

VISVESVARAYA TECHNOLOGICAL UNIVERSITY

Jnana Sangama, Belagavi-590018



A Database Management System Mini Project Report on

“BOOKSTORE MANAGEMENT SYSTEM”

Submitted in Partial fulfillment of the Requirements for the IV Semester of the Degree of

Bachelor of Engineering in
Computer Science & Engineering

By

MUDUNURI SHRILEKHA (1CR23CS112)

Under the Guidance of

Prof. Manjula Subramaniam,
Assistant Professor, Dept of CSE



DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

CMR INSTITUTE OF TECHNOLOGY

Affiliated to VTU, Approved by AICTE, Accredited by NBA and NAAC with “A++” Grade

ITPL MAIN ROAD, BROOKFIELD, BENGALURU-560037, KARNATAKA, INDIA

2024-25

CMR INSTITUTE OF TECHNOLOGY

Affiliated to VTU, Approved by AICTE, Accredited by NBA and NAAC with “A++” Grade

ITPL MAIN ROAD, BROOKFIELD, BENGALURU-560037, KARNATAKA, INDIA

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING



CERTIFICATE

This is to certify that the Database Management System Project work entitled "**Bookstore Management System**" has been carried out by **Mudunuri Shrilekha(1CR23CS112)**, bonafide student of CMR Institute of Technology, Bengaluru in partial fulfillment for the award of the Degree of **Bachelor of Engineering in Computer Science and Engineering** of the Visvesvaraya Technological University, Belagavi during the year **2024-2025**. It is certified that all corrections/suggestions indicated for the Internal Assessment have been incorporated in the report deposited in the departmental library. This Database Management System Project report has been approved as it satisfies the academic requirements in respect of project work prescribed for the said Degree.

Signature of Guide

Prof. Manjula Subramaniam

Assistant Professor

Dept. of CSE, CMRIT

Signature of HOD

Dr. Kesavamoorthy R

Professor & HoD

Dept. of CSE, CMRIT

DECLARATION

I, a student of IV semester from Department of Computer Science and Engineering, CMR Institute of Technology, Bangalore declare that the project work entitled "**Bookstore Management System**" has been successfully completed under the guidance of Mrs. Manjula Subramaniam (Assistant Professor) and Mr. Prateek Kumar (Teaching Assistant), both from Dept. of Computer Science and Engineering, CMR Institute of technology, Bengaluru. This project work is submitted in partial fulfillment of the requirements for the award of the Degree of Bachelor of Engineering in Computer Science and Engineering during the academic year 2024-2025. The matter embodied in the project report has not been submitted previously by anybody for the award of any degree or diploma to any university.

Place: Bangalore

Date: 28/5/2025

Student details:

MUDUNURI SHRILEKHA (1CR23CS112)	
--	--

ABSTRACT

Bookstore Management System is a comprehensive desktop application developed using Python and Tkinter, with MySQL for database management. It features staff authentication, book and customer management, order processing, and reporting. The system includes role-based access control, with managers having additional privileges for staff management and reports. Key functionalities include inventory tracking, sales processing, and logging of all book-related actions. The application ensures data integrity through validation checks and transaction management. Its user-friendly interface and robust backend make it suitable for efficient bookstore operations.

Keywords: Bookstore Management, Tkinter, MySQL, Inventory System, Role-Based Access

ACKNOWLEDGEMENT

I take this opportunity to express my sincere gratitude and respect to **CMR Institute of Technology, Bengaluru** for providing me a platform to pursue my studies and carry out the Database Management System Project.

It gives me an immense pleasure to express my deep sense of gratitude to **Dr. Sanjay Jain**, Principal, CMRIT, Bengaluru, for his constant encouragement.

I would like to extend my sincere gratitude to **Dr. Kesavamoorthy R**, HOD, Department of Computer Science and Engineering, CMRIT, Bengaluru, who has been a constant support and encouragement throughout the course of this project.

I would like to thank my guides, Mrs. Manjula Subramaniam (Assistant Professor) and Mr. Prateek Kumar (Teaching Assistant), Department of Computer Science and Engineering, for the valuable guidance throughout the tenure of the project work.

I would also like to thank all the faculty members of Department of Computer Science and Engineering who directly or indirectly encouraged me.

Finally, I thank my parents and friends for all the moral support they have given me during the completion of this work.

TABLE OF CONTENTS

Contents	Page No.
Certificate	ii
Declaration	iii
Abstract	iv
Acknowledgement	v
Table of contents	vi
List of Figures	vii
1. Introduction	1-2
1.1 Objectives	
1.2 Scope of the Project	
2. System Requirements	3-4
2.1 Front-End Tools	
2.2 Back-End Tools	
3. Design	5-7
3.1 Schema Diagram	
3.2 ER Diagram	
3.3 Database Tables	
4. Implementation	8-10
4.1 Installation Steps for Execution of the Project	
4.2 Front-End Code Snippets	
4.3 Back-End Code Snippets	
4.4 Integration Highlights	
5. Interpretation of Result	11-14
5.1 Result Snapshots	
6. Conclusion and Future Scope	15
6.1 Future Scope	
References	16

LIST OF FIGURES

	Page No.
Fig 3.1 Schema Diagram	5
Fig 3.2 ER Diagram	6
Fig 5.1 (a) Login Page for Admin	11
(b) Login Page for Other Staff	11
Fig 5.2 CRUD operations on Books Table	
(a) Adding books	12
(b) Updating a record	12
(c) Deletion	13
(d) Viewing	13
Fig 5.3 Order Processing Dashboard	13-14

CHAPTER 1

INTRODUCTION

The **Bookstore Management System** is a software solution designed to streamline the operations of a bookstore by automating inventory management, sales processing, customer relations, and staff administration. In today's digital era, manual record-keeping systems are inefficient and prone to errors. This project addresses these challenges by providing a secure, user-friendly, and scalable system that enhances productivity and ensures accurate data management.

The motivation behind this project stems from the need for small and medium-sized bookstores to adopt modern technology for efficient operations. Traditional methods of managing books, customers, and sales are time-consuming and lack real-time tracking capabilities.

Bookstores, whether physical or online, require robust systems to handle inventory, sales, and customer data. Manual processes often lead to discrepancies in stock records, delayed order processing, and difficulties in generating reports. The **Bookstore Management System** leverages database technology and a graphical user interface (GUI) to provide a seamless experience for staff and administrators.

1.1 Objectives

The primary objectives of this project are:

- To automate bookstore operations, including book inventory, customer management, and order processing.
- To implement role-based access control, ensuring secure and restricted access for staff members.
- To provide real-time tracking of sales, stock levels, and financial reports.
- To enhance data accuracy and reduce manual errors through validation checks and database constraints.
- To offer an intuitive and user-friendly interface for efficient system navigation.

1.2 Scope of the project

This system is designed for bookstore administrators, clerks, and librarians, allowing them to:

- Manage Books: Add, update, delete, and track book inventory.
- Handle Customers: Register customers, track memberships, and manage purchase history.
- Process Orders: Create, modify, and complete sales transactions.
- Generate Reports: Analyze sales trends, inventory status, and financial summaries.
- Administer Staff: Add and manage employee accounts with different access levels.

By integrating these features, the Bookstore Management System serves as a comprehensive tool for modern bookstore operations, ensuring efficiency, security, and scalability.

CHAPTER 2

SYSTEM REQUIREMENTS

The Bookstore Management System is built using a combination of front-end and back-end technologies to ensure a responsive, secure, and efficient application. The system requirements have been carefully selected to provide a seamless user experience while maintaining robust database management and functionality. This chapter discusses the tools and technologies used in the development of this project, along with their benefits and relevance.

2.1 Front-End Tools

The front-end of the application is developed using Python with the Tkinter library, which provides a simple yet powerful way to create graphical user interfaces (GUIs). Tkinter is widely used for desktop applications due to its cross-platform compatibility and ease of integration with Python.

Key benefits of using Tkinter in this project include:

- User-Friendly Interface: Tkinter's widgets (buttons, labels, treeviews, etc.) help create an intuitive and interactive interface for bookstore staff.
- Customization: The ttk module allows for modern styling and theming, improving the visual appeal of the application.
- Event-Driven Programming: Tkinter supports event handling, making it easy to implement user interactions such as button clicks and form submissions.

Additionally, Tkinter's simplicity ensures faster development while maintaining functionality, making it ideal for small to medium-sized business applications like this bookstore management system.

2.2 Back-End Tools

The back-end of the system relies on MySQL, a robust relational database management system (RDBMS), for storing and managing all bookstore-related data.

MySQL provides a structured and efficient way to handle large datasets while ensuring data integrity and security.

Key advantages of using MySQL in this project include:

- Data Security: User authentication, foreign key constraints, and transaction management ensure that data remains consistent and secure.
- Scalability: MySQL efficiently handles growing amounts of data, making it suitable for expanding bookstore operations.
- Query Efficiency: Complex operations such as sales reports, inventory checks, and customer records are optimized through SQL queries.

The integration of Python (with mysql-connector) allows seamless communication between the front-end and the database, ensuring real-time updates and smooth data retrieval.

CHAPTER 3

DESIGN

The design of the Bookstore Management System follows a structured approach to ensure efficient data organization, seamless functionality, and optimal user experience. This chapter presents the system's architectural blueprint through schema diagrams, entity-relationship models, and detailed database tables. These design elements form the foundation of the application, enabling smooth interactions between users and the database while maintaining data integrity and security.

3.1 Schema Diagram

The schema diagram provides a high-level overview of the database structure, illustrating how different tables relate to one another. In this system, the schema consists of interconnected tables for Staff, Books, Customers, Orders, and Order Items, with clearly defined relationships.

Key features of the schema include:

- Primary and Foreign Key Relationships that maintain referential integrity
- Normalized Tables to minimize data redundancy
- Constraints and Validations to ensure data accuracy

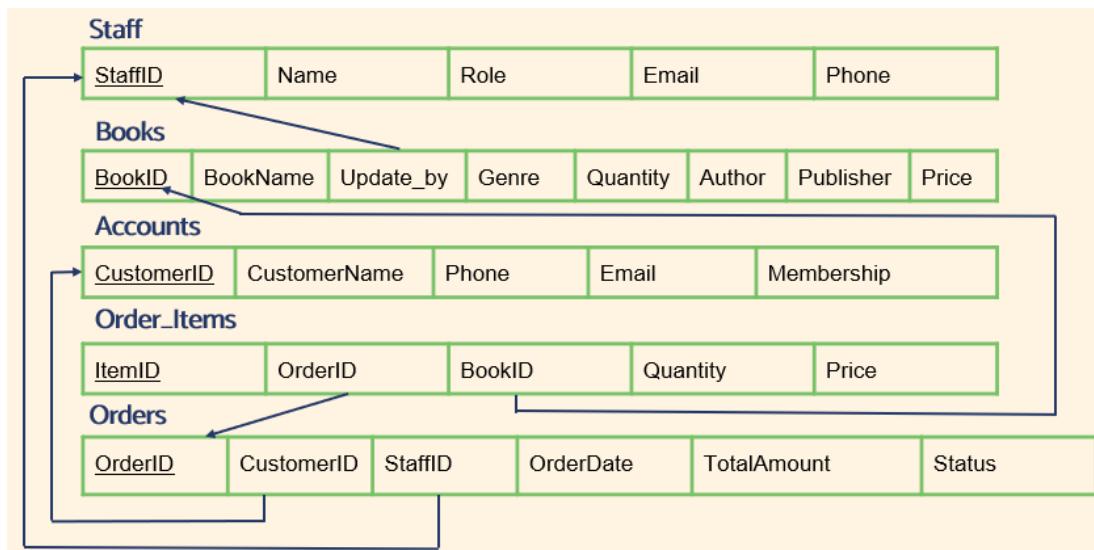


Fig 3.1 Schema Diagram

3.2 ER Diagram

The Entity-Relationship diagram visually represents the logical structure of the database, highlighting entities (as listed), their attributes, and the relationships.

- Staff (Managers, Clerks, Librarians)
- Books (Inventory details)
- Customers (Membership and purchase history)
- Orders (Sales transactions)
- Order Items (Books purchased in each order)

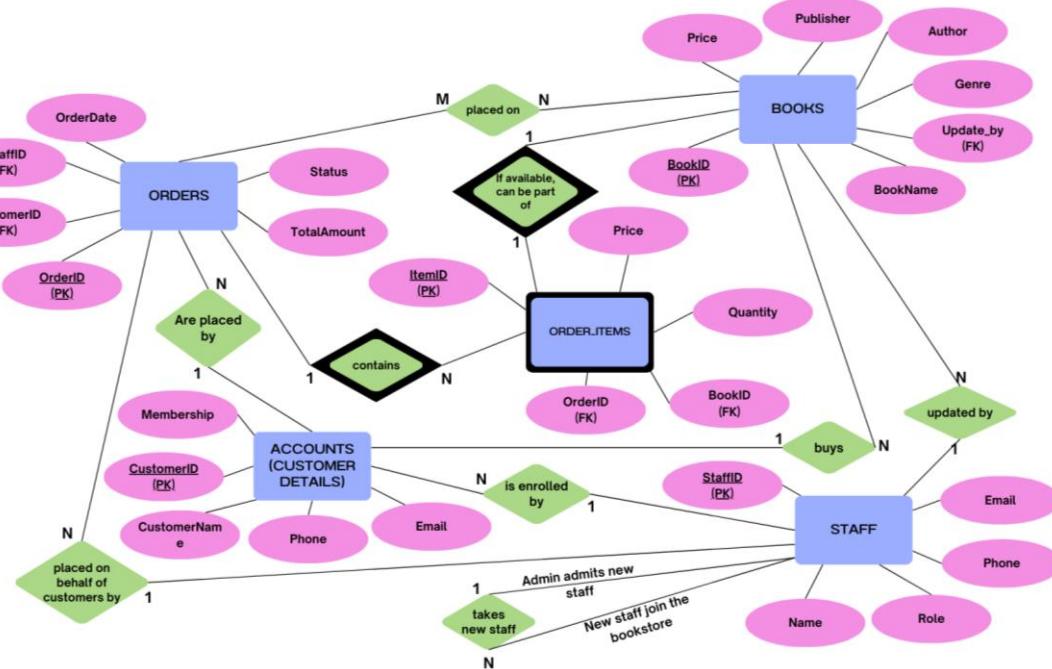


Fig 3.2 ER Diagram

3.3 Database Tables

Staff Table:

StaffID, Name, Role, Email, Phone, HireDate, PasswordHash
[Manages employee accounts with role-based access control]

Books Table:

BookID, BookName, Genre, Quantity, Author, Publisher, Price, Update_by (FK)
[Tracks inventory with real-time stock updates]

Accounts (Customers) Table:

CustomerID, CustomerName, Phone, Email, Membership
[Stores customer details and membership status]

Orders Table:

OrderID, CustomerID (FK), StaffID (FK), OrderDate, TotalAmount, Status
[Records sales transactions and their status (Pending/Completed/Cancelled)]

OrderItems Table:

OrderItemID, OrderID (FK), BookID (FK), Quantity, Price
[Links orders to purchased books and quantities]

BookLog Table:

LogID (PK), BookID (FK), Action (INSERT/UPDATE/DELETE), ActionTime, ActionBy (FK)
[Maintains an audit trail of all book-related changes using a trigger stored procedure]

CHAPTER 4

IMPLEMENTATION

The implementation phase of the **Bookstore Management System** involved transforming the design into a fully functional application. This chapter details the technical execution, including installation steps, front-end development using Tkinter, and back-end database operations with MySQL. The system was built using Python for its versatility and ease of integration with databases. Below, we break down each component with relevant code snippets and explanations.

4.1 Installation Steps for Execution of the Project

To run the Bookstore Management System, follow these steps:

Prerequisites:

- 1) Python 3.8+: Install from [python.org](https://www.python.org/).
- 2) MySQL Server: Download from MySQL Official Site.
- 3) Required Libraries: Install using pip:
`pip install mysql-connector-python Tkinter`
- 4) Database Setup: Create a MySQL database named ElDorado:
`[CREATE DATABASE ElDorado;]`
- 5) Launching the Application:
 - a. Clone the repository or download the source file.
 - b. Run the main script:
`python bookstore.py`

4.2 Front-End Code Snippets

The front-end was developed using Tkinter, providing an intuitive GUI for users. Key modules include:

Login Screen:

```
def show_login_screen(self):  
    self.login_window = tk.Toplevel(self.root)  
    self.login_window.title("Staff Login")
```

```
ttk.Label(self.login_window, text="Email:").pack()
self.login_email = ttk.Entry(self.login_window)
self.login_email.pack()
ttk.Button(self.login_window, text="Login", command=self.authenticate_staff).pack()
```

Features: Role-based authentication (Manager/Clerk/Librarian), Secure password handling (SHA-256 hashing).

Main Dashboard:

```
def setup_main_ui(self):
    buttons = [
        ("📚 Book Management", self.book_management),
        ("👤 Customer Management", self.customer_management)
    ]
    for text, command in buttons:
        ttk.Button(button_frame, text=text, command=command).pack()
```

Features: Dynamic button grid based on user role, disabled options for non-Manager staff.

Book Management Module:

```
def book_management(self):
    notebook = ttk.Notebook(window)
    add_frame = ttk.Frame(notebook)
    notebook.add(add_frame, text="➕ Add Book")
    ttk.Label(add_frame, text="Book Name:").grid(row=0, column=0)
    self.add_entries["Book Name"] = ttk.Entry(add_frame)
    self.add_entries["Book Name"].grid(row=0, column=1)
```

Features: CRUD operations via tabbed interface, real-time validation (e.g., Quantity ≥ 0).

4.3 Back-End Code Snippets

The back-end uses MySQL for data persistence, with a dedicated DatabaseManager class.

Database Connection:

```
class DatabaseManager:
    def __init__(self):
        self.connection = mysqlcon.connect(
            host="localhost",
            user="root",
            password="Password@123",
            database="ElDorado"
        )
        self.initialize_database()
```

Order Processing:

```
def place_order(self, status):
    try:
        cursor = self.db.connection.cursor()
        cursor.execute(
            "INSERT INTO Orders (CustomerID, StaffID, TotalAmount, Status) "
            "VALUES (%s, %s, %s, %s)",
            (customer_id, self.current_staff['id'], total_amount, status)
        )
        self.db.connection.commit()
    except Exception as e:
        self.db.connection.rollback()
```

Key Operations: Transaction management (commit/rollback), inventory updates on order completion.

Audit Logging:

```
def execute_query(self, query, params=None):
    cursor = self.connection.cursor()
    cursor.execute(query, params or ())
    if "INSERT" in query:
        self.log_action(cursor.lastrowid, "INSERT")
```

Features: Automatic logging of all book modifications, trigger-based updates (e.g., AfterBookInsert).

4.4 Integration Highlights

Data Validation:

```
@staticmethod
def validate_quantity(quantity: str) -> bool:
    return quantity.isdigit() and int(quantity) > 0
```

Security: (Password hashing via SHA-256)

```
def hash_password(password: str) -> str:
    return hashlib.sha256(password.encode()).hexdigest()
```

CHAPTER 5

INTERPRETATION OF RESULT

The Bookstore Management System was successfully implemented and tested, demonstrating efficient handling of bookstore operations. This chapter presents key result snapshots of the system in action, along with interpretations of its functionality and performance. These screenshots validate the system's ability to manage books, customers, orders, and staff while maintaining data integrity and security.

5.1 Result Snapshots

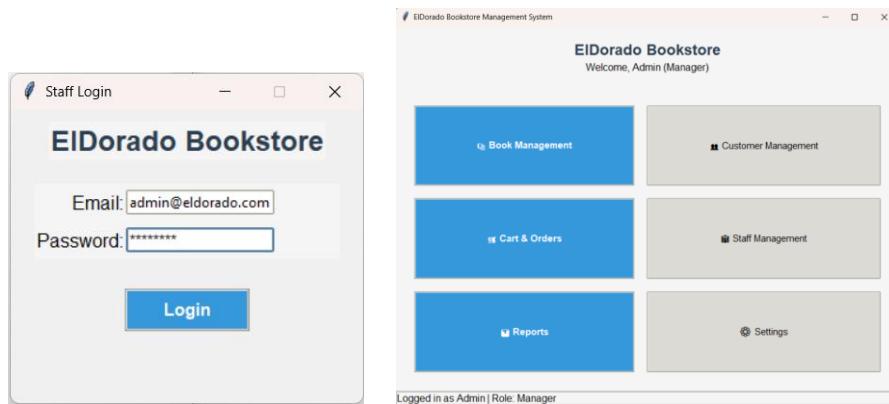


Fig 5.1(a) Login Page for Admin

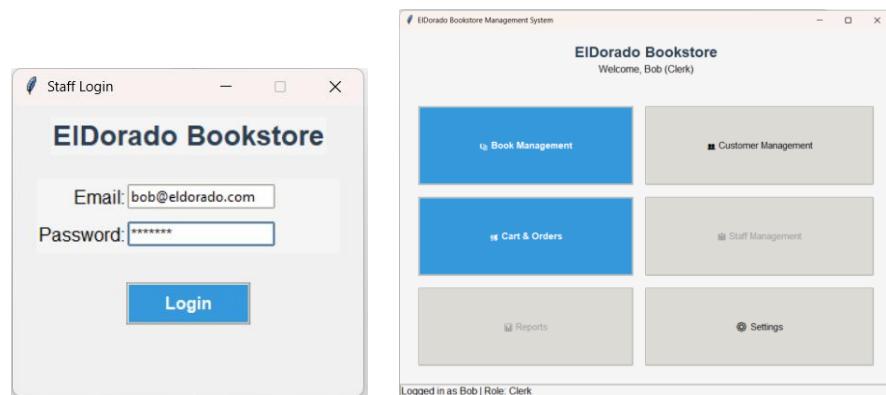
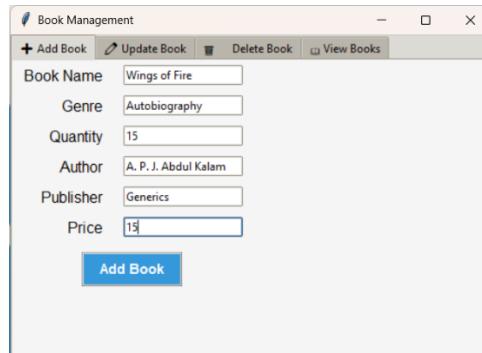


Fig 5.1(b) Login Page for Other Staff

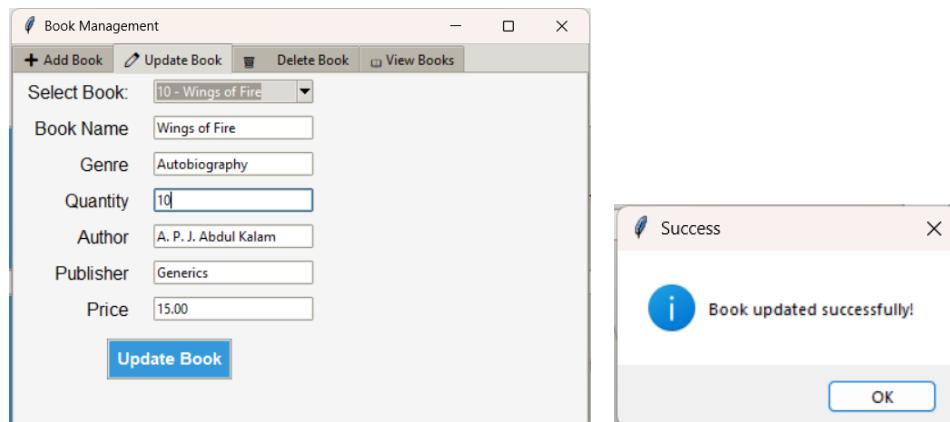
Interpretation: The login screen (Fig 5.1) showcases role-based authentication, where staff members must enter their email and password to access the system. Successful login redirects users to a dashboard tailored to their role (Manager, Clerk, or Librarian). The SHA-256 password hashing ensures security, preventing unauthorized access. This

snapshot confirms the system's adherence to authentication protocols and role-based access control (RBAC), a critical feature for multi-user environments.



ID	Book Name	Genre	Quantity	Author	Publisher	Price	Last Updated
1	The Alchemist	Fiction	15	Paulo Coelho	HarperOne	15.99	2025-05-20 11:47:35
2	1984	Dystopian	6	George Orwell	Secker & Warburg	13.50	2025-05-20 14:39:15
3	Sapiens	History	5	Yuval Noah Harari	Harper	18.75	2025-05-20 10:46:38
4	Clean Code	Programming	6	Robert C. Martin	Prentice Hall	25.00	2025-05-20 18:31:33
5	The Pragmatic Programmer	Programming	4	Andrew Hunt	Addison-Wesley	22.99	2025-05-20 10:46:38
6	IMH	Fantasy	0	Amish	Penguin Bookhouse	40.00	2025-05-20 18:36:07
10	Wings of Fire	Autobiography	15	A. P. J. Abdul Kalam	Generics	15.00	2025-05-27 09:30:07

Fig 5.2(a) Adding books



ID	Book Name	Genre	Quantity	Author	Publisher	Price	Last Updated
1	The Alchemist	Fiction	15	Paulo Coelho	HarperOne	15.99	2025-05-20 11:47:35
2	1984	Dystopian	6	George Orwell	Secker & Warburg	13.50	2025-05-20 14:39:15
3	Sapiens	History	5	Yuval Noah Harari	Harper	18.75	2025-05-20 10:46:38
4	Clean Code	Programming	6	Robert C. Martin	Prentice Hall	25.00	2025-05-20 18:31:33
5	The Pragmatic Programmer	Programming	4	Andrew Hunt	Addison-Wesley	22.99	2025-05-20 10:46:38
6	IMH	Fantasy	0	Amish	Penguin Bookhouse	40.00	2025-05-20 18:36:07
10	Wings of Fire	Autobiography	10	A. P. J. Abdul Kalam	Generics	15.00	2025-05-27 09:35:02

Fig 5.2(b) Updating a record

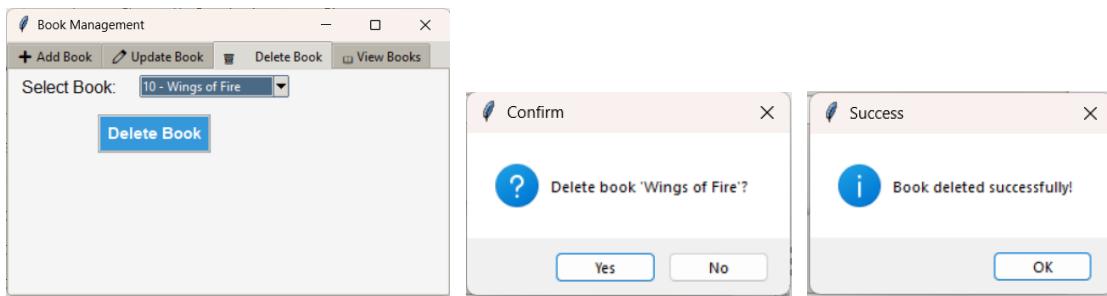


Fig 5.2(c) Deletion

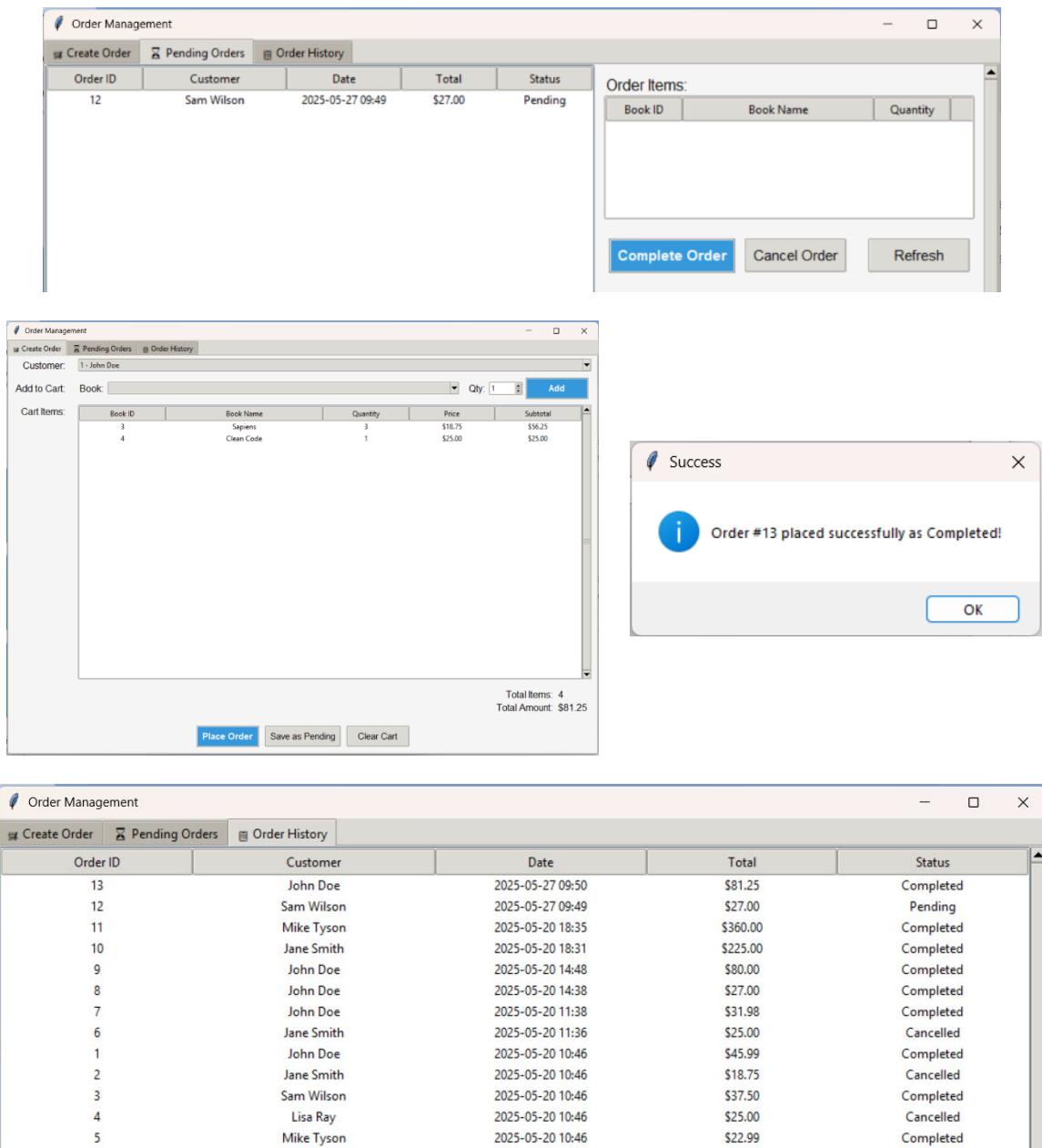
ID	Book Name	Genre	Quantity	Author	Publisher	Price	Last Updated
1	The Alchemist	Fiction	15	Paulo Coelho	HarperOne	15.99	2025-05-20 11:47:35
2	1984	Dystopian	6	George Orwell	Secker & Warburg	13.50	2025-05-20 14:39:15
3	Sapiens	History	5	Yuval Noah Harari	Harper	18.75	2025-05-20 10:46:38
4	Clean Code	Programming	6	Robert C. Martin	Prentice Hall	25.00	2025-05-20 18:31:33
5	The Pragmatic Programmer	Programming	4	Andrew Hunt	Addison-Wesley	22.99	2025-05-20 10:46:38
6	IMH	Fantasy	0	Amish	Penguin Bookhouse	40.00	2025-05-20 18:36:07

Fig 5.2(d) Viewing

Fig 5.2 CRUD operations on Books Table

Interpretation: The book management interface (Fig 5.2) displays a tabbed layout for adding, updating, and deleting books. The "View Books" tab lists all inventory items with details like genre, quantity, and price. Real-time validation ensures quantities and prices are non-negative, while the "Last Updated" timestamp reflects audit compliance. This result validates the CRUD (Create, Read, Update, Delete) functionality and the system's ability to maintain accurate inventory records.

Order Management					
Customer:		Cart Items:			
Add to Cart:		Book:	Qty:	Subtotal	
Book ID	Book Name	Quantity	Price	Total Items: 2	Total Amount: \$27.00
2	1984	2	\$13.50		
Place Order Save as Pending Clear Cart					



The figure displays three windows of the Order Management System:

- Pending Orders Window:** Shows a table with columns Order ID, Customer, Date, Total, and Status. One row is selected for Sam Wilson.
- Create Order Window:** Shows a table of Order Items with columns Book ID, Book Name, and Quantity. Buttons for Complete Order, Cancel Order, and Refresh are at the bottom.
- Order History Window:** Shows a table of completed orders with columns Order ID, Customer, Date, Total, and Status. The table lists 13 completed orders from various customers.

Fig 5.3 Order Processing Dashboard

Interpretation: The order management snapshot (Fig 5.3) illustrates a completed sales transaction, including customer details, purchased books, and total amount. The "Pending Orders" tab allows staff to update order statuses (Completed/Cancelled), with automatic inventory deduction upon completion. This demonstrates the system's transactional integrity—orders are either fully processed or rolled back in case of errors, ensuring data consistency.

CHAPTER 6

CONCLUSION AND FUTURE SCOPE

The Bookstore Management System achieved its objectives by providing a robust, user-friendly platform for managing bookstore operations. Key outcomes include:

- Automation: Streamlined inventory, sales, and customer management.
- Security: Role-based access and password hashing protected sensitive data.
- Reliability: Transactional database operations ensured data integrity.
- Auditability: The BookLog table tracked all changes for compliance.

The project also reinforced practical skills in Python (Tkinter), MySQL, and software design principles like MVC (Model-View-Controller).

6.1 Future Scope

- Advanced Analytics: Integrate tools like Tableau for sales trend analysis.
- Mobile Integration: Develop a companion app for customers to browse inventory and place orders.
- Cloud Migration: Host the database on AWS/Azure for scalability.
- Barcode Scanning: Add support for ISBN-based book lookup.
- AI Recommendations: Suggest books based on customer purchase history.
- This system lays a foundation for future enhancements to meet evolving business needs.

REFERENCES

- [1] E. Matthes, Python Crash Course, 2nd ed. San Francisco: No Starch Press, 2019.
(Core reference for Tkinter GUI development and Python best practices.)
- [2] P. DuBois, MySQL Cookbook, 3rd ed. Sebastopol: O'Reilly Media, 2014.
(Essential guide for database schema design and query optimization in MySQL.)
- [3] Python Software Foundation, Tkinter Documentation, 2023. [Online]. Available: <https://docs.python.org/3/library/tkinter.html>
(Official documentation for all Tkinter widgets and event handling.)