

BOOKSTORE MANAGEMENT SYSTEM



A PROJECT REPORT

Submitted by

NETHAJI J(2303811724321077)

in partial fulfillment of requirements for the award of the course
CGB1221-DATABASE MANAGEMENT SYSTEMS

in

ARTIFICIAL INTELLIGENCE AND DATA SCIENCE

K. RAMAKRISHNAN COLLEGE OF TECHNOLOGY

(An Autonomous Institution, affiliated to Anna University Chennai and Approved by AICTE, New Delhi)

SAMAYAPURAM – 621 112

JUNE- 2025

**K. RAMAKRISHNAN COLLEGE OF TECHNOLOGY
(AUTONOMOUS)**

SAMAYAPURAM – 621 112

BONAFIDE CERTIFICATE

Certified that this project report on “**BOOKSTORE MANAGEMENT SYSTEM**” is the bonafide work of **NETHAJI J(2303811724321077)** who carried out the project work during the academic year 2024 - 2025 under my supervision.

SIGNATURE

Dr.T. AVUDAIAPPAN, M.E.,Ph.D.,

HEAD OF THE DEPARTMENT

ASSOCIATE PROFESSOR

Department of Artificial Intelligence

K.Ramakrishnan College of Technology
(Autonomous)

Samayapuram–621112.

SIGNATURE

Mrs.S. GEETHA, M.E.,

SUPERVISOR

ASSISTANT PROFESSOR

Department of Artificial Intelligence

K.Ramakrishnan College of Technology
(Autonomous)

Samayapuram–621112.

Submitted for the viva-voce examination held on ...04.06.2025.....

INTERNAL EXAMINER

EXTERNAL EXAMINER

DECLARATION

I declare that the project report on “**BOOKSTORE MANAGEMENT SYSTEM**” is the result of original work done by me and best of my knowledge, similar work has not been submitted to “**ANNA UNIVERSITY CHENNAI**” for the requirement of Degree of **BACHELOR OF TECHNOLOGY**. This project report is submitted on the partial fulfilment of the requirement of the completion of the course **CGB1221 – DATABASE MANAGEMENT SYSTEMS.**

Signature

NETHAJI J

Place: Samayapuram

Date:04.06.2025

ACKNOWLEDGEMENT

It is with great pride that I express my gratitude and in-debt to my institution “**K.Ramakrishnan College of Technology (Autonomous)**”, for providing me with the opportunity to do this project.

I glad to credit honourable chairman **Dr. K. RAMAKRISHNAN, B.E.**, for having provided for the facilities during the course of my study in college.

I would like to express my sincere thanks to my beloved Executive Director **Dr. S. KUPPUSAMY, MBA, Ph.D.**, for forwarding to my project and offering adequate duration in completing my project.

I would like to thank **Dr. N. VASUDEVAN, M.Tech., Ph.D.**, Principal, who gave opportunity to frame the project the full satisfaction.

I whole heartily thanks to **Dr. T. AVUDAIAPPAN, M.E.,Ph.D.**, Head of the department, **ARTIFICIAL INTELLIGENCE** for providing his encourage pursuing this project.

I express my deep expression and sincere gratitude to my project supervisor **Mrs.S.GEETHA, M.E.**, Department of **ARTIFICIAL INTELLIGENCE**, for her incalculable suggestions, creativity, assistance and patience which motivated me to carry out this project.

I render my sincere thanks to Course Coordinator and other staff members for providing valuable information during the course.

I wish to express my special thanks to the officials and Lab Technicians of my departments who rendered their help during the period of the work progress.

INSTITUTE

Vision:

- To serve the society by offering top-notch technical education on par with global standards.

Mission:

- Be a center of excellence for technical education in emerging technologies by exceeding the needs of industry and society.
- Be an institute with world class research facilities.
- Be an institute nurturing talent and enhancing competency of students to transform them as all – round personalities respecting moral and ethical values.

DEPARTMENT

Vision:

- To excel in education, innovation, and research in Artificial Intelligence and Data Science to fulfil industrial demands and societal expectations.

Mission

- To educate future engineers with solid fundamentals, continually improving teaching methods using modern tools.
- To collaborate with industry and offer top-notch facilities in a conducive learning environment.
- To foster skilled engineers and ethical innovation in AI and Data Science for global recognition and impactful research.
- To tackle the societal challenge of producing capable professionals by instilling employability skills and human values.

PROGRAM EDUCATIONAL OBJECTIVES (PEO)

- **PEO1:** Compete on a global scale for a professional career in Artificial Intelligence and Data Science.
- **PEO2:** Provide industry-specific solutions for the society with effective communication and ethics.
- **PEO3** Enhance their professional skills through research and lifelong learning initiatives.

PROGRAM SPECIFIC OUTCOMES (PSOs)

- **PSO1:** Capable of finding the important factors in large datasets, simplify the data, and improve predictive model accuracy.
- **PSO2:** Capable of analyzing and providing a solution to a given real-world problem by designing an effective program.

PROGRAM OUTCOMES (POs)

Engineering students will be able to:

1. **Engineering knowledge:** Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.
2. **Problem analysis:** Identify, formulate, review research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences
3. **Design/development of solutions:** Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations
4. **Conduct investigations of complex problems:** Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions
5. **Modern tool usage:** Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modeling to complex engineering activities with an understanding of the limitations
6. **The engineer and society:** Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice
7. **Environment and sustainability:** Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development

- 8. Ethics:** Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.
- 9. Individual and team work:** Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings.
- 10. Communication:** Communicate effectively on complex engineering activities with the engineering community and with society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions.
- 11. Project management and finance:** Demonstrate knowledge and understanding of the engineering and management principles and apply these to one's own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.
- 12. Life-long learning:** Recognize the need for, and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change.

ABSTRACT

The Book Store Management System is a comprehensive software application designed to automate and streamline the daily operations of a bookstore. This system addresses the challenges of manually managing inventory, sales, customer records, and billing processes, offering an efficient digital solution to traditional store management. The main objective of the system is to simplify and optimize the management of book-related information, improve operational efficiency, and enhance customer satisfaction. The system allows administrators and staff to perform essential functions such as adding new books, updating book details, deleting old entries, tracking stock levels, and processing sales transactions. Additionally, it features customer management capabilities, enabling the store to maintain a database of customer interactions and purchase history. By automating routine tasks and providing a centralized platform for all bookstore activities, the Book Store Management System reduces manual errors, increases productivity, and supports the growth and modernization of bookstore businesses. It serves as a vital tool for ensuring smooth operations and improved customer engagement in the increasingly digital retail environment.

ABSTRACT WITH POs AND PSOs MAPPING

CO 5 : BUILD DATABASES FOR SOLVING REAL-TIME PROBLEMS.

ABSTRACT	POs MAPPED	PSOs MAPPED
<p>The Book Store Management System is a comprehensive software application designed to automate and streamline the daily operations of a bookstore. This system addresses the challenges of manually managing inventory, sales, customer records, and billing processes, offering an efficient digital solution to traditional store management. The main objective of the system is to simplify and optimize the management of book-related information, improve operational efficiency, and enhance customer satisfaction.</p>	PO1 -3 PO2 -3 PO3 -3 PO5 -3 PO6 -2 PO8 - 2 PO10 -2 PO11-2 PO12 -3	PSO1 -3 PSO2 -3

Note: 1- Low, 2-Medium, 3- High

TABLE OF CONTENTS

CHAPTER NO.	TITLE	PAGE NO.
	ABSTRACT INTRODUCTION	viii
1		1
	1.1 OBJECTIVE	1
	1.2 OVERVIEW	1
	1.3 SQL AND DATABASE CONCEPTS	2
2	PROJECT METHODOLOGY	3
	2.1 PROPOSED WORK	3
	2.2 BLOCK DIAGRAM	4
3	MODULE DESCRIPTION	5
	3.1 BOOK MANAGEMENT MODULE	5
	3.2 CUSTOMER MANAGEMENT MODULE	5
	3.3 AUTHENTICATION MODULE	6
	3.4 SALES BILLING MODULE	6
	3.5 INVENTORY MONITORING MODULE	7
	3.6 REPORT AND ANALYTICS MODULE	7
	3.7 SEARCH AND FILTER MODULE	7
4	CONCLUSION & FUTURE SCOPE	8
	APPENDIX A SOURCE CODE	10
	APPENDIX B SCREENSHOTS	16
	REFERENCES	19

CHAPTER 1

INTRODUCTION

1.1 OBJECTIVE

The objective of the **Book Store Management System** is to develop an efficient, reliable, and user-friendly software solution that automates the core functions of a bookstore, including inventory management, sales processing, and customer data handling. This system aims to eliminate the need for manual record-keeping by providing a centralized platform where administrators can easily add, update, delete, and search for books, generate invoices, and track transactions. It also facilitates real-time stock monitoring, secure user access, and the generation of analytical reports to support decision-making. Overall, the system is designed to enhance operational efficiency, reduce errors, and improve customer service in a modern bookstore environment.

1.2 OVERVIEW

The **Book Store Management System** is a comprehensive software application designed to streamline and digitize the daily operations of a bookstore. It provides an integrated platform to manage book inventory, handle sales transactions, maintain customer records, and generate reports, all within a secure and user-friendly interface. The system supports functionalities such as adding, updating, and deleting book entries, tracking stock levels in real time, processing invoices, and managing user access based on roles. By automating routine tasks and centralizing data management, the system enhances accuracy, improves efficiency, and supports better decision-making, making it an essential tool for modern bookstore operations.

1.3 SQL AND DATABASE CONCEPTS

Structured Query Language (SQL) is the standard language used for managing and manipulating relational databases. In a Book Store Management System, SQL plays a critical role in storing, retrieving, updating, and deleting data related to books, customers, and transactions. SQL provides the ability to create and manage tables, define relationships between data entities, and ensure data integrity using constraints such as primary keys and foreign keys. Through SQL queries, the system can quickly access and manipulate large volumes of data efficiently and reliably.

A well-structured database design is essential for any management system. In this project, normalization is applied to eliminate redundancy and ensure data consistency across related tables. For instance, separate tables are created for books, customers, sales, and staff, each linked through unique identifiers. This modular structure simplifies updates and prevents anomalies during data operations.

The core operations in the Book Store Management System are based on CRUD — Create, Read, Update, and Delete. These operations are implemented using SQL queries to manage book entries, customer details, and sales records. Additionally, SQL supports transactions to ensure that multiple related operations execute successfully as a unit. For example, when a book is sold, the system updates the inventory, records the sale, and generates an invoice—all within a single transaction to prevent inconsistencies if an error occurs during the process.

Data integrity and security are enforced through SQL constraints such as NOT NULL, UNIQUE, CHECK, and DEFAULT, which validate inputs and ensure the reliability of stored data. Indexes are used to speed up search operations, especially when querying large datasets such as book catalogs or sales histories.

CHAPTER 2

PROJECT METHODOLOGY

2.1 PROPOSED WORK

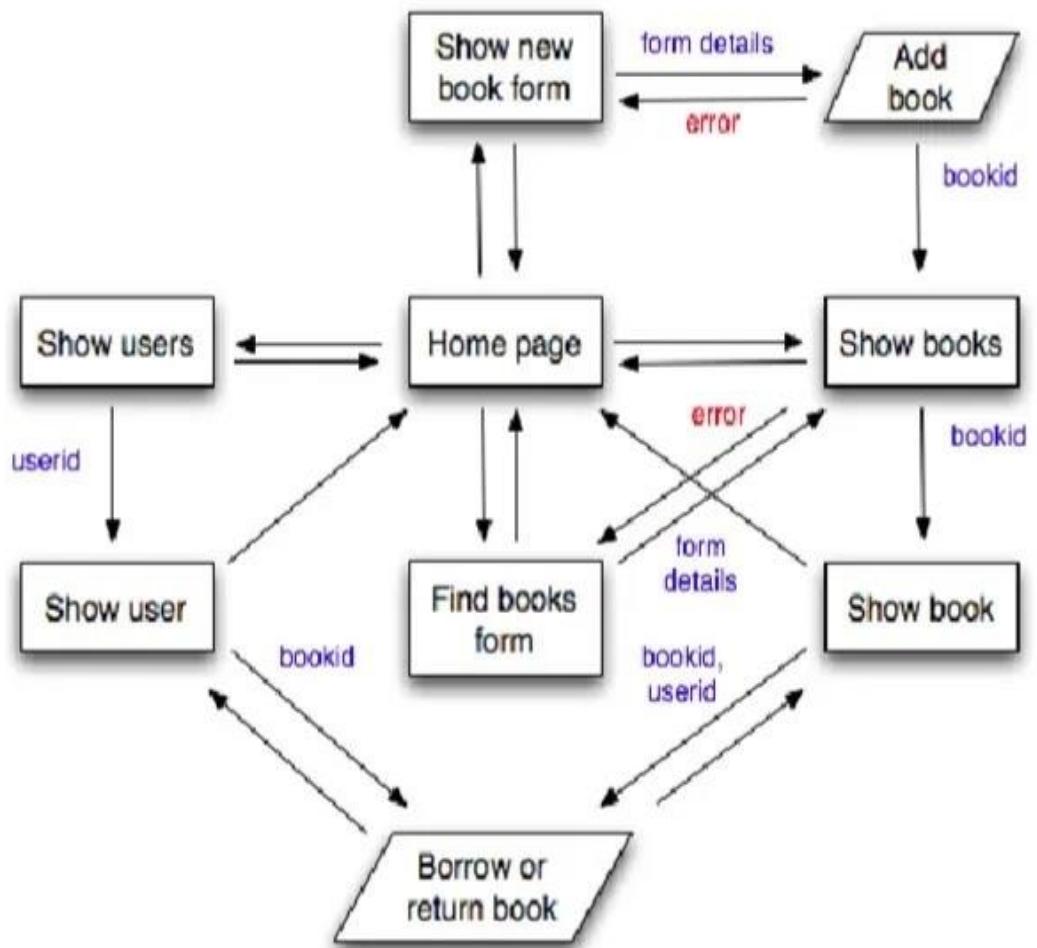
The proposed Book Store Management System is designed as a modular, interactive platform with separate components for inventory management, sales processing, customer handling, and reporting. Each module will function independently but remain interconnected through a centralized relational database. The system architecture follows a client-server model, where the front end provides a user-friendly interface for store staff, and the backend handles all logic and database interactions using SQL. This separation ensures scalability, maintainability, and smooth user experiences.

A key feature of the proposed system is the real-time management of book inventory. Staff will be able to add new book records, edit existing ones, and delete outdated entries. The system will track the number of copies available and automatically flag low stock levels to avoid shortages. Additional book information such as title, author, genre, price, and ISBN will be stored in the database, allowing easy filtering and searching based on various criteria.

Next, The proposed system will include a billing module that generates invoices upon the completion of each sale. Sales data will be stored along with timestamps, item details, and customer information (if available). A built-in reporting module will allow administrators to generate daily, weekly, or monthly summaries of sales performance, most sold books, and revenue trends.

The proposed design also includes the flexibility to integrate additional features in the future, such as online order handling. This forward-compatible structure ensures the system remains relevant as the bookstores operations grow.

2.2 BLOCK DIAGRAM



CHAPTER 3

MODULE DESCRIPTION

3.1 BOOK MANAGEMENT MODULE

The Book Management Module is a core component of the Book Store Management System that enables efficient handling of the entire book inventory. This module allows authorized users to add new books to the system by entering comprehensive details such as the book title, author name, publisher, genre, ISBN, price, and the number of copies available. It provides functionality to update existing book records, including modifying prices, correcting details, or adjusting stock quantities as new shipments arrive or sales are made.

The Book Management Module reduces errors associated with manual record-keeping and ensures that the bookstore always has accurate, up-to-date information on its inventory. This facilitates better stock control, timely restocking, and improved overall management of the bookstore's offerings.

3.2 CUSTOMER MANAGEMENT MODULE

The **Customer Management Module** is designed to efficiently handle all customer-related data within the bookstore system. It stores essential details such as customer names, contact information (phone number, email), and purchase history. This information helps the bookstore track customer preferences, identify frequent buyers, and offer personalized promotions or loyalty programs. The module allows authorized users to add new customers, update existing profiles, and remove outdated records, ensuring the database remains current and accurate. By maintaining comprehensive customer records, the system supports improved communication, better customer service, and targeted marketing efforts, ultimately helping to build stronger relationships between the bookstore and its customers.

3.3 AUTHENTHENTICCATION MODULE

They ensures secure access control by verifying user credentials before allowing entry into the system. It requires each user to log in with a unique username and password, protecting the system from unauthorized access. This module supports multiple user roles such as admin, cashier, and inventory manager, each with specific permissions and access levels. For example, only admins may access sensitive data and generate reports, while cashiers may only process sales and view inventory.

The system validates login credentials against stored records in a secure database and provides appropriate access based on the user's role. In case of incorrect credentials, the module prevents login and can log failed attempts for auditing purposes. Passwords are typically stored in hashed format to enhance security, and additional measures such as session timeouts or CAPTCHA verification can be integrated to prevent misuse.

3.4 SALES BILLING MODULE

The **Sales and Billing Module** is designed to handle all purchase-related transactions within the bookstore. It allows users to efficiently record sales by selecting books from the inventory, calculating the total cost, applying discounts if applicable, and generating a final bill or invoice. This module ensures fast and accurate billing by automatically updating stock quantities after each sale and saving transaction details in the database for future reference.

It captures important data such as date, time, items sold, quantity, price, and customer information (if available). The system generates printable receipts and maintains a record of all transactions for reporting and auditing purposes. By automating the entire sales process, this module minimizes manual errors, speeds up customer service, and provides valuable insights into daily sales performance and revenue generation.

3.5 INVENTORY MONITORING MODULE

The **Inventory Monitoring Module** helps the bookstore keep track of the number of books available in stock. It automatically updates the quantity whenever a book is sold or new stock is added. This module also gives alerts when any book is running low, so the store can restock on time. It helps avoid out-of-stock problems and ensures that popular books are always available. This keeps the bookstore running smoothly and helps in managing inventory more effectively.

3.6 REPORT AND ANALYTICS MODULE

The **Report and Analytics Module** helps the bookstore see how well it is doing. It shows reports of sales, top-selling books, and how much money was earned in a day, week, or month. It also helps track which books are in demand and which are not. This makes it easier for the owner to make smart decisions and improve the business.

3.7 SEARCH AND FILTER MODULE

The **Search and Filter Module** is designed to help users quickly and easily find the books they are looking for in the bookstore system. Instead of browsing through the entire book list, users can simply type in keywords such as the book title, author name, or ISBN number to locate specific books. In addition to basic search, the module allows users to apply filters like genre, language, price range, availability (in stock or out of stock), and publication year.

This is especially helpful when customers have specific needs, like finding all fiction books under a certain price or locating books by a particular author. The search is fast, accurate, and saves time for both staff and customers. It improves the overall user experience and makes the system much easier to use, especially when the inventory grows larger.

CHAPTER 4

CONCLUSION & FUTURE SCOPE

CONCLUSION

The Book Store Management System provides an efficient and reliable solution for managing all key aspects of a bookstore's operations. By automating inventory control, sales processing, customer management, and reporting, the system reduces manual effort, minimizes errors, and improves overall productivity. Features like role-based authentication, real-time inventory monitoring, and powerful search and filter options ensure security, accuracy, and ease of use. The system supports better decision-making through detailed analytics and helps maintain strong customer relationships. Overall, this management system modernizes bookstore operations, making them faster, more organized, and scalable for future growth.

FUTURE SCOPE

- The Developing a fully functional online storefront would allow customers to browse books, place orders, and make payments online. This would significantly widen the customer base beyond the physical location, offering convenience and 24/7 availability.
- Beyond just catalog browsing and online ordering, future development can include features such as real-time inventory syncing, customer reviews and ratings, wish lists, personalized book recommendations, and secure payment gateways. Integration with global shipping services can enable the store to deliver books internationally, opening up new markets.
- Integrating AI-powered customer support into they System can greatly enhance user experience and operational efficiency. AI chatbots or virtual assistants can handle common customer queries such as checking book availability, tracking orders, providing store hours, and answering frequently asked questions instantly and around the clock.

- Adding community features and social engagement tools can transform the Book Store Management System into a vibrant platform that connects readers, authors, and staff. Features such as discussion forums, book review sections, and user-generated content allow customers to share opinions, recommend books, and engage in meaningful conversations.
- Adopting a mobile-first approach means designing the Book Store Management System primarily for mobile devices, ensuring it works smoothly on smartphones and tablets before scaling up to desktops. This approach is important because many customers and staff use mobile devices to access the system on the go. Progressive Web Apps (PWAs) take this further by combining the best of websites and mobile apps.
- The Book Store Management System uses a database to store and manage all important information such as book details, customer records, sales transactions, inventory data, and user authentication. The database helps organize this data efficiently so that the system can quickly retrieve, update, or delete information as needed.

APPENDIX A SOURCE CODE

```
conn = sqlite3.connect("books.db", check_same_thread=False)
cursor = conn.cursor()

cursor.execute("")
CREATE TABLE IF NOT EXISTS books (
    id INTEGER PRIMARY KEY AUTOINCREMENT,
    title TEXT NOT NULL,
    author TEXT NOT NULL,
    genre TEXT NOT NULL,
    price REAL NOT NULL,
    stock INTEGER NOT NULL
)
""")
conn.commit()
```

```
INSERT def delete_book(book_id):
```

```
    cursor.execute("SELECT * FROM books WHERE id=?", (book_id,))
```

```
    if not cursor.fetchone():
```

```
        return "Book ID not found."
```

```
    cursor.execute("DELETE FROM books WHERE id=?", (book_id,))
```

```
    conn.commit()
```

```
    return f"Book ID {book_id} deleted successfully."
```

```
with gr.Tab("➕ Add Book"):
```

```
    with gr.Row():
```

```
        title = gr.Text(label="Title")
```

```
        author = gr.Text(label="Author")
```

```
    with gr.Row():
```

```
genre = gr.Text(label="Genre")

price = gr.Number(label="Price (₹)", precision=2)

stock = gr.Number(label="Stock", precision=0)

add_btn = gr.Button("Add Book")

add_result = gr.Textbox(label="Status")
```

```
add_btn.click(add_book, inputs=[title, author, genre, price, stock],  
outputs=add_result)
```

with gr.Tab(" Update Book"):

with gr.Row():

```
book_id = gr.Number(label="Book ID", precision=0)
```

```
new_title = gr.Text(label="New Title")
```

with gr.Row():

```
new_author = gr.Text(label="New Author")
```

```
new_genre = gr.Text(label="New Genre")
```

with gr.Row():

```
new_price = gr.Number(label="New Price (₹)", precision=2)
```

```
new_stock = gr.Number(label="New Stock", precision=0)
```

```
update_btn = gr.Button("Update Book")
```

```
update_result = gr.Textbox(label="Update Status")
```

```
update_btn.click(update_book, inputs=[book_id, new_title, new_author, new_genre,  
new_price, new_stock], outputs=update_result)
```

```
with gr.Tab("Delete Book"):
```

```
    del_book_id = gr.Number(label="Book ID to Delete", precision=0)
    del_btn = gr.Button("Delete Book")
    del_result = gr.Textbox(label="Delete Status")
    del_btn.click(delete_book, inputs=[del_book_id], outputs=del_result)
```

```
INSERT INTO `deductions` (`id`, `deduction`, `description`) VALUES
```

```
(1, 'Cash Advance', 'Cash Advance'),
```

```
(3, 'Sample', 'Sample Deduction');
```

```
with gr.Tab(" View Books"):
```

```
    view_btn = gr.Button("Load Books")
    book_output = gr.Textbox(label="Book List", lines=10)
    view_btn.click(list_books, outputs=book_output)
```

```
with gr.Row():
```

```
    genre = gr.Text(label="Genre")
    price = gr.Number(label="Price (₹)", precision=2)
    stock = gr.Number(label="Stock", precision=0)
    add_btn = gr.Button("Add Book")
    add_result = gr.Textbox(label="Status")
    add_btn.click(add_book, inputs=[title, author, genre, price, stock],
outputs=add_result)
```

```
INSERT INTO `employee` (`id`, `employee_no`, `firstname`, `middlename`, `lastname`,
`department_id`, `position_id`, `salary`) VALUES
```

```
(9, '2020-9838', 'John', 'C', 'Smith', 1, 1, 30000);
```

```
CREATE TABLE `employee_allowances`
```

```

`type` tinyint(1) NOT NULL COMMENT '1 = Monthly, 2= Semi-Montly, 3 = once',
`amount` float NOT NULL,
`effective_date` date NOT NULL,
`date_created` datetime NOT NULL DEFAULT current_timestamp()
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;

```

```

INSERT INTO `employee_allowances` (`id`, `employee_id`, `allowance_id`, `type`,
`amount`, `effective_date`, `date_created`) VALUES
(1, 9, 4, 1, 1000, '0000-00-00', '2020-09-29 11:20:04'),
(3, 9, 3, 2, 300, '0000-00-00', '2020-09-29 11:37:31'),
(5, 9, 1, 3, 1000, '2020-09-16', '2020-09-29 11:38:31');

```

```

CREATE TABLE `employee_deductions` (
`id` int(30) NOT NULL,
`employee_id` int(30) NOT NULL,
`deduction_id` int(30) NOT NULL,
`type` tinyint(1) NOT NULL COMMENT '1 = Monthly, 2= Semi-Montly, 3 = once',
`amount` float NOT NULL,
`effective_date` date NOT NULL,
`date_created` datetime NOT NULL DEFAULT current_timestamp()
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;

```

```

INSERT INTO `employee_deductions` (`id`, `employee_id`, `deduction_id`, `type`,

```

```
`amount`, `effective_date`, `date_created`) VALUES  
(2, 9, 3, 2, 500, '0000-00-00', '2020-09-29 11:52:46'),  
(3, 9, 1, 3, 1500, '2020-09-16', '2020-09-29 11:53:27');
```

```
CREATE TABLE `payroll` (  
    `id` int(30) NOT NULL,  
    `ref_no` text NOT NULL,  
    `date_from` date NOT NULL,  
    `date_to` date NOT NULL,  
    `type` tinyint(1) NOT NULL COMMENT '1 = monthly ,2 semi-monthly',  
    `status` tinyint(1) NOT NULL DEFAULT 0 COMMENT '0 =New,1 = computed',  
    `date_created` datetime NOT NULL DEFAULT current_timestamp()  
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;
```

```
INSERT INTO `payroll` (`id`, `ref_no`, `date_from`, `date_to`, `type`, `status`,  
    `date_created`) VALUES  
(1, '2020-3543', '2020-09-16', '2020-09-30', 2, 1, '2020-09-29 15:04:13');
```

```
CREATE TABLE `payroll_items` (  
    `id` int(30) NOT NULL,  
    `payroll_id` int(30) NOT NULL,  
    `employee_id` int(30) NOT NULL,  
    `present` int(30) NOT NULL,
```

```
INSERT INTO `position` (`id`, `department_id`, `name`) VALUES (1, 1,
'Programmer'),
(2, 2, 'HR Supervisor'),
(4, 3, 'Accounting Clerk');
```

```
CREATE TABLE `users` (
`id` int(30) NOT NULL,
`doctor_id` int(30) NOT NULL,
`name` varchar(200) NOT NULL,
`address` text NOT NULL,
`contact` text NOT NULL,
`username` varchar(100) NOT NULL,
`password` varchar(200) NOT NULL,
`type` tinyint(1) NOT NULL DEFAULT 2 COMMENT '1=admin , 2 = staff
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;
```

```
del_btn.click(delete_book, inputs=[del_book_id], outputs=del_result)

app.launch(share=True)
```

APPENDIX B SCREENSHOTS

Bookstore Management System

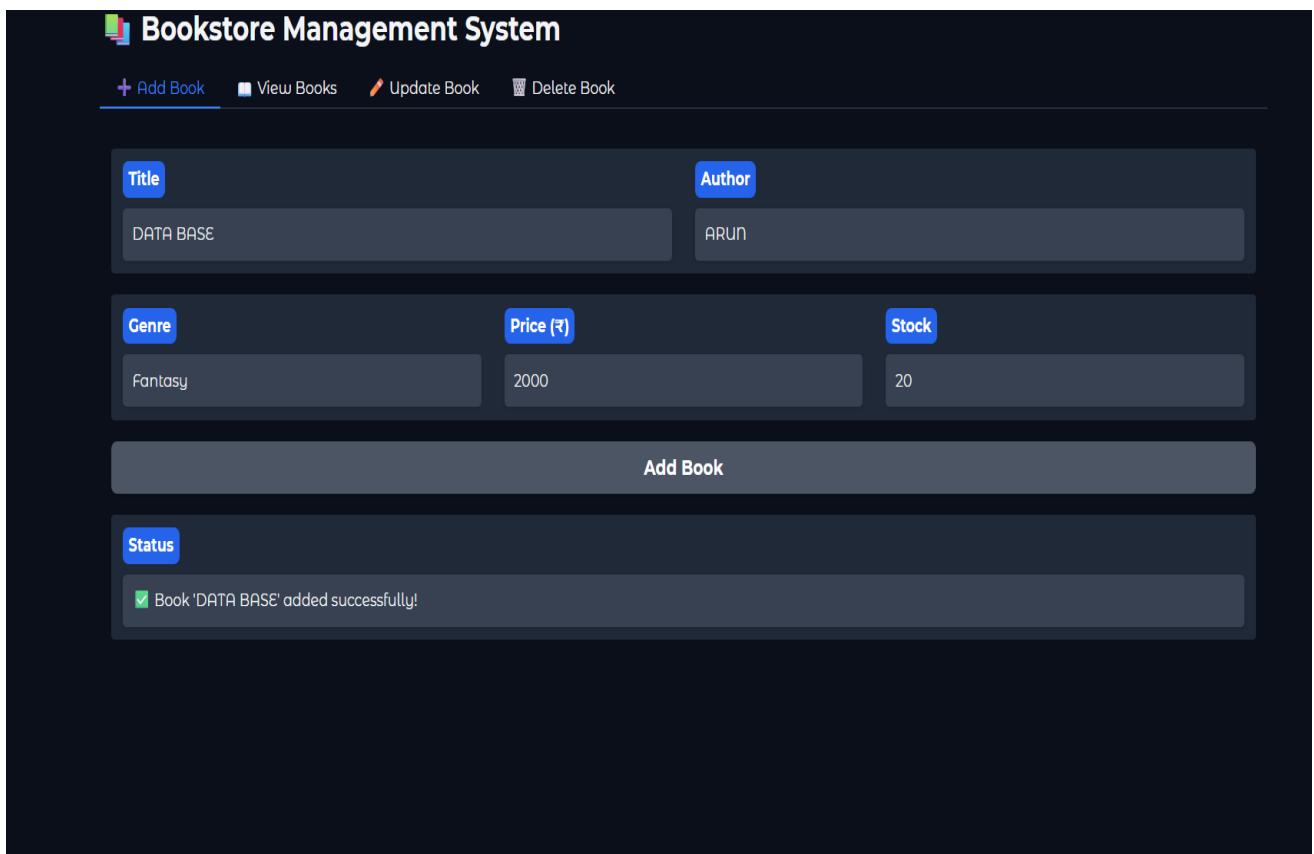
+ Add Book View Books Update Book Delete Book

Title	Author	
DATA BASE	ARUN	
Genre	Price (₹)	Stock
Fantasy	2000	20

Add Book

Status

Book 'DATA BASE' added successfully!



Bookstore Management System

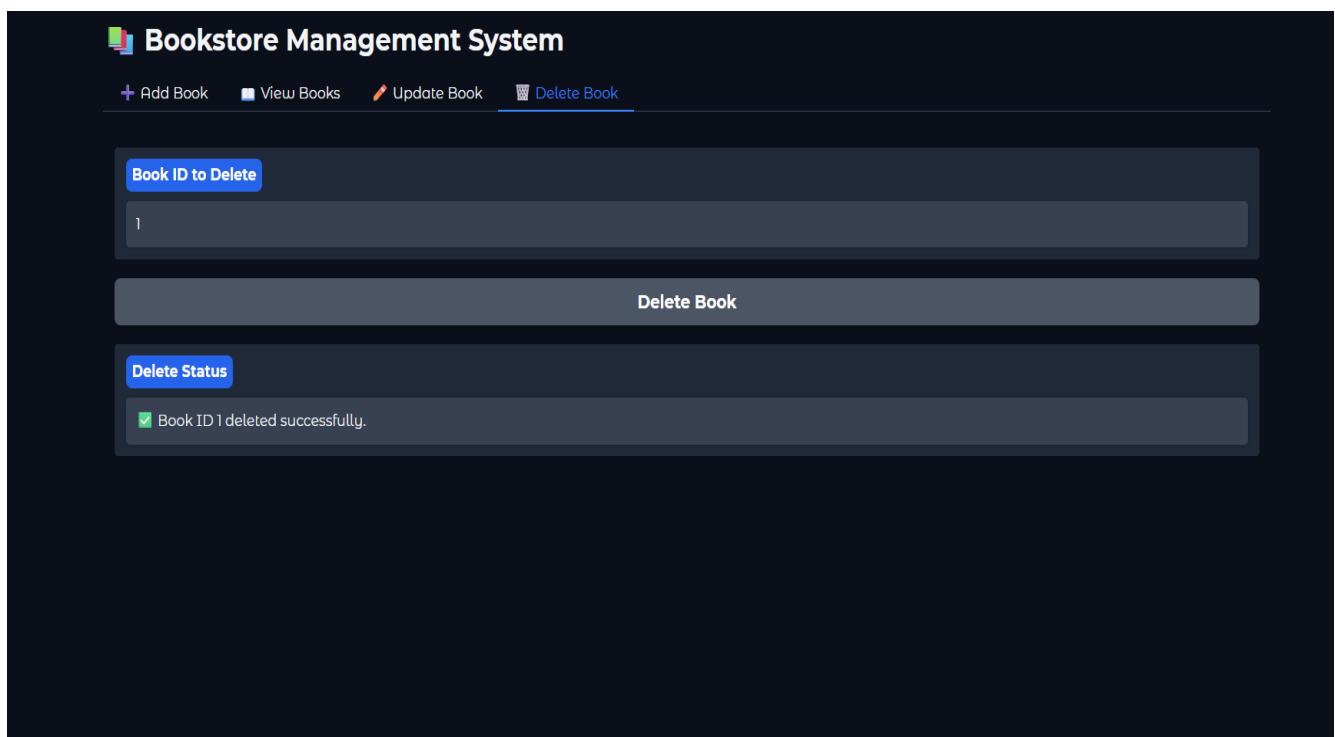
+ Add Book View Books Update Book Delete Book

Book ID to Delete
1

Delete Book

Delete Status

Book ID1 deleted successfully.



Book Store Management System

+ Add Book View Books Add Customer Purchase Book View Purchases Raw Database View

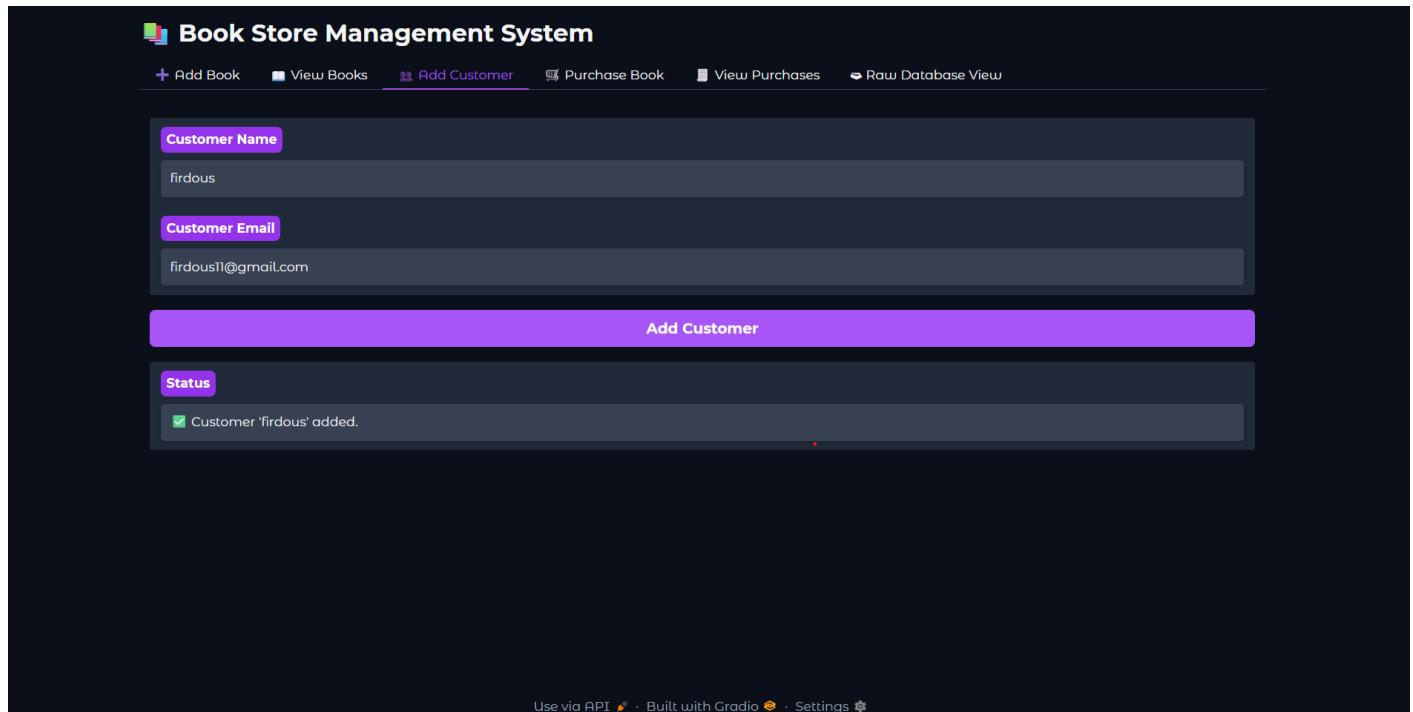
Customer Name
firdous

Customer Email
firdous1@gmail.com

Add Customer

Status
 Customer 'firdous' added.

Use via API 🔍 · Built with Gradio 🎨 · Settings⚙️



This screenshot shows the 'Add Customer' page of the Book Store Management System. It features a dark-themed interface with purple highlights. At the top, there are navigation links for adding books, viewing books, adding customers, purchasing books, viewing purchases, and raw database access. The main form has fields for 'Customer Name' (containing 'firdous') and 'Customer Email' (containing 'firdous1@gmail.com'). A central button labeled 'Add Customer' is highlighted in purple. Below the form is a status bar with the message 'Customer "firdous" added.' preceded by a green checkmark icon. At the bottom of the page, there are links for using the API, built-in Gradio, and settings.

Book Store Management System

+ Add Book View Books Add Customer Purchase Book View Purchases Raw Database View

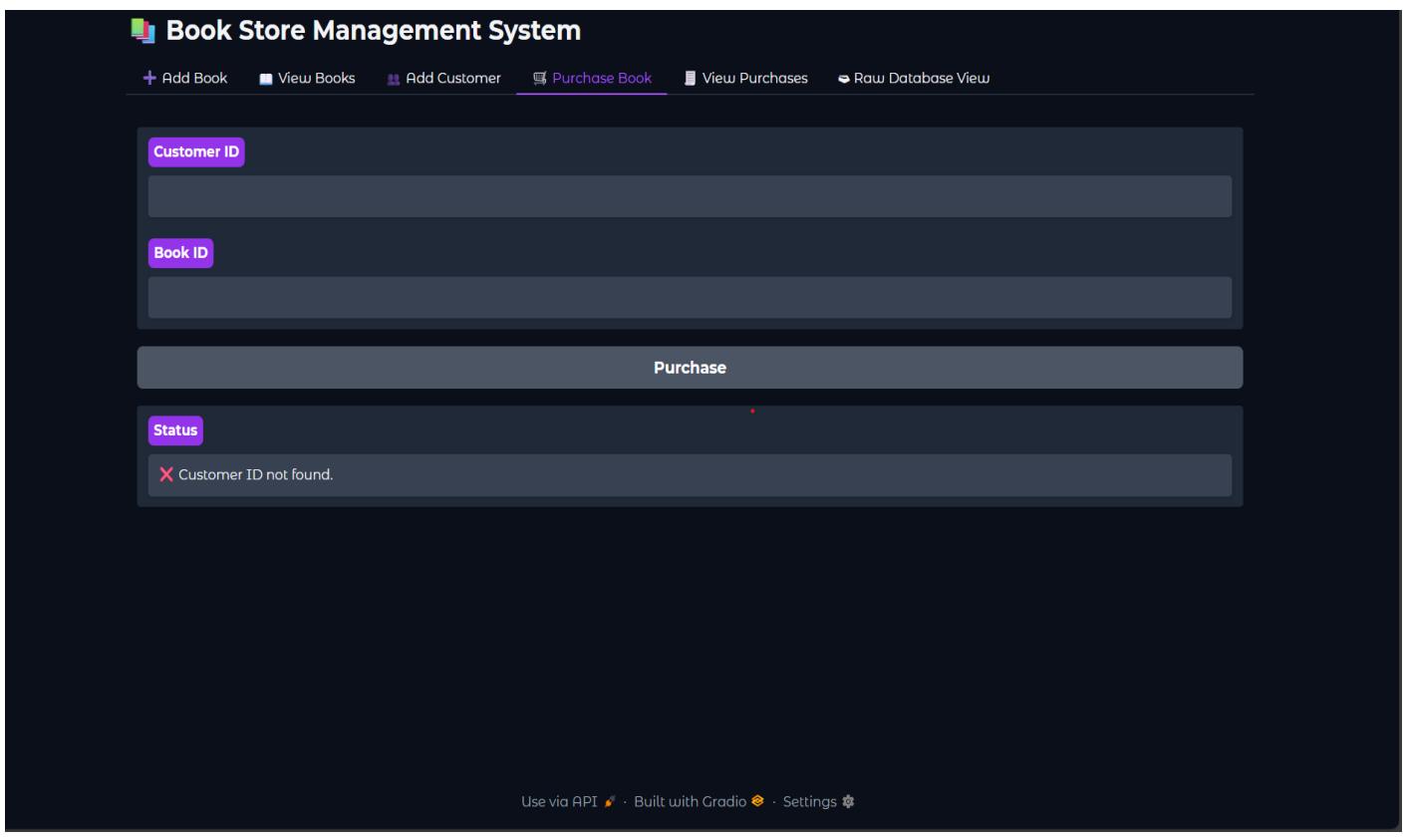
Customer ID

Book ID

Purchase

Status
✗ Customer ID not found.

Use via API 🔍 · Built with Gradio 🎨 · Settings⚙️



This screenshot shows the 'Purchase Book' page of the Book Store Management System. The interface is similar to the previous one, with a dark theme and purple highlights. The 'Purchase' section contains fields for 'Customer ID' and 'Book ID', both of which are currently empty. Below the fields is a large button labeled 'Purchase' centered under the heading. Underneath the button is a 'Status' section containing a red '✗' icon and the message 'Customer ID not found.'. At the bottom of the page, there are links for using the API, built-in Gradio, and settings.

Book Store Management System

+ Add Book View Books Add Customer Purchase Book View Purchases Raw Database View

Show Purchases

Purchases

ID: 1 | Customer: firdous | Book: java | Date: 2025-05-29 05:19

Bookstore Management System

+ Add Book View Books Update Book Delete Book

Book ID: 0 New Title:

New Author: New Genre:

New Price (₹): 0 New Stock: 0

Update Book

Update Status:

Use via API 🔍 · Built with Gradio 🎨 · Settings ⚙️

REFERENCES

1. Korth, H. F., Silberschatz, A., & Sudarshan, S. (2010). *Database System Concepts* (6th Edition). McGraw-Hill Education.
2. Connolly Helly, G. B., & Vermaat, M. E. (2012). *Discovering Computers: Fundamentals*.
3. Ullman, J. D. (2016). *A First Course in Database Systems* (3rd Edition). Pearson.
4. Pressman, R. S. (2014). *Software Engineering: A Practitioner's Approach* (8th Edition). McGraw-Hill Education.
5. OpenAI. (2023). *ChatGPT: Optimizing Language Models for Dialogue*. Retrieved from <https://openai.com/chatgpt>
6. XAMPP Control Panel. (2020). *XAMPP Official Documentation*. Available at: <https://www.apachefriends.org/index.html>
7. Mozilla Developer Network (MDN). (2024). *Progressive Web Apps (PWA)*. https://developer.mozilla.org/en-US/docs/Web/Progressive_web_apps.