



Data-Driven Online Book Store with ASP.NET Core 8: Sales Tracking and Best-Seller Analysis

Jaber Rafiyoddin Sayyed

Bachelor of Technology
(Computer Engineering)
Gramin Technical and
Management Campus
Nanded, India

Taha Ali Kaleem Ali

Bachelor of Technology
(Computer Engineering)
Gramin Technical and
Management Campus
Nanded, India

Mohammad Irshad

Bachelor of Technology
(Computer Engineering)
Gramin Technical and
Management Campus
Nanded, India

Abstract

It outlines how one can implement a secure, efficient online bookshop using ASP.NET Core 8, where security, usability, and data-driven decision-making come first, giving the best digital experience in buying. On the admin side, it provides an organized, alphanumeric sales tracking system that provides real-time reports on the inventory levels, most sold titles, and total sales. Otherwise, with text-based reports that are not graphic-intensive, the administrators would identify times when there is a need to stock or a marketing opportunity exists, gaining an instantaneous insight into what is selling. The payment gateway has adopted multiple safe payments and user data security methods, transaction integrity, such as cash on delivery and credit/debit cards, and UUPI all strictly made secure.

This is one of the characteristics of this application through which intelligent data processing evaluates sales data to show best-selling and trending books. This feature will display the in-demand items in real-time according to the client's interests, creating a compelling user experience. We detail the system's design, security protocols, data management techniques, and performance optimizations throughout the article, which shows how structured reporting and sophisticated data processing can offer improvements in operational efficiency and user happiness. This strategy provides a template for any future e-commerce platforms that want to balance security, usability, and user engagement well.

Keywords

ASP.NET Core 8, Online Book Store, Sales Tracking, Secure Transactions, Data Processing, E-commerce Security, Structured Reporting, User Engagement, Alphanumeric Reporting, Inventory Management, Role-Based Access Control, Payment Integration, UPI Transactions, Digital Bookstore, Customer Loyalty Programs, Data Encryption, Password Security

1. Introduction

Context and Relevance

With fast growth in e-commerce, the art of retailing has been redefined and provides customers with an infinitely practical alternative to traditional purchases. Online bookselling has proved to be an extremely popular tool in many industries by catering to the needs of its users from one of the largest selections available. This article speaks about the construction of a rich-featured, safe online bookshop using ASP.NET Core 8, which would provide an effective administrative panel to monitor and manage sales, inventory, and customer interaction besides the end-user-friendly interface.

Problem Statement

Some of the key problems that occur in the context of an e-bookshop include the security of data, dynamic control of levels of inventory, meaningful client preferences, and dependable transaction processing. It demands proper backend systems, safe processing of transactions, and an organized approach toward data presentation.

Research Objectives

The following are the main goals of this study:

1. Develop a secure, structured online bookstore with efficient data handling.
2. Implement text-based sales and inventory reports for streamlined decision-making.
3. Support multiple secure payment methods to enhance user experience.

2. System Architecture and Technology Stack

ASP.NET Core 8 Framework

This application is developed using ASP.NET Core 8, for which reasons are a great performance alongside cross-platform interoperability and modularity. Coupled with the architectural background laid by ASP.NET Core, built around MVC, separation of concerns is supported in order to allow truly scalable development and simple maintenance, two important attributes required for an e-commerce platform.

Database Design and Management

The system uses SQL Server for database management, structured to efficiently handle large volumes of transactions, user data, and inventory records. Key tables include:

- **Users:** Contains details for user authentication and role assignments, secured through ASP.NET Core Identity.
- **Books:** Stores book information, including title, author, category, price, and current stock levels.
- **Sales:** Logs each transaction, providing valuable data for sales reports and inventory monitoring.
- **Cart:** Manages items that users have added for checkout, facilitating seamless transitions to the purchase process.

Table	Key Fields	Purpose
User	UserID, Email, Role	Manages user credentials and roles for security.
Books	BookID, Title, Price	Stores book details for inventory tracking.
Sales	SaleID, UserID, Amount	Logs sales data for revenue tracking.
Cart	CartID, UserID, BookID	Tracks user-selected items for purchase.

Table 1: Database Schema Overview

Role-Based Access Control (RBAC)

By using ASP.NET Core Identity to enforce RBAC, the application makes sure that only authorized administrators can access sensitive features like viewing sales data, managing book inventory, and changing prices. By restricting access to essential functions according to user roles, RBAC improves security.

3. Sales and Revenue Tracking: Text-Based Reports for the Admin Panel

Another core module of this application is the Sales and Revenue Tracking tool, designed to give administrators organized, alphanumeric data reports showing the clarity of platform performance. The text-based report is delivered in real-time to provide platform administrators with the total revenue, sales volume by book category, and the status of the stock. Administrators can easily identify peak demand periods, evaluate recent promotions, and make analytics-driven inventory decisions without laborious visualizations. Apart from the ideal inventory levels maintained, real-time low-stock warnings also ensure that popular inventory is always available to customers and avoid probable sales loss due to a stockout.

3.1 Daily, Weekly, and Monthly Sales Data

The system generates sales reports based on specific timeframes, displaying key metrics without graphical visualizations. Reports include:

- **Total Revenue:** Shown as cumulative values for daily, weekly, and monthly periods, providing insight into revenue growth.
- **Sales Volume by Category:** Includes breakdowns by genre or category, allowing administrators to track demand patterns.
- **Inventory Alerts:** Displays real-time stock levels for each book, with alerts for low-stock items, helping admins avoid potential stockouts.

Example Report Output: A typical daily sales report might display:

- **Date:** 2024-10-01
- **Total Revenue:** \$2,500
- **Fiction Sales Volume:** 125 units
- **Non-Fiction Sales Volume:** 75 units
- **Top-Selling Book:** "The Art of Science" – 50 units sold
- **Low Stock Alert:** "Business Essentials" – 10 units left

3.2 Benefits of Structured Reporting

The alphanumeric report format offers several benefits:

- **Rapid Identification of Trends:** Administrators can quickly scan sales figures and stock alerts, identifying peak sales periods without the need for complex visuals.
- **Optimized Inventory Management:** Real-time stock levels and low-stock warnings enable proactive restocking, minimizing interruptions in availability.
- **Evaluation of Marketing Efforts:** Administrators can compare revenue data across different days or weeks to assess the impact of recent marketing campaigns or promotions.

This structured reporting system supports efficient data analysis and decision-making, providing administrators with immediate, actionable insights.

4. Data Processing for Recommendations and Customer Insights

This technology mines sales history based on complex data computation and can identify highly demanded books and increase client engagement through appropriate recommendations. The popular titles are determined by the software along with the dynamic change of suggested books through past and current sales data. This strategy encourages greater connection and customer satisfaction through pertinent recommendations based on the present demand and general trends. To guarantee that popular titles are always available, administrators can also use this data to modify stock levels. The platform's offerings are successfully matched with user interests through this methodical, data-driven approach, making for a more interesting and customized purchasing experience.

4.1 Identifying Top-Selling Books

Through transactional data analysis, the platform identifies books with high sales volumes, highlighting these as "top-sellers" on the user dashboard. This functionality does not require graphical visualization, as the best-seller status is updated directly based on accumulated sales figures.

Example of Data-Driven Recommendations

The system will automatically mark "mystery" and "self-help" books as best-sellers if transactional data shows that these genres are regularly popular. This will enable users to quickly view popular recommendations. Discoverability is improved by this feature, particularly for new users.

4.2 Customer Incentives and Loyalty Programs

The system includes logic-driven incentives to foster customer loyalty. Examples of these incentives include:

- **Discounts for High-Value Purchases:** Users who exceed a certain spending threshold receive additional discounts or vouchers, encouraging larger transactions.
- **Free Delivery for Bulk Orders:** Orders above a specified value qualify for free delivery, increasing the platform's value for frequent shoppers.
- **Loyalty Rewards:** Users who consistently purchase books receive exclusive vouchers, promoting long-term engagement.

These incentives are applied based on predefined rules, ensuring that loyal customers are rewarded and new users are encouraged to make repeat purchases.

5. Secure Payment and Transaction System

Complying with the strict security measures in place, the program enables safe transactions through various payment options that are cash on delivery, credit/debit cards, and UPI. While the UPI-based transactions are OTP-based for verifications by a particular user, the credit/debit card transactions are accepted by PCI-DSS-compliant gateways, whereby all sensitive data is encrypted and safe. All cash-on-delivery transactions are recorded to ascertain that traceability and offline payment security are in place. In addition to improving the convenience of customers, this multi-option strategy has further protected the anonymity and integrity of financial data, promoting dependability and trust in the operations of transactions in the platform.

5.1 Payment Options

1. **UPI Transactions:** Provides a fast, secure payment method, requiring OTP verification to confirm payments.
2. **Credit/Debit Card Transactions:** Processes payments through PCI-DSS-compliant gateways, ensuring that sensitive data is encrypted and protected.
3. **Cash on Delivery:** Offers flexibility for customers who prefer offline payments, enhancing accessibility for all user types.

5.2 Security Measures in Transactions

Sensitive data is protected by the system's strong security features, which include data encryption and tokenization. Users are shielded from potential data breaches by the secure storage and encryption of all financial data.

Payment Method	Security Protocol	Description
UPI	OTP Authentication	Mobile-based, secure authentication via OTP.
Credit/Debit	PCI-DSS Compliance	Securely processes card data, using encryption.
Cash on Delivery	Transaction Logging	Records each offline payment for traceability.

Table 2: Secure Payment Options and Security Protocols

5.3 Secure Storage of User Passwords

The application uses robust encryption when saving the passwords to the database, and it is very helpful in the protection of user credentials and the prevention of any undesirable access. Controls password security for ASP.NET Core Identity by automatically using hashing and salting techniques to safety-protect passwords. This approach ensures that a hacker cannot obtain unauthorized administrator privileges nor extract plaintext passwords even if they manage to access the database.

Encryption and Salting Process:

- **Hashing with Salt:** When users set or update their password, the application generates a unique salt and combines it with the plaintext password. This salted password is then hashed using a strong algorithm (e.g., SHA-256 or SHA-512).
- **Secure Hash Storage:** Only the hashed password (not the original) is stored in the database, along with the unique salt. This process makes it nearly impossible for attackers to reverse-engineer the original password.
- **Authentication Verification:** During login, the application hashes the user-entered password with the stored salt and compares it with the hashed password in the database. Only matching hashes allow access.

This method has much reduced the possibility of information theft using encryption and salted hashing. It is designed in a way that hackers may not get user passwords, nor can they be elevated to an administrator level even if data is breached. These mechanisms will maintain the integrity of this application and ensure user's information is not compromised in such possible risks.

6. Scalability and Performance Optimization

To ensure a seamless user experience as the platform scales, several optimization techniques are applied:

1. **Database Optimization:** Techniques like indexing and data partitioning support faster data retrieval, essential for real-time sales and inventory tracking.
2. **Caching:** Frequently accessed data, such as top-selling books, is stored in memory, reducing server load and improving response times.
3. **Load Balancing:** Distributes incoming requests across multiple servers during peak times, maintaining consistent performance.

These techniques allow the platform to efficiently handle increasing traffic and data volume without compromising speed or reliability.

7. Testing and Quality Assurance

The platform underwent rigorous testing to ensure functionality, security, and usability:

1. **Unit Testing:** Verified core functionalities, such as user authentication, payment processing, and report generation.
2. **Integration Testing:** Ensured that various components, like the admin panel and payment system, interact smoothly.
3. **User Acceptance Testing:** Collected feedback from end-users to refine the platform's usability, ensuring a smooth and intuitive shopping experience.

8. Results and Discussion

Results

Upon implementation, the system achieved several key objectives:

- **Effective Sales Tracking:** The alphanumeric reports provided administrators with accessible, real-time data on revenue and stock levels, supporting data-driven decisions.
- **Enhanced User Engagement:** The recommendation system and incentives encouraged users to explore and purchase more books, increasing customer satisfaction.
- **Secure Payment Processing:** The multi-option, secure payment structure offered users a convenient and safe transaction experience, bolstering trust in the platform.

Discussion

Incentive-driven user interaction together with structured, text-based reporting make this e-commerce platform incredibly more efficient. The system serves as a complete solution to an online bookstore-it meets all the requirements for its users and administrative objectives to stiff positioning on aspects of data correctness, security, and efficiency in reporting.

9. Conclusion

The development and deployment of the safety online bookshop application data-driven through ASP.NET Core 8 will be presented, and it is an important challenge of the electronic commerce industry: improving organization in sales tracking, handling a safe payment operation, and optimizing user interaction. It will ensure real-time access to data insight for administrators while allowing users to enjoy a smooth and safe buying experience. The application will be assured of strong data protection and operational dependability with the adoption of role-based access control, secure encrypted password storage and many safe payment options. Future research may extend the recommendation capability to possibly serve a larger number of users by incorporating more data processing methods and widening the selection of payment choices. These improvements will make the platform more adaptable in the competitive and changing online retailing environment, operational effectiveness, and increase user satisfaction.

Research Through Innovation

References

1. Kumar, G. S., & Jose, J. T. (2017). *Developing an Electronic Commerce Platform*. IEEE International Conference on Power, Control, Signals, and Instrumentation Engineering. <https://ieeexplore.ieee.org/Xplore/home.jsp>
2. ASP.NET Core Documentation. (2023). Microsoft. *Secure ASP.NET Core Identity*. <https://learn.microsoft.com/en-us/aspnet/core/security/authentication/identity?view=aspnetcore-8.0&tabs=visual-studio>
3. Reddy, R. (2012). *The Environmental Impacts of E-Commerce*. HuffPost UK. https://www.huffingtonpost.co.uk/robbie-reddy/the-environmental-impacts_b_1750438.html
4. Liu, L., & Wang, L. (2020). *Role-Based Access Control Applications*. International Journal of Web Information Systems, 16(3), 432-455. <https://www.emerald.com/insight/content/doi/10.1108/ijwis-04-2022-0077/full/pdf?title=a-systematic-literature-review-for-authorization-and-access-control-definitions-strategies-and-models>
5. Jalali, F., Nia, M. A., Ermakova, T., Abdollahi, M., & Fabian, B. (2021). Investigating the role of usable security in developers' intention toward security enhancement in service-oriented applications. *Security and Privacy*, 4(2), e199. <https://doi.org/10.1002/spy2.199>
6. El Aissoug, C., Kim, T.-Y., Wang, X.-Q., & Choi, D.-H. (2024). Cross-cultural exploration of determinants of e-commerce adoption: Perspectives from logistics service quality and payment driver. *Journal of Retailing and Consumer Services*, 81, 104033. <https://doi.org/10.1016/j.jretconser.2024.104033>
7. Abbas, N., & Farah, J. (2023). Optimizing e-commerce databases: A comparative analysis of SQL and NoSQL solutions. *ResearchGate*. <https://doi.org/10.13140/RG.2.2.22028.53121>
8. Zan, C. (2023). Development of e-commerce Big Data model based on machine learning and user recommendation algorithm. *International Journal of System Assurance Engineering and Management*. <https://doi.org/10.1007/s13198-023-02157-y>
9. National Institute of Standards and Technology (NIST). (2021). *Data Encryption Standard (DES) and Advanced Encryption Standard (AES)*. <https://csrc.nist.gov/pubs/fips/197/final>
10. Maixé-Altés, J. C. (2018). *Retail trade and payment innovations in the digital era: A cross-industry and multi-country approach*. Business History, 62(4), 588-612. <https://doi.org/10.1080/00076791.2018.1471062>

Research Through Innovation