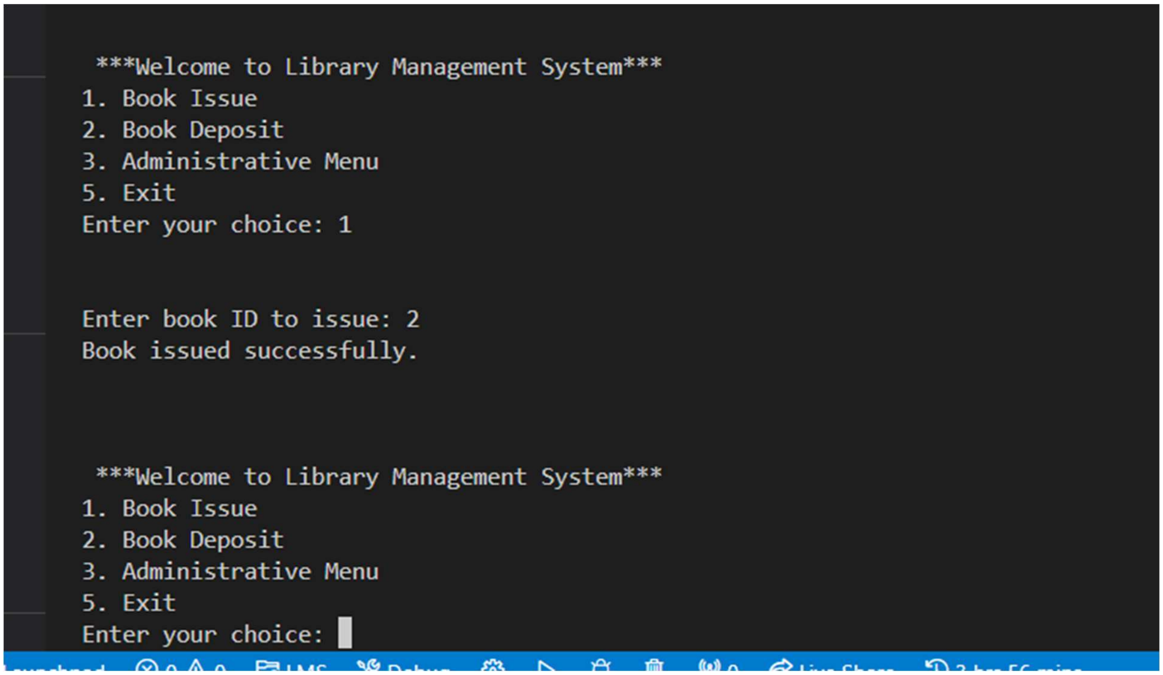
```
    }
}

void deleteBook()
{
    int id;
    cout << "Enter book ID to delete: ";
    cin >> id;

    ifstream inFile("books.txt");
    ofstream outFile("temp.txt");
    if (!inFile || !outFile)
    {
        cout << "Error opening file.\n";
        return;
    }

    Book book;
    bool found = false;
    while (book.loadFromFile(inFile))
    {
        if (book.getId() == id)
        {
            found = true;
        }
        else
        {
            book.saveToFile(outFile);
        }
    }
    inFile.close();
    outFile.close();

    remove("books.txt");
    rename("temp.txt", "books.txt");

    if (found)
    {
        cout << "Book deleted successfully.\n";
    }
    else
    {
        cout << "Book with ID " << id << " not found.\n";
    }
}
```

Login Pannel



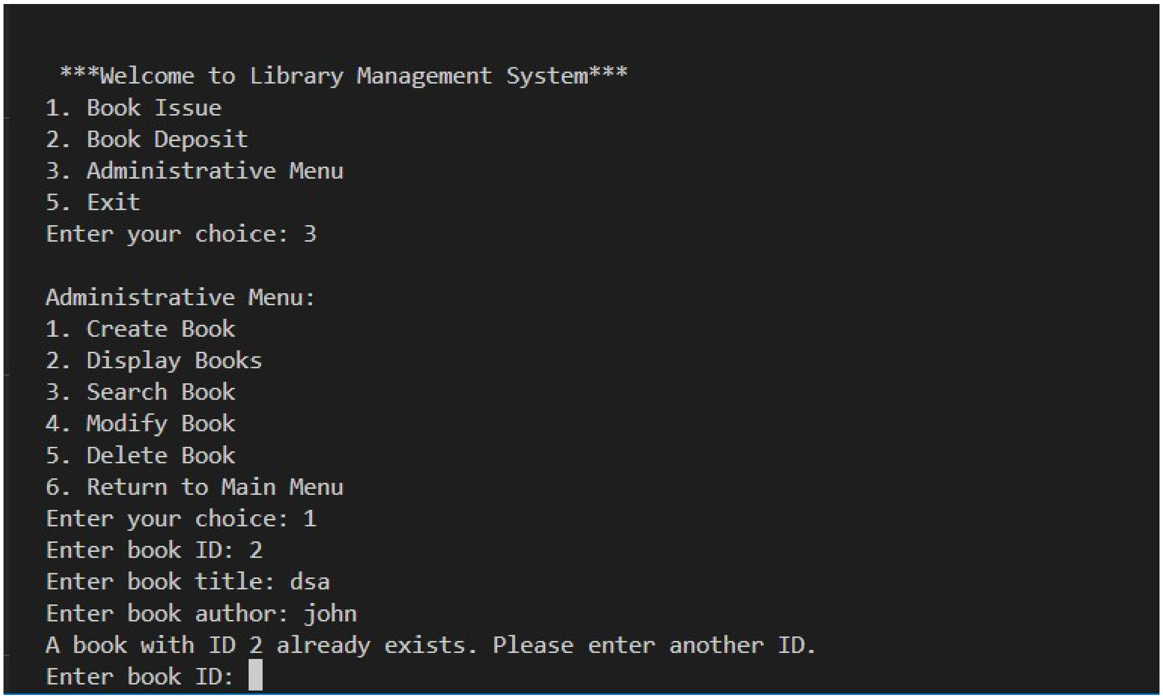
```
  Enter the password to access the system: library



 ***Welcome to Library Management System***
1. Book Issue
2. Book Deposit
3. Administrative Menu
5. Exit
Enter your choice: 
```

Book Issue



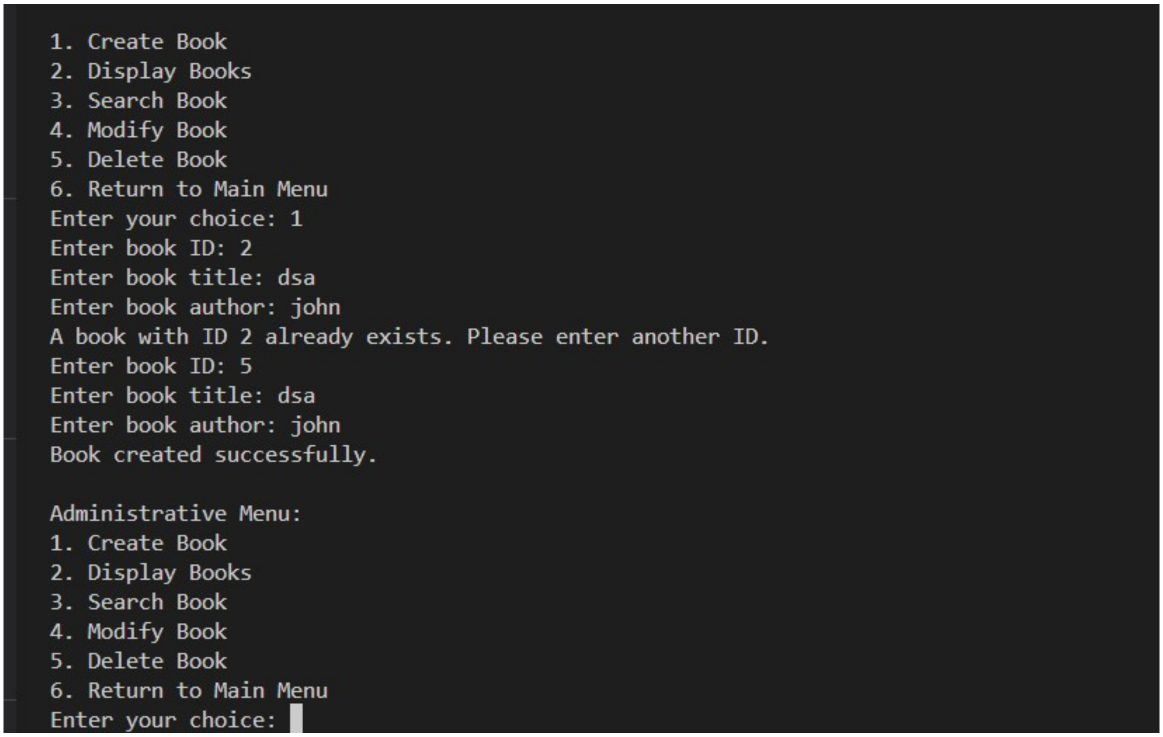
```
 ***Welcome to Library Management System***
1. Book Issue
2. Book Deposit
3. Administrative Menu
5. Exit
Enter your choice: 1


Enter book ID to issue: 2
Book issued successfully.



 ***Welcome to Library Management System***
1. Book Issue
2. Book Deposit
3. Administrative Menu
5. Exit
Enter your choice: 
```

Administrative Menu



```
 ***Welcome to Library Management System***
1. Book Issue
2. Book Deposit
3. Administrative Menu
5. Exit
Enter your choice: 3

Administrative Menu:
1. Create Book
2. Display Books
3. Search Book
4. Modify Book
5. Delete Book
6. Return to Main Menu
Enter your choice: 1
Enter book ID: 2
Enter book title: dsa
Enter book author: john
A book with ID 2 already exists. Please enter another ID.
Enter book ID: 
```

Create Book



```
1. Create Book
2. Display Books
3. Search Book
4. Modify Book
5. Delete Book
6. Return to Main Menu
Enter your choice: 1
Enter book ID: 2
Enter book title: dsa
Enter book author: john
A book with ID 2 already exists. Please enter another ID.
Enter book ID: 5
Enter book title: dsa
Enter book author: john
Book created successfully.

Administrative Menu:
1. Create Book
2. Display Books
3. Search Book
4. Modify Book
5. Delete Book
6. Return to Main Menu
Enter your choice: 
```

Abstract

This project showcases the design and implementation of Library Management System (LMS) using C++ programming Language. This project is designed to facilitate the efficient management of library operations like Automation of Library Operations Automation of Library Operations like Efficient catalog management, User account management, Enhanced search functionality, tracking borrowing and returning and also report generation.

The core functionalities of the system are to adding, updating, and deleting the book records, registering and managing the user accounts, and processing the borrowing. The user interface is designed to be intuitive and user-friendly, providing a seamless interaction experience for librarians and library patrons. The implementation leverages C++ Standard Library components and follows best practices in software engineering to ensure high performance and stability.

This Library Management System demonstrates the applicability of C++ in developing comprehensive management systems, offering a valuable tool for libraries to streamline their operations and improve service delivery.

Conclusion

Finally, we are success to create the project for 2nd semester. Our knowledge and skill now come in work. We are very glad and happy because we meet our success point. All the activities of project like documentation, coding is successfully carried in the certain period.

This experience of carrying our project, entitled “Library Management System”, will help us in developing big projects in coming days. We, the beginners of BCA 2^{nd} semester are hopeful and confident that with the learning of many more topic related to such development work, we will be able to do something better in our future projects in upcoming semesters. All feedback is very welcome- let us know what you found useful and what you found not-so-useful and what additional information you would have liked to be included in future program.

We are thankful to the management of our college for this creative work.

Without their cooperation it would not be possible to complete the project. We would like to express our gratitude to all colleagues who helped us by providing valuable suggestion. It’s very challenging for us to develop project on C++ without any previous knowledge. We would like to thank to our supervisor Mr.Jhalnath Chapagain for this creative and great support. We would also like to thank all our colleagues and Chief of this college. Finally thank you all individual who have assisted and encouraged us in the developing of this project.