**BOOKSHOP MANAGEMENT SYSTEM**

A CAPSTONE PROJECT REPORT

# (Object Oriented Programming with C++ in Advanced Topics- DSA0199)

***Submitted to***

**SAVEETHA INSTITUTE OF MEDICAL AND TECHNICAL SCIENCES**

***In partial fulfillment for the award of the degree of***

**BACHELOR OF ENGINEERING IN COMPUTER SCIENCE & ENGINEERING**

***By***

**R.DINESH (19211430),**

**V.VASANTH (19211416)**

***Course Faculty***

**Ms.S.Jayanthi**



**SAVEETHA SCHOOL OF ENGINEERING, SIMATS, CHENNAI - 602105**

**SEPTEMBER-2024**

**SAVEETHA SCHOOL OF ENGINEERING**

**SAVEETHA INSTITUE OF MEDICAL AND TECHNICAL SCIENCES,CHENNAI - 602105**

**BONAFIDE CERTIFICATE**

Certified that this project report **“BOOKSHOP MANAGEMENT SYSTEM”** is the Bonafide work of **“R.DINESH,V.VASANTH”** who carried out the project work under my supervision.

**Submitted to**

**Ms. S. Jayanthi**

(Course Faculty)

Department of Information security, Saveetha School of Engineering, SIMATS

## SIGNATURE of Course Faculty

**ACKNOWLEDGEMENT**

This project work would not have been possible without the contribution of many people. It gives me immense pleasure to express my profound gratitude to our Honorable Chancellor **Dr. N. M. Veeraiyan**, Saveetha Institute of Medical and Technical Sciences, for his blessings and for being a source of inspiration. I sincerely thank our Director of Academics **Dr. Deepak Nallaswamy,** SIMATS, for his visionary thoughts and support. I am indebted to extend my gratitude to our Director **Dr. Ramya Deepak,** Saveetha School of Engineering, for facilitating us all the facilities and extended support to gain valuable education and learning experience.

I register my special thanks to **Dr. B. Ramesh,** Principal, Saveetha School of Engineering for the support given to me in the successful conduct of this project. I wish to express my sincere gratitude to my Course faculty **Ms.Jayanthi**, for his inspiring guidance, personal involvement and constant encouragement during the entire course of this work.

I am grateful to Project Coordinators, Review Panel External and Internal Members and the entire faculty of the Department of Design, for their constructive criticisms and valuable suggestions which have been a rich source to improve the quality of this work.

**STUDENT NAME’s**

**TABLE OF CONTENTS**

|  |  |
| --- | --- |
| **CHAPTER**  **NO** | **TITLE** |
| **1** | **Introduction** |
| **2** | **Project Description and Goals:** |
| **3** | **Technical Specifications:** |
| **4** | **Design Approach and Details** |
| **5** | **Schedule, Tasks, and Milestones:** |
| **6** | **Project Demonstration:** |
| **7** | **Cost Analysis:** |
| **8** | **Result:** |
| **9** | **Discussion:** |
| **10** | **Conclusion:** |

**Introduction:**

The **Bookshop Management System Application** is a state-of-the-art solution designed to transform the operations of bookshops. By employing object-oriented programming principles, the application provides a comprehensive platform for managing all aspects of bookshop operations with efficiency and ease. At its core, the system includes essential features such as a secure login page for various user roles, an admin page for overseeing book inventory, sales reports, and user accounts, and a user page that offers a personalized experience for browsing books, making purchases, and tracking orders. This robust application is aimed at streamlining bookshop management, enhancing user experience, and boosting overall operational effectiveness in the book retail industry.

the Bookshop Management System Application is a cutting-edge solution designed to transform bookshop operations. By employing object-oriented programming principles, the application provides a comprehensive platform for managing all aspects of bookshop management efficiently. It features a secure login page for various user roles, an admin page for overseeing book inventory, sales reports, and user accounts, and a user page that offers a personalized experience for browsing books, making purchases, and tracking orders. Moreover, the application facilitates seamless order management, allowing staff to efficiently process orders, track their status, and manage details effectively. With real-time order tracking functionality, both customers and staff can monitor progress, ensuring transparency and timely service delivery. The billing feature simplifies invoicing by accurately calculating totals and generating invoices for customers. By integrating these functionalities into a cohesive and user-friendly interface, the Bookshop Management System empowers bookshops to optimize operations, improve customer service, and drive business growth. As the retail industry evolves, this application stands out as a vital tool for modern bookshops striving to stay ahead and deliver exceptional experiences.Project

**Description and Goals:**

The Bookshop Management System is a sophisticated software solution designed to enhance the efficiency of bookshop operations through a class-based architecture. This comprehensive system includes key features such as a secure login page, an admin page for managing inventory and user accounts, a user page for customer interactions, and advanced functionalities for handling order details, real-time order tracking, and billing. The primary goal of the system is to create a robust platform that streamlines various aspects of bookshop management, improves operational efficiency, and enriches the customer experience. By integrating these features into a user-friendly interface, the system aims to facilitate smoother bookstore operations, enhance customer service, and support the growth and success of modern bookshops.

**Functionality:**

The Bookshop Management System is designed to offer a range of essential functionalities to streamline bookshop operations. It provides user authentication to ensure secure access for different user roles, including customers and administrator

**User-Friendly Interface:**

the Bookshop Management System will feature an intuitive interface designed to ensure ease of use for both staff and customers. The interface will incorporate clear navigation and user-friendly controls, making it straightforward for users to access and utilize the system’s various functionalities.

**Accurate Calculation:**

The Bookshop Management System will implement precise algorithms to ensure accurate calculations for order totals, taxes, discounts, and additional charges. By employing meticulous computational methods

**Error Handling:**

The Bookshop Management System will incorporate robust error handling mechanisms to detect and manage exceptions gracefully

**Cross-Platform Compatibility:**

the Bookshop Management System will be developed using technologies that ensure compatibility across various platforms, including web and mobile devices.

**Documentation and Support:**

The Bookshop Management System will include comprehensive documentation to assist users in navigating the system effectively.

**Testing and Validation:**

The Bookshop Management System will undergo thorough testing of all its components to ensure optimal functionality, performance, and security. Rigorous testing procedures will be implemented, including unit testing

**Technical Specifications:**

the Bookshop Management System will be developed using appropriate programming languages, frameworks, and databases selected to meet the project's specific requirements. The choice of technology stack will focus on ensuring system efficiency, scalability

**Platform Compatibility:**

The Bookshop Management System will ensure compatibility with popular operating systems and web browsers

**Design Approach and Details:**

The Bookshop Management System will be developed using a modular and scalable design approach to facilitate future enhancements and modifications. By adopting object-oriented design principles

**Schedule, Tasks, and Milestones:**

**Planning Phase:**

Define project scope, objectives, and requirements.

Identify stakeholders and user personals.

Create a project plan with timelines and resource allocation**.**

**Design Phase:**

Design system architecture and database schema.

Develop wire frames and mock ups for user interfaces.

Define class structures and relationships.

**Project Demonstration:**

A demonstration will be organized to showcase system features and functionalities, gathering feedback for further refinement and improvement.

**Cost Analysis:**

Project costs, including development resources, licenses, and infrastructure, will be estimated and compared with expected benefits and returns on investment.

**Result:**

The Bookshop Management System stands as a comprehensive solution catering to the complex needs of modern bookshops. It successfully integrates functionalities such as order management, inventory tracking, customer interactions, and billing, streamlining operations and enhancing overall efficiency.

**Discussion:**

The development process for the Bookshop Management System involved meticulous planning, rigorous testing, and iterative refinement to address the diverse needs of bookshop owners and staff. Collaboration with stakeholders was crucial in shaping the system's features and usability, ensuring alignment with real-world requirements. By incorporating continuous feedback loops, the project maintained a user-centric approach throughout its lifecycle, allowing for timely adjustments and enhancements based on actual user experiences.

Furthermore, the system's scalability and adaptability empower bookshops to accommodate growth and evolving operational dynamics seamlessly. By leveraging technology effectively, bookshops can enhance customer service, optimize resource utilization, and achieve operational excellence. Looking ahead, ongoing support and maintenance will be essential to sustain the system's performance and address emerging challenges in the ever-changing retail landscape.

**Summary:**

the Bookshop Management System revolutionizes bookshop operations by offering a comprehensive platform for efficient management of orders, inventory, customer interactions, and billing. With its user-friendly interface and accurate functionalities, the system streamlines daily operations and significantly enhances customer satisfaction

**Conclusion:**

In conclusion, the Bookshop Management System emerges as a vital tool for modern bookshops aiming to excel in a competitive market. By leveraging technology to automate and optimize key processes, the system enhances operational efficiency, reduces costs, and improves the overall customer experience.

**Code:**

#include <fstream>

#include <iostream>

#include <stdio.h>

#include <stdlib.h>

using namespace std;

class bookshope {

public:

void control\_panel();

void add\_book();

void show\_book();

void check\_book();

void update\_book();

void del\_book();

};

void bookshope::control\_panel()

{

system("cls");

cout << "\n\n\t\t\t\tCONTROL PANEL";

cout << "\n\n1. ADD BOOK";

cout << "\n2. DISPLAY BOOKS";

cout << "\n3. CHECK PARTICULAR BOOK";

cout << "\n4. UPDATE BOOK";

cout << "\n5. DELETE BOOK";

cout << "\n6. EXIT";

}

void bookshope::add\_book()

{

system("cls");

fstream file;

int no\_copy;

string b\_name, a\_name, b\_id;

cout << "\n\n\t\t\t\tADD BOOKS";

cout << "\n\nBook ID : ";

cin >> b\_id;

cout << "\nBook Name : ";

cin >> b\_name;

cout << "\nAuthor Name : ";

cin >> a\_name;

cout << "\nNo. of Books : ";

cin >> no\_copy;

file.open("D:// book.txt",

ios::out | ios::app);

file << " " << b\_id << " "

<< b\_name << " " << a\_name

<< " " << no\_copy << "\n";

file.close();

}

void bookshope::show\_book()

{

system("cls");

fstream file;

int no\_copy;

string b\_name, b\_id, a\_name;

cout << "\n\n\t\t\t\t\tAll BOOKS";

file.open("D:// book.txt", ios::in);

if (!file)

cout << "\n\nFile Opening Error!";

else {

cout << "\n\n\nBook ID\t\tBook"

<< "\t\tAuthor\t\tNo. of "

"Books\n\n";

file >> b\_id >> b\_name;

file >> a\_name >> no\_copy;

while (!file.eof()) {

cout << " " << b\_id

<< "\t\t" << b\_name

<< "\t\t" << a\_name

<< "\t\t" << no\_copy

<< "\n\n";

file >> b\_id >> b\_name;

file >> a\_name >> no\_copy;

}

system("pause");

file.close();

}

}

void bookshope::check\_book()

{

system("cls");

fstream file;

int no\_copy, count = 0;

string b\_id, b\_name, a\_name, b\_idd;

cout << "\n\n\t\t\t\tCheck "

<< "Particular Book";

// Open the file in input mode

file.open("D:// book.txt", ios::in);

if (!file)

cout << "\n\nFile Opening Error!";

else {

cout << "\n\nBook ID : ";

cin >> b\_idd;

file >> b\_id >> b\_name;

file >> a\_name >> no\_copy;

while (!file.eof()) {

if (b\_idd == b\_id) {

system("cls");

cout << "\n\n\t\t\t\t"

<< "Check Particular Book";

cout << "\n\nBook ID : "

<< b\_id;

cout << "\nName : "

<< b\_name;

cout << "\nAuthor : "

<< a\_name;

cout << "\nNo. of Books : "

<< no\_copy;

cout << endl

<< endl;

count++;

break;

}

file >> b\_id >> b\_name;

file >> a\_name >> no\_copy;

}

system("pause");

file.close();

if (count == 0)

cout << "\n\nBook ID Not"

<< " Found...";

}

}

// Function to update the book

void bookshope::update\_book()

{

system("cls");

fstream file, file1;

int no\_copy, no\_co, count = 0;

string b\_name, b\_na, a\_name;

string a\_na, b\_idd, b\_id;

cout << "\n\n\t\t\t\tUpdate Book Record";

file1.open("D:// book1.txt",

ios::app | ios::out);

file.open("D:// book.txt", ios::in);

if (!file)

cout << "\n\nFile Opening Error!";

else {

cout << "\n\nBook ID : ";

cin >> b\_id;

file >> b\_idd >> b\_name;

file >> a\_name >> no\_copy;

// Till end of file is reached

while (!file.eof()) {

if (b\_id == b\_idd) {

system("cls");

cout << "\t\t\t\t"

<< "Update Book Record";

cout << "\n\nNew Book Name : ";

cin >> b\_na;

cout << "\nAuthor Name : ";

cin >> a\_na;

cout << "\nNo. of Books : ";

cin >> no\_co;

file1 << " " << b\_id << " "

<< b\_na << " "

<< a\_na << " " << no\_co

<< "\n\n";

count++;

}

else

file1 << " " << b\_idd

<< " " << b\_name

<< " " << a\_name

<< " " << no\_copy

<< "\n\n";

file >> b\_idd >> b\_name;

file >> a\_name >> no\_copy;

}

if (count == 0)

cout << "\n\nBook ID"

<< " Not Found...";

}

cout << endl;

system("pause");

// Close the files

file.close();

file1.close();

remove("D:// book.txt");

rename("D:// book1.txt",

"D:// book.txt");

}

void bookshope::del\_book()

{

system("cls");

fstream file, file1;

int no\_copy, count = 0;

string b\_id, b\_idd, b\_name, a\_name;

cout << "\n\n\t\t\t\tDelete a Book";

file1.open("D:// book1.txt",

ios::app | ios::out);

file.open("D:// book.txt",

ios::in);

if (!file)

cout << "\n\nFile Opening Error...";

else {

cout << "\n\nBook ID : ";

cin >> b\_id;

file >> b\_idd >> b\_name;

file >> a\_name >> no\_copy;

while (!file.eof()) {

if (b\_id == b\_idd) {

system("cls");

cout << "\n\n\t\t\t\t"

<< "Delete a Book";

cout << "\n\nBook is Deleted "

"Successfully...\n\n";

count++;

}

else

file1 << " " << b\_idd

<< " " << b\_name

<< " " << a\_name

<< " " << no\_copy

<< "\n\n";

file >> b\_idd >> b\_name;

file >> a\_name >> no\_copy;

}

if (count == 0)

cout << "\n\nBook ID "

<< "Not Found...";

}

system("pause");

// Close the file

file.close();

file1.close();

remove("D:// book.txt");

rename("D:// book1.txt",

"D:// book.txt");

}

void bookShopRecord()

{

int choice;

char x;

bookshope b;

while (1) {

b.control\_panel();

cout << "\n\nEnter your choice : ";

cin >> choice;

switch (choice) {

case 1:

do {

b.add\_book();

cout << "\n\nWant to add"

<< " another book? "

"(y/n) : ";

cin >> x;

} while (x == 'y');

break;

case 2:

b.show\_book();

break;

case 3:

b.check\_book();

break;

case 4:

b.update\_book();

break;

case 5:

b.del\_book();

break;

case 6:

exit(0);

break;

default:

cout << "\n\nINVALID CHOICE\n";

}

}

}

// Driver Code

int main()

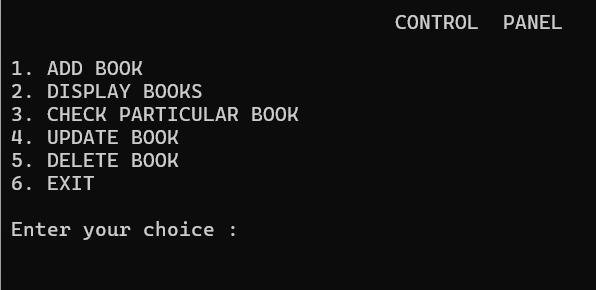
{

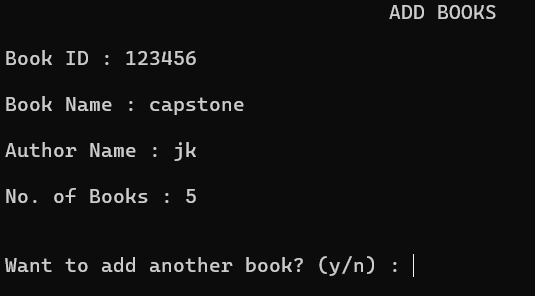
bookShopRecord();

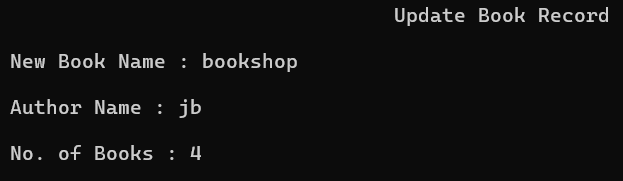
return 0;

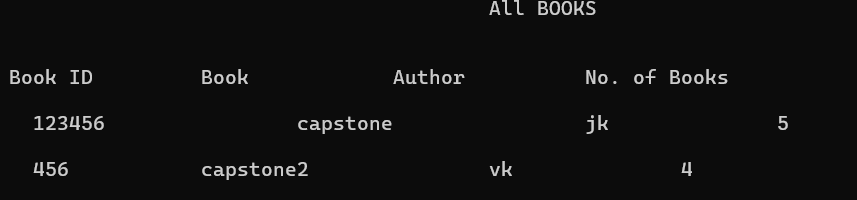
}

**OUTPUT:**



****

****

****