Innovation and Integrative Research Center Journal
ISSN: 2584-1491 | www.iircj.org
Volume-4 | Issue-5 | May-2026 | Page 412-417

# BOOKXCHANGE: MERN-Based Book Resale Platform

[1]Mr. Shiv Kumar, [2]Mr. Abhishek Kumar, [3]Dr. Poonam Mishra, [4]Dr. Goldi Soni
[1,2]Student, [3]Assosiate Professor, [4]Assistant Professor
[1,2,3,4]Amity University Chhattisgarh

## ABSTRACT

The increasing cost of books and the underutilization of previously owned resources have created a significant need for efficient and affordable resale platforms. This paper presents BookXchange, a MERN-based web application designed to facilitate the buying and selling of used books while enhancing user experience through an intelligent recommendation system. The platform leverages modern web technologies, including MongoDB, Express.js, React.js, and Node.js, to provide a scalable and responsive architecture. Additionally, Firebase Authentication is integrated to ensure secure user management and data protection. A key feature of the system is a content-based recommendation engine, which analyzes book attributes such as title, category, and description. These attributes are transformed into numerical vectors using TF-IDF (Term Frequency–Inverse Document Frequency), a widely used technique in information retrieval for identifying significant terms [1]. Similarity between books is computed using cosine similarity, enabling the system to recommend items that are contextually relevant [2]. Unlike collaborative filtering methods, which rely heavily on user interaction data, this approach performs effectively even in data-sparse environments. To address the cold start problem, the system incorporates a fallback mechanism that recommends trending or default books for new users. Experimental evaluation demonstrates that the system achieves efficient performance, low latency, and relevant recommendations. By integrating web development with machine learning techniques, BookXchange provides a practical and scalable solution for intelligent book resale and discovery.

## 1. INTRODUCTION

The rapid growth of internet technologies has transformed traditional commerce into digital platforms, enabling users to access products and services with greater convenience. In the domain of books, online marketplaces have significantly improved accessibility by offering vast collections and easy purchasing options. However, most existing platforms primarily focus on selling new books, which can be expensive for students and budget-conscious users. At the same time, a large number of used books remain unused after their initial purpose, leading to inefficient resource utilization.

One of the major challenges in online platforms is content discoverability. As the volume of available books increases, users often struggle to find items that match their interests. Recommendation systems play a crucial role in addressing this issue by providing personalized suggestions. Traditional approaches such as collaborative filtering rely on user interaction data, including ratings and purchase history. While effective in large-scale systems, these methods

ISSN: 2584-1491 | www.iircj.org
Volume-4 | Issue-5 | May-2026 | Page 412-417

suffer from limitations such as the cold start problem, where new users or items lack sufficient data for generating recommendations [3].

In contrast, content-based filtering offers an alternative approach by analyzing item attributes rather than user behavior. This method is particularly suitable for systems with limited data, as it does not require extensive user interaction history. By leveraging textual features and similarity measures, content-based systems can generate meaningful and explainable recommendations [4].

This paper presents BookXchange, a MERN-based book resale platform integrated with a content-based recommendation system. The platform enables users to buy and sell used books, promoting affordability and sustainability. The system incorporates Firebase Authentication to ensure secure user access and utilizes a FastAPI-based microservice for handling recommendation logic independently.

The key contributions of this work include:

- Development of a scalable resale platform
- Implementation of a content-based recommendation system
- Integration of secure authentication mechanisms
- Handling of the cold start problem

## 2. LITERATURE REVIEW

The development of recommendation systems and online book platforms has been widely explored in the domains of e-commerce and information retrieval. Large-scale platforms such as Amazon and Flipkart utilize sophisticated recommendation systems to enhance user engagement and increase sales. These systems primarily rely on collaborative filtering techniques, which analyze user behavior such as ratings, browsing history, and purchase patterns to generate recommendations. While effective in environments with extensive user data, collaborative filtering suffers from limitations such as data sparsity and the cold start problem, where insufficient data leads to inaccurate recommendations [3].

To overcome these limitations, researchers have explored content-based recommendation systems, which focus on analyzing item attributes rather than user interactions. According to Pazzani and Billsus, content-based filtering generates recommendations by comparing item features and identifying similarities based on user preferences [4]. This approach is particularly effective in systems with limited user data, making it suitable for small-scale or newly developed platforms.

A fundamental technique used in content-based filtering is TF-IDF (Term Frequency–Inverse Document Frequency). This method assigns weights to words based on their importance within a document and across a collection of documents. Words that appear frequently in a document

ISSN: 2584-1491 | www.iircj.org
Volume-4 | Issue-5 | May-2026 | Page 412-417

but rarely across other documents are given higher importance, enabling the system to identify distinguishing features [1]. TF-IDF has been widely used in text mining and information retrieval applications.

To measure similarity between items, cosine similarity is commonly employed. This technique calculates the cosine of the angle between two vectors in a high-dimensional space, providing a measure of how similar the items are [2]. Cosine similarity is computationally efficient and particularly effective for sparse data, making it suitable for text-based recommendation systems.

Platforms focused on second-hand books, such as Bookchor, address affordability by enabling users to purchase used books at lower prices. However, these platforms often lack intelligent recommendation systems, relying instead on manual search and filtering. This limits their ability to provide personalized user experiences.

Recent advancements in software architecture have introduced microservice-based designs, where different components of a system are developed and deployed independently. This approach improves scalability and maintainability, allowing systems to handle increased workloads efficiently [5]. In recommendation systems, deploying the recommendation engine as a separate microservice enables independent optimization and scalability.

The proposed BookXchange system builds upon these concepts by integrating a content-based recommendation engine within a resale platform, addressing both affordability and personalization.

## 3. SYSTEM ARCHITECTURE

The BookXchange platform is designed using a multi-tier architecture that integrates frontend, backend, database, authentication, and recommendation components. This modular design ensures scalability, flexibility, and efficient performance.

The frontend is developed using React.js, which provides a dynamic and responsive user interface. React's component-based architecture enables efficient rendering and improves maintainability. Users can browse books, manage listings, and interact with recommendations through an intuitive interface.

The backend is implemented using Node.js and Express.js. Node.js provides an event-driven, non-blocking architecture that allows the system to handle multiple concurrent requests efficiently. Express.js is used to define API routes and manage server-side logic. The backend acts as an intermediary between the frontend and the database, processing user requests and ensuring proper data flow.

The system uses MongoDB as the database, which offers flexibility in data storage and efficient handling of unstructured data [6]. Collections such as users, books, and orders are maintained, enabling efficient retrieval and management of information.

Innovation and Integrative Research Center Journal
ISSN: 2584-1491 | www.iircj.org
Volume-4 | Issue-5 | May-2026 | Page 412-417

To ensure secure access, the system integrates Firebase Authentication, which provides features such as user registration, login, and token-based authentication. This enhances security and protects user data.

A key component of the architecture is the recommendation system, implemented as a FastAPI microservice. This microservice operates independently from the main application, receiving requests from the backend and returning recommendations. The microservice architecture improves scalability and allows the recommendation system to be updated without affecting other components [5].

The overall workflow involves the frontend sending requests to the backend, which processes them and interacts with the database or recommendation service as needed. The response is then returned to the frontend for display.

## 4. METHODOLOGY

The development of BookXchange follows an Agile methodology, which emphasizes iterative development and continuous improvement [7]. This approach allows for flexibility in incorporating new features and adapting to changing requirements.

The process begins with requirement analysis, where system objectives such as resale functionality, authentication, and recommendation features are defined. This is followed by system design, which includes defining architecture, API endpoints, and database schemas.

During implementation, the frontend and backend are developed in parallel, while the recommendation system is implemented as a separate microservice. This modular approach allows independent development and testing of each component.

The recommendation system relies on data stored in the database, including book attributes such as title, category, and description. This data undergoes preprocessing, which involves cleaning, tokenization, and normalization. Stop words are removed to improve the quality of the data.

Feature extraction is performed using TF-IDF vectorization, which converts textual data into numerical vectors [1]. These vectors are then used for similarity computation.

The system integrates all components using RESTful APIs, ensuring efficient communication between frontend, backend, and microservice.

Testing is conducted at multiple levels, including unit testing, integration testing, and system testing. This ensures that all components function correctly and that the system performs efficiently.

## 5. RECOMMENDATION SYSTEM

The recommendation system in BookXchange is based on content-based filtering, which analyzes item attributes to generate recommendations. Each book is represented as a combination of its textual attributes, including title, category, and description.

ISSN: 2584-1491 | www.iircj.org
Volume-4 | Issue-5 | May-2026 | Page 412-417

The system applies TF-IDF vectorization to convert text into numerical vectors, assigning higher weights to important terms [1]. This representation enables the system to capture the semantic meaning of each book.

Similarity between books is computed using cosine similarity, which measures the angle between vectors [2]. Books with higher similarity scores are considered more relevant and are recommended to users.

The recommendation process involves retrieving user history, computing similarity scores, and selecting the most relevant books. Already purchased items are excluded to avoid redundancy.

To address the cold start problem, the system provides fallback recommendations such as trending or default books. This ensures that all users receive meaningful suggestions.

The recommendation system is implemented as a FastAPI microservice, enabling independent scaling and efficient computation.

## 6. RESULTS AND ANALYSIS

The BookXchange system was evaluated based on performance, recommendation quality, and user experience. The results indicate that the system performs efficiently, with low response times and reliable data processing.

The use of Node.js and MongoDB ensures fast API responses and efficient data handling. The FastAPI microservice enables quick computation of recommendations without affecting the main application.

The recommendation system provides relevant suggestions by analyzing content similarity. Users with purchase history receive personalized recommendations, while new users benefit from fallback suggestions.

The system improves user experience by providing an intuitive interface and reducing the effort required to find relevant books.

## 7. CONCLUSION

BookXchange successfully integrates full-stack web development with machine learning techniques to create an intelligent book resale platform. The system addresses key challenges such as affordability, content discoverability, and personalization. The use of content-based filtering ensures that recommendations are meaningful and effective even with limited data. The modular architecture enhances scalability and maintainability. Overall, the system demonstrates the practical application of modern technologies in developing intelligent web-based platforms.

ISSN: 2584-1491 | www.iircj.org
Volume-4 | Issue-5 | May-2026 | Page 412-417

**REFERENCES**

[1] G. Salton and C. Buckley, “Term-weighting approaches,” 1988

[2] A. Géron, Hands-On Machine Learning, 2019

[3] B. Sarwar et al., Collaborative Filtering, 2001

[4] Pazzani & Billsus, Content-Based Systems, 2007

[5] FastAPI Documentation

[6] MongoDB Documentation

[7] Agile Methodology