

PROJECT REPORT

Computer Programming Lab (CSL-113)



PROJECT TITLE:

BS (CS) – 1B

Group Members

Name	Enrollment
1. Sadia Saeed	02-134242-005
2. Ayesha Khan	02-134242-091
3. Kumail Abbas Tabassum	02-134242-107

Submitted to:

Ma'am Mehwish Saleem

BAHRIA UNIVERSITY KARACHI CAMPUS

Department of Computer Science

ABSTRACT

The **Online Book Inventory Management System** is designed to revolutionize the way book inventories are managed by providing an efficient, user-friendly platform. This system facilitates essential tasks such as managing book records, tracking sales, handling customer data, and overseeing supply operations. With features like advanced search and filter capabilities, billing integration, and data consistency checks, the system eliminates the inefficiencies and errors of manual inventory management. By automating processes, it ensures real-time tracking of inventory, ultimately enhancing the productivity and profitability of bookstores.

INTRODUCTION:

The Online Book Inventory Management System aims to provide a comprehensive digital solution for managing book inventories, streamlining operations, and improving data accuracy. It enables users to perform various operations such as adding, deleting, and updating book records, searching and filtering books by criteria like title and author, and managing sales and customer records. Additional features include user login functionalities, supply chain management, and sorting books by year or author. By transitioning from manual to automated processes, the system ensures efficient inventory management and supports the smooth functioning of online and offline bookstores.

PROBLEM STATEMENT:

Before this there was book inventory management system in which records were maintained on paper. It became difficult to modify data. In manual system it is quite time consuming to check previous records and there is a chance of errors. There is a possibility that data in different registers have different values which means entries of same data do not match. They can't perform calculation perfectly manually. A system must be needed to keep proper record and handle all inventory efficiently. This aims to provide ease of use for customers and operators through effective record keeping and efficient operations.

METHODOLOGY:

1. Planning

- Define the system's objectives, core functionalities, and user requirements, focusing on inventory tracking, sales management, and customer data handling.
- Choose appropriate language such as C++ for implementation, and plan the system's architecture and user interface design.
- Develop a timeline for project development phases including design, implementation, and testing.

2. Data Collection

- Compile data for books, including title, author, genre, publication year, ISBN, price, and quantity.
- Gather information on customer records, purchase history, and supplier details.

3. System Design

- Create a design for the system with separate components for inventory management, sales processing, and user authentication.
- Develop database for storing book details, transaction records, and user information.
- Design a user-friendly interface for easy interaction.

4. Implementation

- Write the code using C++ to implement the designed modules, ensuring proper handling of CRUD operations (Create, Read, Update, Delete) for book records.
- Integrate features like book search and filter, inventory alerts, and billing functionalities.
- Connect the application to a relational database for efficient data storage and retrieval.

5. Testing

- Conduct unit testing for individual modules to ensure they work as expected.
- Perform integration testing to verify the seamless interaction between various components.
- Carry out user acceptance testing to gather feedback and refine the system based on user needs.

6. Deployment and Training

- Develop the system on the target platform for real-world use.
- Train bookstore staff on using the system, including data entry, transaction processing, and report generation.

7. Monitoring and Maintenance

- Regularly monitor the system for performance issues.
- Implement updates and enhancements based on evolving requirements and user feedback.

PROJECT SCOPE:

Implementing the core functionality of the inventory system using programming languages like C++. It employs software engineering principles for efficient code design, modularization, and maintainability, designing and managing a database to store information about books and the inventory. It manages a collection of books with details including title, author, genre, price, and quantity, Sales management, customer (Records, Purchase history), supplies management, Search and filter(genre/author) (Name, author, ISBN) , loyalty program, its billing and perform CRUD operations (Create, Read, Update, Delete) on book records and provide an intuitive interface for bookstore staff to interact with the inventory.

CODE:

```
Miscellaneous Files (Global Scope)
1  #include <iostream>
2  #include <string>
3  #include <vector>
4  #include <fstream>
5  #include <iomanip>
6  #include <algorithm>
7  #include <cctype>
8  using namespace std;
9
10 struct book
11 {
12     int Id;
13     string Title;
14     string Author;
15     int publicationYear;
16     float Price;
17     int Stock;
18 };
19
20 struct Supplier
21 {
22     int Id;
23     string Name;
24     string Contact;
25 };
26
27 struct customer
28 {
29     string CustomerName;
30     long long CustomerContact;
31     int CustomerID;
32 };
93 % No issues found
```

```
Miscellaneous Files (Global Scope)
33
34 struct Bill
35 {
36     int BillID;
37     string CustomerName;
38     string BookTitle;
39     int Quantity;
40     float PricePerBook;
41     float TotalCost;
42     string PaymentMethod;
43 };
44
45 vector<book> books;
46 vector<Supplier> suppliers;
47 vector<customer> customers;
48 vector<Bill> bill;
49
50 void saveBooks()
51 {
52     ofstream myfile("Booksdata.txt");
53     if (myfile.is_open())
54     {
55         for (int i = 0; i < books.size(); i++)
56         {
57             myfile << books[i].Id << "\n";
58             myfile << books[i].Title << "\n";
59             myfile << books[i].Author << "\n";
60             myfile << books[i].publicationYear << "\n";
61             myfile << books[i].Price << "\n";
62             myfile << books[i].Stock << "\n";
63         }
64         myfile.close();
93 % No issues found
```

```
project code.cpp  X
Miscellaneous Files (Global Scope)
65 }
66 else
67 {
68     cout << "Error saving data to file.\n";
69 }
70 }
71 void loadBooks()
72 {
73     ifstream myfile("Booksdata.txt");
74     if (myfile.is_open())
75     {
76         book newBook;
77         while (myfile >> newBook.Id)
78         {
79             myfile.ignore();
80             getline(myfile, newBook.Title);
81             getline(myfile, newBook.Author);
82             myfile >> newBook.publicationYear;
83             myfile >> newBook.Price;
84             myfile >> newBook.Stock;
85             myfile.ignore();
86             books.push_back(newBook);
87         }
88         myfile.close();
89     }
90     else
91     {
92         cout << "No book data found. Starting with an empty inventory.\n";
93     }
94 }
95 void addBook()
96 {
```

93 % No issues found

Ready

```
project code.cpp  X
Miscellaneous Files (Global Scope)
97     book newBook;
98     cout << "Enter Book ID: ";
99     cin >> newBook.Id;
100     cin.ignore();
101     cout << "Enter Book Title: ";
102     getline(cin, newBook.Title);
103     cout << "Enter Book Author: ";
104     getline(cin, newBook.Author);
105     cout << "Enter Publication Year: ";
106     cin >> newBook.publicationYear;
107     cout << "Enter Book Price: ";
108     cin >> newBook.Price;
109     cout << "Enter Book Stock: ";
110     cin >> newBook.Stock;
111
112     books.push_back(newBook);
113     saveBooks();
114
115     cout << "Book added successfully!\n";
116 }
117
118 void displayBooks()
119 {
120     if (books.empty())
121     {
122         cout << "No books available.\n";
123         return;
124     }
125
126     cout << "Books in the system:\n";
127     for (int i = 0; i < books.size(); i++) {
128         cout << "ID: " << books[i].Id << "\n";
```

93 % No issues found

Ready

```
project code.cpp X
Miscellaneous Files (Global Scope) searchB

129     cout << "Title: " << books[i].Title << "\n";
130     cout << "Author: " << books[i].Author << "\n";
131     cout << "Publication Year: " << books[i].publicationYear << "\n";
132     cout << "Price: $" << books[i].Price << "\n";
133     cout << "Stock: " << books[i].Stock << "\n";
134     cout << "-----\n";
135 }
136 }
137 void deleteBook()
138 {
139     int id;
140     cout << "Enter Book ID to delete: ";
141     cin >> id;
142
143     for (int i = 0; i < books.size(); i++)
144     {
145         if (books[i].Id == id)
146         {
147             books.erase(books.begin() + i);
148             saveBooks();
149             cout << "Book deleted successfully!\n";
150             return;
151         }
152     }
153     cout << "Book not found!\n";
154 }
155
156 void searchBook()
157 {
158     string keyword;
159     cout << "Enter Title or Author to search: ";
160     cin.ignore();
```

```
project code.cpp X
Miscellaneous Files (Global Scope) upo

161     getline(cin, keyword);
162     transform(keyword.begin(), keyword.end(), keyword.begin(), ::tolower);
163
164     bool found = false;
165     for (int i = 0; i < books.size(); i++)
166     {
167         string title = books[i].Title;
168         string author = books[i].Author;
169         transform(title.begin(), title.end(), title.begin(), ::tolower);
170         transform(author.begin(), author.end(), author.begin(), ::tolower);
171
172         if (title.find(keyword) != string::npos || author.find(keyword) != string::npos)
173         {
174             cout << "\nBook Found:\n";
175             cout << "ID: " << books[i].Id << "\nTitle: " << books[i].Title
176                 << "\nAuthor: " << books[i].Author << "\nPrice: $" << books[i].Price
177                 << "\nPublication Year: " << books[i].publicationYear << "\n";
178             found = true;
179         }
180     }
181
182     if (!found)
183     {
184         cout << "No book found with the given keyword.\n";
185     }
186 }
187
188 void updateBook()
189 {
190     int id;
191     cout << "Enter Book ID to update: ";
192
```

```
project code.cpp [X]
Miscellaneous Files (Global Scope)

193     cin >> id;
194
195     for (int i = 0; i < books.size(); i++)
196     {
197         if (books[i].Id == id)
198         {
199             cin.ignore();
200             cout << "Enter new Title: ";
201             getline(cin, books[i].Title);
202             cout << "Enter new Author: ";
203             getline(cin, books[i].Author);
204             cout << "Enter new Price: ";
205             cin >> books[i].Price;
206             cout << "Enter new Stock: ";
207             cin >> books[i].Stock;
208
209             saveBooks();
210             cout << "Book updated successfully!\n";
211             return;
212         }
213     }
214     cout << "Book not found!\n";
215 }
216
217 void saveSuppliers()
218 {
219     ofstream myfile("Suppliersdata.txt");
220     if (myfile.is_open())
221     {
222         for (int i = 0; i < suppliers.size(); i++)
223         {
224             myfile << suppliers[i].Id << "\n";
```

```
project code.cpp [X]
Miscellaneous Files (Global Scope)

225             myfile << suppliers[i].Name << "\n";
226             myfile << suppliers[i].Contact << "\n";
227         }
228         myfile.close();
229     }
230     else
231     {
232         cout << "Error saving data to file.\n";
233     }
234 }
235
236 void addSupplier()
237 {
238     Supplier s;
239     cout << "Enter Supplier ID: ";
240     cin >> s.Id;
241     cin.ignore();
242     cout << "Enter Supplier Name: ";
243     getline(cin, s.Name);
244     cout << "Enter Supplier Contact: ";
245     getline(cin, s.Contact);
246
247     suppliers.push_back(s);
248     saveSuppliers();
249
250     cout << "Supplier added successfully!\n";
251 }
252
253 void displaySuppliers()
254 {
255     if (suppliers.empty())
256     {
```

```
project code.cpp  X
Miscellaneous Files (Global Scope) filterBooksByYear()
257     cout << "No suppliers available.\n";
258     return;
259 }
260
261     cout << "Displaying Suppliers:\n";
262     cout << setw(10) << "ID" << setw(30) << "Name" << setw(20) << "Contact" << endl;
263
264     for (const auto& supplier : suppliers)
265     {
266         cout << setw(10) << supplier.Id << setw(30) << supplier.Name << setw(20) << supplier.Contact << endl;
267     }
268 }
269
270 void sortBooks()
271 {
272     if (books.empty()) {
273         cout << "No books available to sort.\n";
274         return;
275     }
276     sort(books.begin(), books.end(), [](const book& a, const book& b) {
277         return a.Title < b.Title;
278     });
279     cout << "Books sorted by title successfully.\n";
280 }
281
282 void filterBooksByYear()
283 {
284     if (books.empty())
285     {
286         cout << "No books available to filter.\n";
287         return;
288     }
}

93 % No issues found
```

```
project code.cpp  X
Miscellaneous Files (Global Scope) saveBillID(int b
289     int year;
290     cout << "Enter year to filter books: ";
291     cin >> year;
292     cout << "Books published in " << year << ":\n";
293     cout << setw(10) << "ID" << setw(30) << "Title" << setw(20) << "Author" << endl;
294     for (const auto& book : books)
295     {
296         if (book.publicationYear == year)
297         {
298             cout << setw(10) << book.Id << setw(30) << book.Title << setw(20) << book.Author << endl;
299         }
300     }
301 }
302 int loadBillID()
303 {
304     ifstream myfile("lastBillID.txt");
305     int lastBillID = 0; // Default starting ID (if file doesn't exist)
306     if (myfile.is_open())
307     {
308         myfile >> lastBillID;
309         myfile.close();
310     }
311     return lastBillID;
312 }
313
314 void saveBillID(int billID)
315 {
316     ofstream myfile("lastBillID.txt");
317     if (myfile.is_open())
318     {
319         myfile << billID;
320         myfile.close();
321     }
}

93 % No issues found
```



```
project code.cpp  X
Miscellaneous Files (Global Scope) generate_bill(string & title,
321     }
322     else
323     {
324         cout << "Error saving Bill ID.\n";
325     }
326 }
327
328 void generate_bill(string& title, string& customer_name, float& price, int& quantity, string& author)
329 {
330     static int billCounter = loadBillID();
331
332     Bill newBill;
333     newBill.BillID = billCounter++;
334     float totalCost = quantity * price;
335     newBill.CustomerName = customer_name;
336     newBill.BookTitle = title;
337     newBill.Quantity = quantity;
338     newBill.PricePerBook = price;
339     newBill.TotalCost = totalCost;
340
341     int paymentChoice;
342     string paymentMethod;
343
344     cout << "\n--- Billing Details ---\n";
345     cout << "Book Title: " << title << "\n";
346     cout << "Author: " << author << "\n";
347     cout << "Quantity: " << quantity << "\n";
348     cout << "Price per book: $" << price << "\n";
349     cout << "Total Cost: $" << totalCost << "\n";
350     cout << "-----\n";
351     cout << "\nChoose a payment method:\n";
352     cout << "1. Credit/Debit Card\n";
93 % No issues found
Ready
```

```
project code.cpp  X
Miscellaneous Files (Global Scope)
353     cout << "2. Easy Paisa\n";
354     cout << "3. Digital Wallet\n";
355     cout << "Enter your choice: ";
356     cin >> paymentChoice;
357
358     switch (paymentChoice)
359     {
360     case 1:
361         cout << "Processing payment via Credit/Debit Card...\n";
362         cout << "Payment successful!\n";
363         paymentMethod = "Credit/Debit Card";
364         break;
365     case 2:
366         cout << "Processing payment via Easy Paisa...\n";
367         cout << "Payment successful!\n";
368         paymentMethod = "Easy Paisa";
369         break;
370     case 3:
371         cout << "Processing payment via Digital Wallet...\n";
372         cout << "Payment successful!\n";
373         paymentMethod = "Digital Wallet";
374         break;
375     default:
376         cout << "Invalid payment method selected. Transaction canceled.\n";
377         return;
378     }
379
380     newBill.PaymentMethod = paymentMethod;
381     bill.push_back(newBill);
382
383     saveBillID(billCounter);
384 }
93 % No issues found
Ready
```

```
project code.cpp  X
Miscellaneous Files (Global Scope) purchaseBook()
385 void purchaseBook()
386 {
387     bool found = false;
388     string Book, customer_name;
389     int quantity;
390     cout << "Enter full name: " << endl;
391     getline(cin, customer_name);
392     cout << "Enter the book you want to purchase: " << endl;
393     getline(cin, Book);
394     cout << "Enter the quantity you want to purchase: ";
395     cin >> quantity;
396     cin.ignore();
397
398     for (int i = 0; i < books.size(); i++)
399     {
400         if (books[i].Title == Book)
401         {
402             found = true;
403             if (books[i].Stock >= quantity)
404             {
405                 cout << "Book purchased successfully: " << books[i].Title << " by " << books[i].Author << "\n";
406                 books[i].Stock -= quantity;
407                 generate_bill(books[i].Title, customer_name, books[i].Price, quantity, books[i].Author);
408             }
409             else if (books[i].Stock == 0)
410             {
411                 cout << "Sorry, the book \"" << books[i].Title << "\" is out of stock.\n";
412             }
413             else
414             {
415                 cout << "Sorry, only " << books[i].Stock << " copies of \"" << books[i].Title << "\" are available.\n";
416             }
417         }
418     }
419 }
93 % No issues found
```

```
project code.cpp  X
Miscellaneous Files (Global Scope) addC
417         break;
418     }
419 }
420 if (found == false)
421 {
422     cout << "Sorry, the book \"" << Book << "\" was not found in the library.\n";
423 }
424 }
425
426 void addCustomer()
427 {
428     customer Customer;
429     cout << "Enter Customer Name: ";
430     getline(cin, Customer.CustomerName);
431     string contact;
432     cout << "Enter Customer Contact (11 digits): ";
433     cin.ignore();
434     getline(cin, contact);
435     while (contact.length() != 11 || contact.find_first_not_of("0123456789") != string::npos)
436     {
437         cout << "Invalid contact number. Please enter a 11-digit number: ";
438         getline(cin, contact);
439     }
440     Customer.CustomerContact = stoll(contact);
441     bool idExists;
442     do
443     {
444         idExists = false;
445         cout << "Enter Customer ID: ";
446         cin >> Customer.CustomerID;
447         for (int i = 0; i < customers.size(); i++)
448         {
449             if (customers[i].CustomerID == Customer.CustomerID)
450             {
451                 idExists = true;
452                 cout << "Customer ID already exists. Please enter a unique ID: ";
453                 cin >> Customer.CustomerID;
454             }
455         }
456     } while (idExists);
457 }
93 % No issues found
Ready
```

```
Miscellaneous Files (Global Scope)
449         if (customers[i].CustomerID == Customer.CustomerID)
450         {
451             cout << "Customer ID already exists. Please enter a unique ID.\n";
452             idExists = true;
453             break;
454         }
455     } while (idExists);
456
457     customers.push_back(Customer);
458     cout << "Customer added successfully!\n";
459 }
460
461 void display_customers()
462 {
463     if (customers.empty())
464     {
465         cout << "\nNo customers found.\n";
466         return;
467     }
468     cout << "\nCustomers Information: \n";
469     for (int i = 0; i < customers.size(); i++)
470     {
471         cout << "Id: " << customers[i].CustomerID
472              << " Name: " << customers[i].CustomerName
473              << " Contact: " << customers[i].CustomerContact
474              << "\n";
475     }
476 }
477
478 void filter_by_author()
479 {
480     string authorName;
```

93 % No issues found

```
project code.cpp Miscellaneous Files (Global Scope)
481     cout << "enter the name of author: " << endl;
482     getline(cin, authorName);
483     cout << "\nBooks by \"" << authorName << ":\n";
484     bool found = false;
485     for (int i = 0; i < books.size(); i++)
486     {
487         if (authorName == books[i].Author)
488         {
489             found = true;
490             cout << "Book ID: " << books[i].Id << "\n";
491             cout << "Title: " << books[i].Title << "\n";
492             cout << "Year of Publication: " << books[i].publicationYear << "\n";
493             cout << "Price: $" << books[i].Price << "\n";
494             cout << "Stock: " << books[i].Stock << "\n";
495             cout << "-----\n";
496         }
497     }
498     if (!found)
499     {
500         cout << "No books found by author \"" << authorName << ":\n";
501     }
502 }
503
504 int main()
505 {
506     loadBooks();
507     int choice;
508     do {
509         cout << "\n--- Online Book Inventory Management System ---\n";
510         cout << "1. Add Book\n";
511         cout << "2. Display Book\n";
512         cout << "3. Delete Book\n";
```

93 % No issues found

Ready

```
project code.cpp  [X]
Miscellaneous Files  (Global Scope)

513     cout << "4. Search Book\n";
514     cout << "5. Update Book\n";
515     cout << "6. Filter Books by Year\n";
516     cout << "7. Filter Book by Author\n";
517     cout << "8. Sort Books \n";
518     cout << "9. Purcahse Book\n";
519     cout << "10.Add Supplier \n";
520     cout << "11.Display Suppliers\n";
521     cout << "12.Add Customer\n";
522     cout << "13.Display Customer\n";
523     cout << "14.Exit \n";
524
525     cout << "Enter your choice: ";
526     cin >> choice;
527     cin.ignore();
528
529     switch (choice)
530     {
531     case 1:
532         addBook();
533         break;
534     case 2:
535         displayBooks();
536         break;
537     case 3:
538         deleteBook();
539         break;
540     case 4:
541         searchBook();
542         break;
543     case 5:
544         updateBook();
545
93 %  No issues found
```

```
project code.cpp  [X]
Miscellaneous Files  (Global Scope)

545         break;
546     case 6:
547         filterBooksByYear();
548         break;
549     case 7:
550         filter_by_author();
551         break;
552     case 8:
553         sortBooks();
554         break;
555     case 9:
556         purchaseBook();
557         break;
558     case 10:
559         addSupplier();
560         break;
561     case 11:
562         displaySuppliers();
563         break;
564     case 12:
565         addCustomer();
566         break;
567     case 13:
568         display_customers();
569         break;
570     case 14:
571         cout << "Exiting the system. Goodbye!\n";
572         break;
573     default:
574         cout << "Invalid choice. Try again.\n";
575     }
576 } while (choice != 14);

93 %  No issues found
```

OUTPUT:

```
--- Online Book Inventory Management System ---
1. Add Book
2. Display Book
3. Delete Book
4. Search Book
5. Update Book
6. Filter Books by Year
7. Filter Book by Author
8. Sort Books
9. Purchse Book
10.Add Supplier
11.Display Suppliers
12.Add Customer
13.Display Customer
14.Exit
Enter your choice: 1
Enter Book ID: 5
Enter Book Title: The Great Gatsby
Enter Book Author: F. Scott Fitzgerald
Enter Publication Year: 1925
Enter Book Price: 2000
Enter Book Stock: 16
Book added successfully!

--- Online Book Inventory Management System ---
1. Add Book
2. Display Book
3. Delete Book
4. Search Book
5. Update Book
6. Filter Books by Year
7. Filter Book by Author
8. Sort Books
9. Purchse Book
10.Add Supplier
11.Display Suppliers
12.Add Customer
13.Display Customer
14.Exit
Enter your choice: 2
Books in the system:
ID: 1
```

```
ID: 1
Title: welcome home
Author: najwa zebian
Publication Year: 2023
Price: $1000
Stock: 15
-----
ID: 2
Title: harry potter
Author: jk rowlings
Publication Year: 2000
Price: $4000
Stock: 10
-----
ID: 3
Title: dream
Author: oscar wild
Publication Year: 2002
Price: $1500
Stock: 8
-----
ID: 4
Title: david copperfield
Author: charles dickens
Publication Year: 2003
Price: $2000
Stock: 6
-----
ID: 5
Title: The Great Gatsby
Author: F. Scott Fitzgerald
Publication Year: 1925
Price: $2000
Stock: 16
-----

--- Online Book Inventory Management System ---
1. Add Book
2. Display Book
3. Delete Book
4. Search Book
5. Update Book
6. Filter Books by Year
```

```
7. Filter Book by Author
8. Sort Books
9. Purchse Book
10.Add Supplier
11.Display Suppliers
12.Add Customer
13.Display Customer
14.Exit
Enter your choice: 3
Enter Book ID to delete: 3
Book deleted successfully!
```

--- Online Book Inventory Management System ---

```
1. Add Book
2. Display Book
3. Delete Book
4. Search Book
5. Update Book
6. Filter Books by Year
7. Filter Book by Author
8. Sort Books
9. Purchse Book
10.Add Supplier
11.Display Suppliers
12.Add Customer
13.Display Customer
14.Exit
```

Enter your choice: 2

Books in the system:

ID: 1

Title: welcome home

Author: najwa zebian

Publication Year: 2023

Price: \$1000

Stock: 15

ID: 2

Title: harry potter

Author: jk rowlings

Publication Year: 2000

Price: \$4000

Stock: 10

ID: 4

Title: david copperfield

Author: charles dickens

Publication Year: 2003

Price: \$2000

Stock: 6

ID: 5

Title: The Great Gatsby

Author: F. Scott Fitzgerald

Publication Year: 1925

Price: \$2000

Stock: 16

--- Online Book Inventory Management System ---

```
1. Add Book
2. Display Book
3. Delete Book
4. Search Book
5. Update Book
6. Filter Books by Year
7. Filter Book by Author
8. Sort Books
9. Purchse Book
10.Add Supplier
11.Display Suppliers
12.Add Customer
13.Display Customer
14.Exit
```

Enter your choice: 4

Enter Title or Author to search: harry potter

Book Found:

ID: 2

Title: harry potter

Author: jk rowlings

Price: \$4000

Publication Year: 2000

--- Online Book Inventory Management System ---

```
1. Add Book
2. Display Book
```



```

3. Delete Book
4. Search Book
5. Update Book
6. Filter Books by Year
7. Filter Book by Author
8. Sort Books
9. Purcahse Book
10.Add Supplier
11.Display Suppliers
12.Add Customer
13.Display Customer
14.Exit
Enter your choice: 5
Enter Book ID to update: 2
Enter new Title: harry potter part 1
Enter new Author: jk rowlings
Enter new Price: 4500
Enter new Stock: 20
Book updated successfully!

--- Online Book Inventory Management System ---
1. Add Book
2. Display Book
3. Delete Book
4. Search Book
5. Update Book
6. Filter Books by Year
7. Filter Book by Author
8. Sort Books
9. Purcahse Book
10.Add Supplier
11.Display Suppliers
12.Add Customer
13.Display Customer
14.Exit
Enter your choice: 6
Enter year to filter books: 2000
Books published in 2000:
      ID              Title              Author
      2              harry potter part 1    jk rowlings

--- Online Book Inventory Management System ---
1. Add Book

```

```

2. Display Book
3. Delete Book
4. Search Book
5. Update Book
6. Filter Books by Year
7. Filter Book by Author
8. Sort Books
9. Purcahse Book
10.Add Supplier
11.Display Suppliers
12.Add Customer
13.Display Customer
14.Exit
Enter your choice: 7
enter the name of author:
najwa zebian

Books by "najwa zebian":
Book ID: 1
Title: welcome home
Year of Publication: 2023
Price: $1000
Stock: 15
-----

```

```
--- Online Book Inventory Management System ---
1. Add Book
2. Display Book
3. Delete Book
4. Search Book
5. Update Book
6. Filter Books by Year
7. Filter Book by Author
8. Sort Books
9. Purcahse Book
10.Add Supplier
11.Display Suppliers
12.Add Customer
13.Display Customer
14.Exit
Enter your choice: 9
Enter full name:
ayesha khan
Enter the book you want to purchase:
welcome home
Enter the quantity you want to purchase: 1
Book purchased successfully: welcome home by najwa zebian
```

```
--- Billing Details ---
Book Title: welcome home
Author: najwa zebian
Quantity: 1
Price per book: $1000
Total Cost: $1000
-----
```

```
Choose a payment method:
1. Credit/Debit Card
2. Easy Paisa
3. Digital Wallet
Enter your choice: 1
Processing payment via Credit/Debit Card...
Payment successful!
```

```
--- Online Book Inventory Management System ---
1. Add Book
2. Display Book
```

```
7. Filter Book by Author
8. Sort Books
9. Purcahse Book
10.Add Supplier
11.Display Suppliers
12.Add Customer
13.Display Customer
14.Exit
Enter your choice: 10
Enter Supplier ID: 201
Enter Supplier Name: asad mehmood
Enter Supplier Contact: 03352254684
Supplier added successfully!
```

```
--- Online Book Inventory Management System ---
```

```
1. Add Book
2. Display Book
3. Delete Book
4. Search Book
5. Update Book
6. Filter Books by Year
7. Filter Book by Author
8. Sort Books
9. Purcahse Book
10.Add Supplier
11.Display Suppliers
12.Add Customer
13.Display Customer
14.Exit
```

```
Enter your choice: 11
Displaying Suppliers:
```

ID	Name	Contact
201	asad mehmood	03352254684

```
--- Online Book Inventory Management System ---
```

```
1. Add Book
2. Display Book
3. Delete Book
4. Search Book
5. Update Book
6. Filter Books by Year
7. Filter Book by Author
8. Sort Books
```



```
9. Purcahse Book
10.Add Supplier
11.Display Suppliers
12.Add Customer
13.Display Customer
14.Exit
Enter your choice: 12
Enter Customer Name: aiman masood
Enter Customer Contact (11 digits): 03362254789
Invalid contact number. Please enter a 11-digit number: 03362542883
Enter Customer ID: 143
Customer added successfully!
```

--- Online Book Inventory Management System ---

```
1. Add Book
2. Display Book
3. Delete Book
4. Search Book
5. Update Book
6. Filter Books by Year
7. Filter Book by Author
8. Sort Books
9. Purcahse Book
10.Add Supplier
11.Display Suppliers
12.Add Customer
13.Display Customer
14.Exit
Enter your choice: 13
```

Customers Information:

Id: 143 Name: aiman masood Contact: 3362542883

--- Online Book Inventory Management System ---

```
1. Add Book
2. Display Book
3. Delete Book
4. Search Book
5. Update Book
6. Filter Books by Year
7. Filter Book by Author
8. Sort Books
9. Purcahse Book
```

```
10.Add Supplier
11.Display Suppliers
12.Add Customer
13.Display Customer
14.Exit
Enter your choice: 14
Exiting the system. Goodbye!
```

FUTURE DEVELOPMENT:

The Online Book Inventory Management System can be expanded with the following enhancements:

1. **Mobile App Integration**
 - Develop a mobile application for accessing inventory and sales data on the go.
2. **AI-Powered Recommendations**
 - Use AI to suggest books to customers based on their purchase history or preferences.
3. **Integration with E-commerce Platforms**
 - Enable seamless integration with popular e-commerce platforms for online sales and inventory synchronization.
4. **Multi-Language and Multi-Currency Support**
 - Add support for multiple languages and currencies to cater to a global audience.
5. **Barcode and QR Code Integration**
 - Simplify inventory management by enabling barcode or QR code scanning for book details and transactions.
6. **Automated Email and SMS Notifications**
 - Notify customers about new arrivals, promotions, and loyalty rewards through Email or SMS.

CONCLUSION:

In conclusion, business that deals with books needs a online book inventory system as a management tool. The technique reduces the possibility of overstocking while assisting in ensuring that there is always a sufficient supply of books. Improved customer happiness, more sales, and lower inventory management costs are all results of efficient inventory management. Lean inventory management, forecasting techniques, and the use of inventory management software are just a few of the approaches to managing book inventory that are covered in depth in the literature on book inventory management. The several publications cover a variety of subjects, including planning for material requirements, forecasting, and inventory management rules. Organizations can use these tools to determine the best method for managing their book inventory. The book inventory system needs to be continually examined and enhanced if it is to be effective. This entails routinely examining inventory data to spot trends and patterns, modifying inventory levels and reordering guidelines as necessary, and making sure the inventory management system is operating at its best. In conclusion, an efficient book inventory system can assist businesses in streamlining their processes and giving customers better service. Businesses may optimize their book inventory, streamline processes, and cut costs by putting best practices in inventory management into practice and employing the right inventory management software.