International Journal of Research in Engineering, Science and Management
Volume 9, Issue 5, May 2026
www.ijresm.com | E-ISSN: 2581-5792 | RESAIM Publishing | www.resaim.com

BOOKVERSE: A Full-Stack Web-Based Book Marketplace with Collaborative Filtering Recommendation System

Shivesh Mishra Suman[1*], Rakesh Kumar Arora[2]
[1]Department of Computer Science and Engineering, Dr. Akhilesh Das Gupta Institute of Professional Studies, Delhi, India
[2]Professor, Department of Computer Science and Engineering, Dr. Akhilesh Das Gupta Institute of Professional Studies, Delhi, India

***Abstract*: The rapid growth of e-commerce platforms has transformed how users discover and purchase products online. However, traditional online bookstores often lack intelligent mechanisms for personalized recommendations. This paper presents BOOKVERSE, a full-stack web-based book marketplace integrating user-based collaborative filtering with real-time user interaction data. The system enables users to browse, search, and purchase books while generating personalized recommendations based on behavioral data such as browsing history and cart activity. A cosine similarity-based approach is used to identify similar users and recommend relevant books. The platform also includes a rule-based chatbot for assistance and an admin analytics dashboard for monitoring system performance. Experimental evaluation demonstrates a precision of 0.72 and an average response time of 0.35 seconds, indicating satisfactory performance. The proposed system enhances user engagement and improves the efficiency of book discovery.**

***Keywords*: Recommender Systems, Collaborative Filtering, Cosine Similarity, User Behavior Analysis, E-commerce, Web Application, Personalization.**

1. Introduction

The rapid growth of digital technologies and internet accessibility has transformed how users interact with information and perform online transactions. Online bookstores provide access to a vast collection of books; however, many existing platforms still function as static catalogs and lack intelligent mechanisms for personalized content discovery. This often leads to inefficient navigation, increased search effort, and reduced user satisfaction. With the continuous expansion of digital content, users face information overload, where traditional keyword-based search methods fail to deliver relevant results efficiently. To address this challenge, recommendation systems have been widely adopted to enhance user experience by analyzing behavioral data such as browsing history and cart activity. Among various techniques, collaborative filtering has proven effective in identifying similarity patterns between users using measures such as cosine similarity. However, many existing implementations focus primarily on standalone algorithms and lack integration within complete real-time web-based systems. Additionally, the effective utilization of dynamic user interaction data for continuously updating recommendations remains limited in practical applications. This paper presents BOOKVERSE, a full-stack web-based book marketplace that integrates a user-based collaborative filtering recommendation system. The system leverages real-time user interaction data to generate personalized recommendations efficiently. It also incorporates a rule-based chatbot for user assistance and an admin analytics dashboard for monitoring system performance.

The main contributions of this work are:

- Development of a full-stack e-commerce platform with integrated recommendation functionality.
- Implementation of user-based collaborative filtering using cosine similarity.
- Incorporation of activity-weighted user interaction data to improve recommendation relevance.
- Design of a real-time recommendation pipeline with efficient response time.

The proposed system aims to enhance personalization, improve user engagement, and simplify intelligent book discovery in modern e-commerce environments.

2. Literature Survey

Recent studies in recommendation systems and intelligent e-commerce platforms have focused on improving personalization and user engagement through data-driven approaches. Ricci F. et al. (2015) provide a comprehensive overview of recommender systems, highlighting collaborative filtering, content-based filtering, and hybrid techniques as key methodologies for generating personalized suggestions. [1] Adomavicius G. et al. (2005) discuss the evolution of recommender systems and emphasize the importance of combining multiple techniques to improve accuracy and scalability. [2] Koren Y. et al. (2009) introduce matrix factorization techniques that significantly enhance recommendation performance by capturing latent user preferences from large datasets. [3] Su X. et al. (2009) present a survey on collaborative filtering methods, identifying challenges such as sparsity and cold-start problems, which

*Corresponding author: mishrashiwesh1976@gmail.com

affect recommendation quality. [4] Linden G. et al. (2003) describe item-based collaborative filtering used in large-scale e-commerce platforms, demonstrating improved scalability and real-time recommendations. [5] Aggarwal C. (2016) explores data mining techniques for recommender systems, focusing on user behavior analysis and predictive modeling. [6] Bobadilla J. et al. (2013) analyze various similarity measures and evaluation metrics such as precision and recall for measuring recommendation effectiveness. [7] Schafer J. et al. (2007) examine the application of recommender systems in e-commerce and their role in enhancing user experience and sales. [8] In addition to recommendation systems, recent works also highlight the integration of intelligent features such as chatbots and analytics dashboards. Studies indicate that chatbot-based assistance improves user interaction, while analytics tools enable better decision-making through data visualization. Despite significant advancements, challenges such as limited datasets, scalability, and real-time processing remain open research areas. A comparative summary of existing approaches, along with their merits and limitations, is presented in Table 1.

Despite advancements in recommendation systems, several challenges remain in real-world deployment. Many approaches rely on complex models like matrix factorization and hybrid techniques, requiring large datasets and high computational resources. Issues such as data sparsity, cold-start problems, and limited handling of real-time user behavior further reduce effectiveness. Additionally, most studies focus on algorithms without integrating them into complete real-time web systems.

To address these limitations, the proposed BOOKVERSE system uses a lightweight user-based collaborative filtering approach with cosine similarity within a full-stack web architecture. It utilizes activity-weighted user data (views and cart actions) to generate efficient recommendations, ensuring low computational cost, faster response, and practical usability in real-time e-commerce environments.

3. Implementation

The BOOKVERSE system is implemented as a full-stack web-based e-commerce platform integrated with a real-time recommendation engine to enhance personalized user experience. The system combines standard marketplace functionalities with a user behavior–driven recommendation mechanism based on interaction data collected from platform usage. Unlike approaches that rely on external benchmark datasets, the proposed system utilizes dynamically generated data from user activities such as browsing, cart interactions, and purchases, making it adaptive and suitable for real-world deployment.

The implementation pipeline consists of dataset construction, data preprocessing, system architecture design, recommendation logic, and performance evaluation. A

Table 1
Comparison of research studies on recommendation systems and e-commerce

S.No.	Title of paper	Author	Key Findings	Merits	Demerits
1	Recommender Systems Handbook [1]	Ricci, F., Rokach, L., Shapira, B.	Provides comprehensive overview of recommendation techniques including collaborative and content-based filtering.	Covers wide range of methods and real-world applications.	Mostly theoretical; lacks practical implementation details.
2	Toward the Next Generation of Recommender Systems [2]	Adomavicius, G., Tuzhilin, A.	Discusses evolution of recommender systems and hybrid approaches.	Introduces advanced concepts for improving accuracy.	Complex models require high computational resources.
3	Matrix Factorization Techniques for Recommender Systems [3]	Koren, Y., Bell, R., Volinsky, C.	Proposes matrix factorization for capturing latent user preferences.	High accuracy and widely used in industry (e.g., Netflix).	Suffers from cold-start problem and data sparsity.
4	A Survey of Collaborative Filtering Techniques [4]	Su, X., Khoshgoftaar, T.	Reviews collaborative filtering methods and their challenges.	Provides comparative analysis of algorithms.	Limited discussion on real-time implementation.
5	Amazon Item-Based Collaborative Filtering [5]	Linden, G., Smith, B., York, J.	Introduces scalable item-based recommendation for e-commerce.	Highly scalable and suitable for large datasets.	Less personalized compared to user-based filtering.
6	Recommender Systems: The Textbook [6]	Aggarwal, C.	Explains data mining and recommendation algorithms in detail.	Strong theoretical foundation and algorithm coverage.	Requires deep mathematical understanding.
7	Recommender Systems Survey [7]	Bobadilla, J., Ortega, F., Hernando, A.	Discusses evaluation metrics like precision, recall, and accuracy.	Helps in performance evaluation of systems.	Does not focus on system implementation aspects.
8	E-commerce Recommendation Applications [8]	Schafer, J., Konstan, J., Riedl, J.	Explores role of recommender systems in e-commerce platforms.	Improves user engagement and sales.	Limited handling of dynamic user behavior.
9	Content-Based Recommendation Systems	Pazzani, M., Billsus, D.	Uses item features for generating recommendations.	Works well for new users (no cold-start).	Limited diversity; overspecialization problem.
10	Hybrid Recommendation Systems	Burke, R.	Combines collaborative and content-based methods.	Improves accuracy and robustness.	Increased system complexity.
11	Web Personalization using Recommendation Systems	Mobasher, B.	Uses web usage mining for personalization.	Enhances user experience through behavior tracking.	Requires large amount of user data.
12	Real-Time Recommendation Systems	Davidson, J. et al.	Focuses on scalable real-time recommendation (e.g., YouTube).	Handles large-scale data efficiently.	High infrastructure and computation cost

Table 2
Dataset composition

Dataset Type	Attributes Included	Purpose
Book Dataset	Title, Author, Genre, Price, Ratings	Book information storage
User Dataset	User ID, Preferences, Activity History	Personalization
Interaction Dataset	View, Cart, Purchases, Timestamp	Collaborative filtering logic

lightweight user-based collaborative filtering approach is adopted, where cosine similarity is applied to an activity-weighted user–item interaction matrix to identify similar users and generate recommendations. The system is designed using a modular architecture, enabling efficient backend processing, low response time, and seamless integration with frontend components through RESTful APIs. This design ensures scalability, simplicity, and real-time responsiveness.

A. Dataset Description

The BOOKVERSE system utilizes a combination of structured book metadata and dynamically generated user interaction data collected from platform usage. The dataset includes attributes such as book ID, title, author, category, price, and ratings, along with user behavioral data including views, cart actions, and purchase history. The initial dataset consists of approximately 10,000+ book records stored in a MySQL database. Unlike static datasets, user interaction data is continuously updated based on real-time activities, enabling the system to adapt to evolving user preferences. For efficient processing, the data is organized into three components: book metadata, user data, and interaction data. The interaction dataset forms the core of the recommendation system by capturing implicit user feedback. This modular structure supports efficient querying, scalability, and seamless integration with the recommendation engine.

B. Data Preprocessing

Data preprocessing is performed to ensure consistency and efficiency in recommendation generation. Book records are cleaned by removing duplicate entries and handling missing values to maintain data integrity. User interaction data is processed to assign weighted importance to different activity types, where actions such as “cart” and “purchase” are given higher significance compared to simple “views.” The processed interaction data is then structured into a user–item interaction matrix, where rows represent users and columns represent books. Each entry in the matrix corresponds to an activity-based score derived from user behavior. Missing interactions are represented as zero values.

Additionally, database-level optimizations such as indexing and efficient query design are applied to ensure fast data retrieval and real-time system responsiveness. This preprocessing step enables effective similarity computation and supports scalable recommendation generation.

Table 3
User–item interaction matrix representation

User & book	Book1	Book2	Book3
User1	5	0	2
User2	0	5	2

C. System Architecture Design

The BOOKVERSE system follows a modular three-tier architecture consisting of presentation, application, and data layers. The presentation layer is implemented using HTML, CSS, JavaScript, and Bootstrap to provide a responsive and interactive user interface. The application layer is developed using PHP, which handles core functionalities such as authentication, book management, cart operations, and API communication. A separate recommendation service is implemented using Python (Flask), which processes user interaction data and generates personalized recommendations. This service communicates with the main application through REST APIs, ensuring loose coupling and scalability. The data layer is managed using MySQL, which stores structured data including users, books, orders, and user activity logs. The database schema is designed using normalization techniques, with key relational tables such as Users, Products, Orders, Cart, and User_Activity interconnected via foreign keys. The system architecture integrates multiple modules including user management, product catalog, recommendation engine, chatbot interface, and admin analytics dashboard. This modular design ensures scalability, maintainability, and efficient data flow between components.

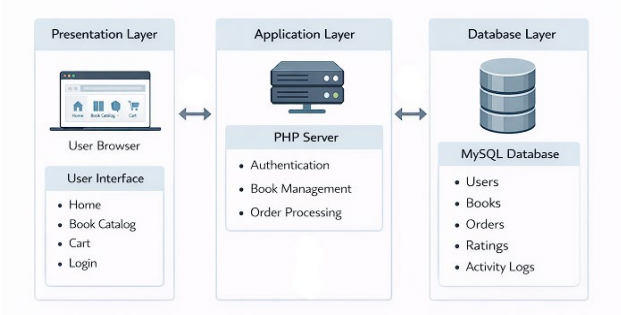

Fig. 1. Overall architecture of BOOKVERSE system

Table 4
System module description

Module Name	Functionality Description
User Module	Registration, login, profile management
Book Module	Book listing, search, filtering
Cart & Order Module	Cart management and order processing
Recommendation Module	Personalized book suggestions
Admin Module	Dashboard, analytics, inventory management
Chatbot Module	User assistance and query handling

D. Recommendation Logic

The execution of the BOOKVERSE system involves real-time interaction between the user interface and the recommendation engine through API calls. When a user interacts with the platform (e.g., views or adds books to cart), the activity is recorded in the database and used for generating recommendations. The recommendation mechanism is based on user-based collaborative filtering using cosine similarity. A user-item interaction matrix is constructed from activity logs, where different user actions are assigned weighted scores (e.g., cart = 5, view = 2, others = 1) to reflect user interest levels.

Table 5
Core recommendation logic and processing mechanism

Component	Logic
User Activity Processing	Assign weighted scores (view=2, cart=5, others=1)
Similarity Computation	Cosine similarity between users
Collaborative Filtering	Identify similar users and extract their preferred books
Recommendation Engine	Executes logic and SQL queries then filters Top-N books
Frontend Display	Renders UI dynamically and display recommendations

Cosine similarity is computed between users to identify similar user profiles. Books preferred by similar users are then recommended to the target user. Additionally, recent user activity is prioritized to ensure relevance in recommendations. To handle cold-start scenarios where user activity is limited or unavailable, the system falls back to random or popular book recommendations. The recommendation engine dynamically updates suggestions based on new interactions, ensuring adaptability to changing user preferences. The backend communicates with the recommendation engine via REST APIs, while asynchronous requests (AJAX) are used to fetch and display recommendations without affecting user experience.

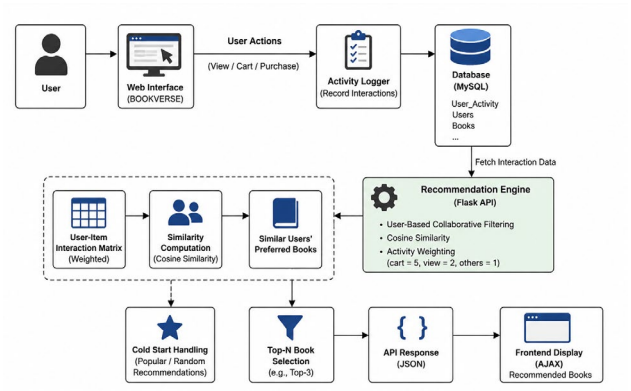

Fig. 2. System execution

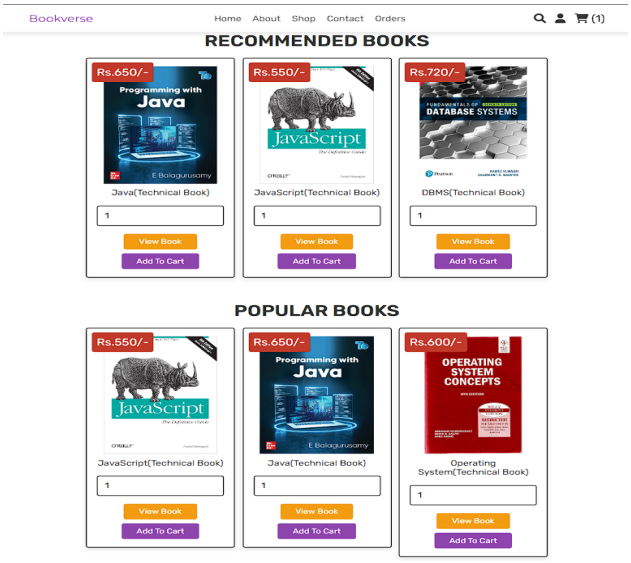

Fig. 3. Sample page showing recommendation

E. Performance Evaluation

The BOOKVERSE recommendation system was evaluated using precision, recall, and response time on user interaction data with Top-3 recommendations. It achieved a precision of 0.72 and recall of 0.65, indicating satisfactory performance for a lightweight approach. The average response time was 0.35 seconds, supporting real-time use. Efficient performance is enabled by activity-weighted data and optimized queries, though cold-start and sparse data issues limit recall and suggest scope for future improvements.

Table 6
Recommendation system performance metrics

Metric	Value
Precision	0.72
Recall	0.65
Average Response Time	**0.35 seconds**

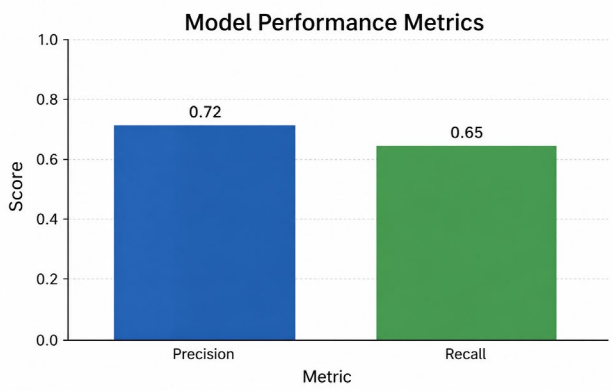

Fig. 4. Precision vs Recall

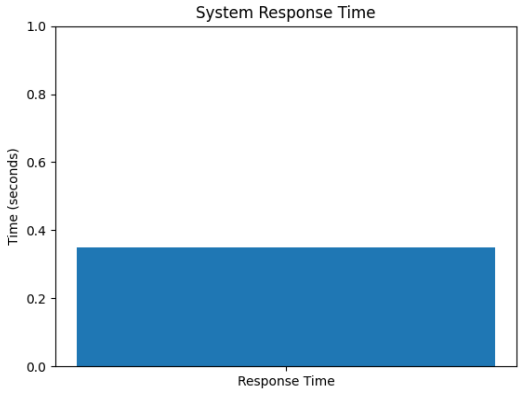

Fig. 5. Response time

4. Conclusion

The BOOKVERSE system is a full-stack web-based book marketplace integrated with a user-based collaborative filtering recommendation engine to enhance personalization. It uses book metadata and user interaction data such as browsing history, cart activity, and purchases to generate relevant suggestions. Key modules include user management, book catalog, chatbot support, and an analytics dashboard.

The recommendation model using cosine similarity achieved a precision of 0.72 and recall of 0.65, indicating satisfactory performance for a lightweight system. With an average response time of 0.35 seconds, the system provides efficient real-time recommendations, improving user engagement and enabling scalable intelligent e-commerce solutions.

References

[1] F. Ricci, L. Rokach, and B. Shapira, *Recommender Systems Handbook*. New York, NY, USA: Springer, 2015.

[2] G. Adomavicius and A. Tuzhilin, "Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions," *IEEE Trans. Knowl. Data Eng.*, vol. 17, no. 6, pp. 734–749, Jun. 2005.

[3] Y. Koren, R. Bell, and C. Volinsky, "Matrix factorization techniques for recommender systems," *Computer*, vol. 42, no. 8, pp. 30–37, Aug. 2009.

[4] X. Su and T. M. Khoshgoftaar, "A survey of collaborative filtering techniques," *Adv. Artif. Intell.*, vol. 2009, Art. ID 421425, 2009.

[5] G. Linden, B. Smith, and J. York, "Amazon.com recommendations: Item-to-item collaborative filtering," *IEEE Internet Comput.*, vol. 7, no. 1, pp. 76–80, Jan.–Feb. 2003.

[6] C. C. Aggarwal, *Recommender Systems: The Textbook*. Cham, Switzerland: Springer, 2016.

[7] J. Bobadilla, F. Ortega, A. Hernando, and A. Gutiérrez, "Recommender systems survey," *Knowl.-Based Syst.*, vol. 46, pp. 109–132, Jul. 2013.

[8] J. B. Schafer, J. Konstan, and J. Riedl, "E-commerce recommendation applications," *Data Min. Knowl. Discov.*, vol. 5, no. 1–2, pp. 115–153, 2001.

[9] M. J. Pazzani and D. Billsus, "Content-based recommendation systems," in *The Adaptive Web*. Berlin, Germany: Springer, 2007, pp. 325–341.

[10] R. Burke, "Hybrid recommender systems: Survey and experiments," *User Model. User-Adapt. Interact.*, vol. 12, no. 4, pp. 331–370, 2002.

[11] B. Mobasher, "Data mining for web personalization," in *The Adaptive Web*. Berlin, Germany: Springer, 2007, pp. 90–135.

[12] J. Davidson *et al*., "The YouTube video recommendation system," in *Proc. 4th ACM Conf. Recommender Syst.*, 2010, pp. 293–296.

[13] J. Shobana, C. Gangadhar, R. K. Arora, *et al*., "E-commerce customer churn prevention using machine learning-based business intelligence strategy," *Measurement: Sensors*, vol. 27, 2023.

[14] R. K. Arora and M. K. Gupta, "e-Governance using data warehousing and data mining," *Int. J. Comput. Appl.*, vol. 169, no. 8, 2017.

[15] S. Kumar and R. K. Arora, "Analyzing customer behaviour through data mining," *Int. J. Comput. Appl. Technol. Res.*, vol. 4, 2015.

[16] R. K. Arora, M. K. Gupta, and S. P. Singh, "Analysis of various smart healthcare systems for heart disease prediction," in *Proc. Int. Conf. Mach. Learn. Adv. Comput.*, 2023.

[17] M. K. Gupta, R. K. Arora, V. Khatri, P. Panchal, L. K. Meena, and A. M. D. Yahya, "Personalized healthcare systems using AI," *Nanotechnol. Perceptions*, pp. 437–452, 2024.