ReadSphere – Your Reading World in One Website

Chaitali Mhatre
Department of Computer Engineering
Universal College of Engineering
Kaman, Mumbai, India
patilchaitali1333@gmail.com

Jaykumar Waghela
Department of Computer Engineering
Universal College of Engineering
Kaman, Mumbai, India
waghelajay305@gmail.com

Roshni Yadav
Department of Computer Engineering
Universal College of Engineering
Kaman, Mumbai, India
roshni022004@gmail.com

Prakashkumar Yadav
Department of Computer Engineering
Universal College of Engineering
Kaman, Mumbai, India
prakashkumarydv3@gmail.com

Rahul Yadav
Department of Computer Engineering
Universal College of Engineering
Kaman, Mumbai, India
ry8055528@gmail.com

***Abstract*—The rapid growth of online libraries has resulted in information overload, making it challenging for readers to identify books that match their interests through traditional search methods. This paper presents ReadSphere, a web-based book discovery platform designed to simplify and personalize the reading experience. The system is developed using modern technologies including Next.js, React, Tailwind CSS, TypeScript, and a Supabase PostgreSQL database.The core of the platform is a mood-based recommendation approach that combines rule-based mapping, keyword analysis, and basic semantic understanding to generate relevant book suggestions. In addition to recommendation functionality, the system incorporates secure authentication using Clerk and an AI-assisted chatbot to provide quick book summaries and assist in exploration.The platform also includes reading activity tracking and dashboard-based analytics, enabling users to monitor their reading patterns over time. The proposed approach demonstrates an effective way to enhance user engagement and simplify book discovery without relying on computationally intensive machine learning models.**

***Index Terms*—Book Recommendation, Personalized Systems, Digital Libraries, Mood-Based Filtering, Reading Analytics, Web Application.**

I. INTRODUCTION

The rapid growth of digital libraries and online reading platforms has significantly increased the availability of content, making it difficult for users to efficiently discover books that match their interests. This issue, commonly referred to as information overload, reduces the effectiveness of traditional search methods and negatively impacts user engagement. Recommendation systems have been widely adopted to address this challenge by providing personalized suggestions based on user preferences and behavior [1]–[4]. However, many existing approaches focus primarily on improving algorithmic accuracy using complex machine learning models, while often neglecting usability, efficiency, and real-time interaction [5].

In addition, most systems rely on static user data and fail to consider dynamic factors such as a user's current mood or intent. Reading preferences are often influenced by emotional context. As a result, relying only on static user data may lead to less relevant recommendations and increased decision fatigue. Furthermore, the lack of integrated features such as interactive assistance and reading analytics leads to fragmented user experiences [6, 7].

To address these limitations, this paper proposes *ReadSphere*, a web-based platform that enhances book discovery through a hybrid recommendation approach combining mood-based mapping and content-based filtering, with a focus on improving user engagement while maintaining system efficiency and practical usability.

II. LITERATURE REVIEW

A. Existing Systems and Limitations

Digital reading platforms and recommendation systems have evolved significantly in recent years; however, several limitations still persist. Traditional platforms often rely on manual browsing and keyword-based search, which contributes to information overload and reduces the efficiency of content discovery [1, 7, 9]. Many existing systems are built on conventional architectures that lack scalability and efficient data interaction, leading to slower performance and limited adaptability.

In addition, several platforms focus primarily on backend optimization while neglecting user-centric design, resulting in poor interface experience and reduced usability. Another key limitation is the absence of integrated engagement features such as reading activity tracking, personalized dashboards, and interactive assistance. As a result, users often experience decision fatigue and decreased reading retention in digital reading environments [10]–[12].

B. Recommendation Techniques and Related Work

Recent research has explored various techniques to enhance recommendation accuracy, including machine learning models, deep learning approaches, collaborative filtering, and system modeling using UML-based architectures [3, 4, 13, 14]. Advanced approaches such as conversational recommendation systems and memory-augmented recommender architectures

TABLE I: Literature Survey

Ref.	Paper Name	Year	Author(s)	Publication	Proposed Work	Research Gap
[6]	Towards Next-Gen Recommender Systems	2026	Huang et al.	Proc. ACM WSDM	Evaluates conversational AI assistants for complex book searches	Lacks integrated platform, authentication, mood-based personalization, and user activity tracking.
[8]	From Textbook to Screen	2026	Aghayeva and Banerji	Sustainable Global Soc. Init.	Explores digital reading impact on comprehension and motivation	Focuses on reading behavior analysis but lacks practical implementation and real-time features.
[5]	Recommender for Learning Contents	2025	Sabateen and Itmazi	Ahliya J. Bus. Technol.	Suggests learning materials using demographics and rules	Limited to educational environments; lacks mood-based recommendations and conversational interfaces.
[3]	AI-Based Recommendation System	2024	Alomran and Basha	Scalable Comput. Pract. Exp.	Uses ML models (Neuro-Fuzzy, SVM) for classification	Lacks user engagement features such as analytics dashboards and chatbot integration.
[7]	Analysis of Digital Reading Platform	2022	Zhang et al.	Wireless Commun. Mob. Comput.	Uses big data and cloud for digital libraries	Backend-focused; limited support for personalized recommendations and user interaction features.

have also been proposed to improve personalization and contextual recommendations [6, 15]. While these techniques demonstrate strong predictive capabilities, many of them are computationally intensive, require large-scale datasets, and are difficult to deploy in real-time web applications. This makes them less suitable for lightweight, scalable, and user-focused systems.

Additionally, several systems remain limited to specific domains, such as educational platforms, and do not generalize well to broader reading applications [5]. Another limitation observed in existing studies is the lack of integration between recommendation engines and user engagement features.To address these challenges, the proposed system, *ReadSphere*, adopts a hybrid recommendation approach that combines rule-based mood mapping with content-based filtering, while also integrating features such as an AI-assisted chatbot and reading analytics to provide a more practical and user-centered solution. The analysis presented in Table I, indicates that although existing systems employ various recommendation techniques, they often lack integration of multiple functionalities within a single platform. Most approaches focus either on algorithmic accuracy or backend processing, while neglecting user-centric features such as mood-based filtering, interactive assistance, and reading analytics. This observation highlights a clear research gap in developing a unified, efficient, and user-oriented book recommendation system.

III. PROPOSED SYSTEM

A. System Overview

ReadSphere is a modern digital reading platform designed to address information overload by enabling efficient and personalized book discovery and reading management. The system is developed using a robust and scalable technology stack, including Next.js and React for the frontend interface, Tailwind CSS for styling, and TypeScript for improved code reliability. Backend operations and data storage are managed using a Supabase PostgreSQL database, while user session security is handled through Clerk authentication.

The platform integrates mood-aware recommendation logic, continuous reading activity tracking, and an AI-assisted chatbot to provide a unified and interactive user experience. The hybrid recommendation engine avoids computationally intensive machine learning or collaborative filtering techniques by utilizing optimized database queries and rule-based mappings. This lightweight approach ensures low computational overhead, faster response times, and efficient real-time interaction. Additionally, visual dashboard analytics and AI-based summarization help reduce decision fatigue and improve overall user engagement.

B. Key Features

- **Hybrid Recommendation Engine:** The system leverages a hybrid of rule-based mapping and content-based filtering to generate mood-aware recommendations efficiently without requiring complex machine learning models.
- **Dynamic Mood-to-Genre Mapping:** Users manually select their mood, which the system internally maps to specific genres using a predefined rule-based table.
- **AI Chatbot for Summarization:** An integrated AI chatbot is dedicated exclusively to generating instant book summaries, allowing users to quickly understand a book's content before reading.
- **Reading Tracking and Analytics:** ReadSphere actively monitors user progress by providing reading activity tracking and visual dashboard analytics, enabling users to better understand their reading habits.
- **Modern Architecture:** The platform is built using a Next.js and React frontend styled with Tailwind CSS, secured by Clerk authentication, and backed by a Supabase PostgreSQL database.

C. System Architecture

The system architecture of ReadSphere consists of multiple interconnected components that work together to deliver a seamless and efficient book recommendation experience. Figure 1, illustrates the overall architecture and data flow between system modules. The design follows a modular and scalable approach, enabling easy maintenance and future enhancements.

The working of the system is explained in the following steps:

Step 1: User Interaction The process begins at the Web Application layer, where users interact with the platform through a responsive interface developed using Next.js, React,

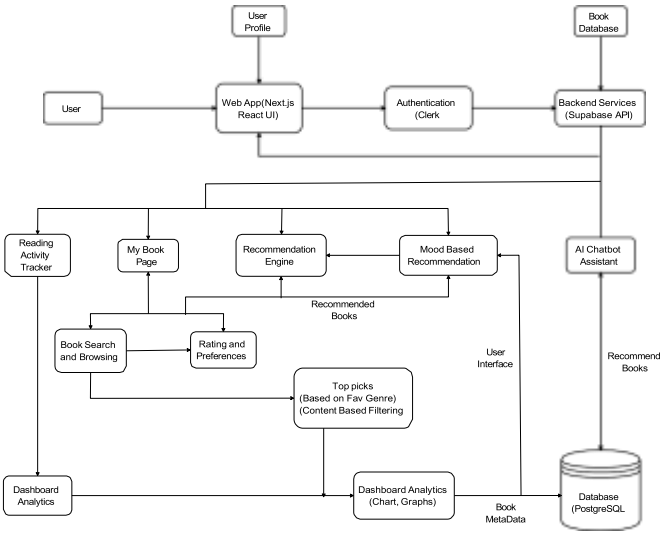

Fig. 1: System Architecture of ReadSphere

and Tailwind CSS. Users can select moods, browse books, and access personalized features.

Step 2: Authentication and Session Management User authentication is handled by Clerk, which securely manages login sessions and ensures safe access to personalized services.

Step 3: Mood Input and Processing The user selects a mood (such as happy, stressed, or curious) through the interface. This input is passed to the backend system for further processing.

Step 4: Mood-to-Genre Mapping The system applies a predefined rule-based mapping mechanism that converts the selected mood into corresponding literary genres. For example, a ”stressed” mood may be mapped to self-help or light fiction categories.

Step 5: Recommendation Generation Based on the mapped genres, the system performs content-based filtering by executing optimized SQL queries on the Supabase PostgreSQL database. This step retrieves books that match the selected criteria.

Step 6: Data Processing and Ranking The retrieved book results are displayed through the user interface, while user reading history and activity are tracked and visualized on the dashboard.

Step 7: Output and Visualization The recommended books are presented through the user interface, allowing users to explore and select content based on their preferences. At the same time, user reading history and activity are tracked and visualized on the dashboard using analytical tools.

Step 8: AI Chatbot Assistance An integrated AI chatbot operates alongside the recommendation system, providing real-time book summaries and assisting users in exploring content efficiently.

This architecture ensures efficient data handling, low computational overhead, and real-time responsiveness, making the system suitable for scalable web-based deployment.

IV. METHODOLOGY

A. Activity Flow

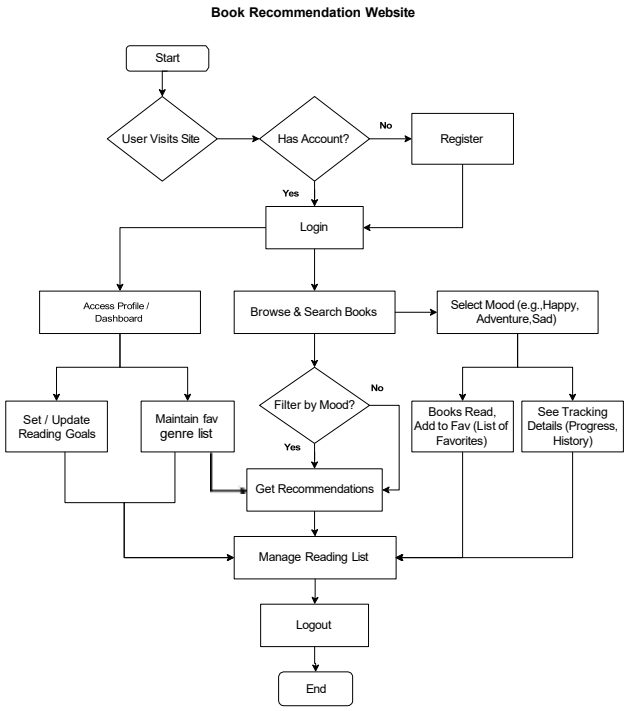

Fig. 2: Activity Flow Diagram

The operational flow of ReadSphere, as illustrated in the Figure 2, represents the complete user interaction lifecycle within the system. The process begins with a secure authentication phase, where the system differentiates between new and returning users. This ensures that all user interactions

are associated with a persistent profile, enabling personalized recommendations and activity tracking.

After successful login, the user is directed to the main interface, where the workflow is divided into two primary paths. The first path focuses on user engagement, allowing individuals to set reading goals, track progress, and monitor reading statistics through a dashboard. The second path focuses on content discovery, where users select a mood (such as Happy, Adventurous, or Relaxed), which acts as the primary input for the recommendation system.

Based on the selected mood, the system retrieves relevant book suggestions from the database using predefined mappings and filtering techniques. The user can then explore recommended books, manage favorites, and update reading progress. This cyclical interaction enables continuous personalization and improves user engagement over time.

B. Recommendation Approach

The recommendation approach in ReadSphere is designed to be efficient, lightweight, and responsive. The system uses a dual-input mechanism to capture user preferences through emoji-based mood selection and keyword-based text analysis. The emoji interface allows users to quickly express their emotional state using predefined icons. Each emoji is mapped to a specific mood category using a rule-based mapping system, simplifying interaction and improving usability.For text input, the system applies a heuristic keyword-matching approach. The user text is normalized and compared with a predefined Mood-Keyword Matrix, where a matching score is calculated for each mood based on keyword frequency. The mood with the highest score is selected.

The detected mood is then passed to the recommendation engine, which retrieves relevant books using content-based filtering.Additionally, the system supports direct genre-based filtering, where users can manually select a genre and optionally an author to receive book suggestions. This input is processed through direct database queries, enabling fast and accurate results.

C. Algorithms Used

The ReadSphere system employs lightweight and efficient algorithms to perform mood detection and book recommendation without relying on computationally intensive machine learning models. The overall approach focuses on rule-based processing and optimized database querying to ensure real-time performance and scalability.

Algorithm 1: Keyword-Based Mood Detection

This algorithm detects the user's mood from textual input using keyword matching.

Input: User text T **Output:** Detected mood M

```
BEGIN
Convert T to lowercase
bestMood <- null
bestScore <- 0
FOR each mood in moodKeywords:
    score <- 0
    FOR each keyword in moodKeywords[mood]:
        IF keyword exists in T:
            score <- score + 1
    IF score > bestScore:
        bestScore <- score
        bestMood <- mood
IF bestMood is not null:
    RETURN bestMood
ELSE:
    RETURN "unknown"
END
```

Algorithm 2: Mood-Based Book Recommendation

This algorithm recommends books by mapping the detected mood to genres and retrieving matching books from the database.

Input: Mood M **Output:** Recommended books R

```
BEGIN
G <- emotionGenreMap[M]
B <- SELECT * FROM books WHERE genre IN G
R <- first 8 books from B
RETURN R
END
```

V. IMPLEMENTATION

- **Development Environment and Technologies:** ReadSphere is developed using a modern JavaScript and TypeScript-based technology stack. The frontend is implemented with Next.js and React to support efficient rendering and responsive user interaction, while Tailwind CSS is used for interface styling. Backend services and database management are handled by Supabase using PostgreSQL, and secure authentication and session management are implemented through Clerk.
- **Frontend and Dashboard Implementation:** The user interface is built using modular React components, including the reading dashboard, mood selection interface, and activity tracker. State management mechanisms handle dynamic updates such as reading progress, streak tracking, and activity logs. The dashboard visualizes reading history and analytics, while recommended books are retrieved through API responses.
- **Recommendation Engine Execution:** The recommendation engine follows a lightweight hybrid approach implemented through server-side API routes. User mood input is mapped to predefined genres, and optimized SQL queries retrieve matching books from the Supabase PostgreSQL database. The results are processed and sorted based on attributes such as popularity and recency before being displayed to the user.
- **AI Chatbot Integration:** An AI chatbot is integrated to provide instant book summaries. The chatbot responds to book-related queries and generates concise descriptions of selected titles, allowing users to quickly understand book content and make informed reading decisions.

VI. RESULT AND DISCUSSION

This section presents the practical implementation results of the ReadSphere platform along with an analysis of system performance and user interaction features.

A. System Interface and AI Chatbot

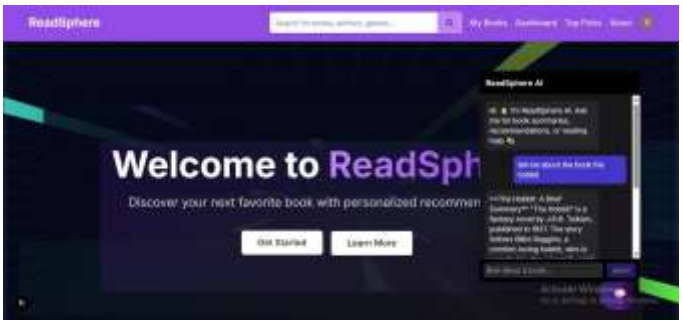

Fig. 3: ReadSphere homepage interface with AI chatbot.

The homepage of ReadSphere provides an intuitive interface for exploring the platform's features. As shown in Fig. 3, it integrates book discovery tools with an AI-powered chatbot that helps users understand book content before reading. The chatbot generates concise summaries of selected books, allowing users to quickly evaluate their relevance. This feature reduces the effort required to analyze book descriptions manually. Additionally, the structured layout enables easy access to recommendation modules, trending books, and personalized reading features, improving overall navigation and user engagement.

B. Mood-Based Recommendation System

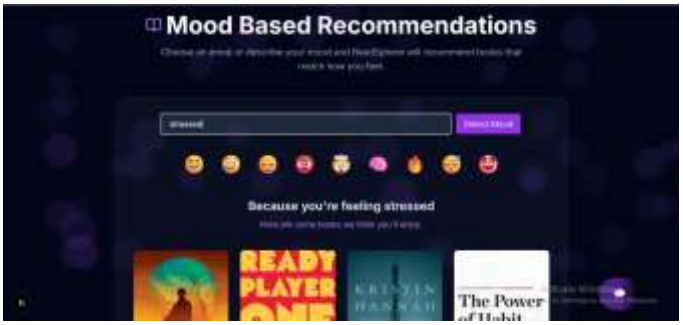

Fig. 4: Mood-based recommendation interface.

ReadSphere introduces a mood-driven recommendation mechanism that enables users to discover books based on their current emotional state. As shown in Fig. 4, users select a mood such as happy, curious, or relaxed, which the system maps to predefined genres stored in the database. The recommendation engine then retrieves relevant books using content-based filtering through optimized SQL queries. This approach avoids complex machine learning models while still providing personalized recommendations in real time. The mood-selection interface simplifies the book discovery process and helps users find suitable reading material quickly.

C. Trending Books Section

Fig. 5: Trending books section displaying popular titles on the platform.

The trending books module highlights popular books that are currently receiving high user engagement on the platform. As shown in Fig. 5, the system dynamically displays books that are frequently viewed or recently added to the database. This feature allows users to quickly explore widely read or recommended titles without performing manual searches. By presenting books that are gaining attention among readers, the module helps users stay updated with popular reading trends and encourages exploration of new titles across different genres.

D. Genre-Based Book Exploration

Fig. 6: Genre-based browsing interface.

In addition to mood-based recommendations, ReadSphere also enables users to explore books through a genre-based browsing interface. As illustrated in Fig. 6, the platform categorizes books into multiple genres such as fiction, mystery, romance, and science fiction. Users can directly navigate to a preferred genre and view curated book suggestions within that category. This structured organization simplifies the discovery process and allows users to locate books that match their reading interests more efficiently. The genre exploration module complements the recommendation system by providing an alternative method for browsing the available book collection.

E. Reading Dashboard and Analytics

The ReadSphere platform includes a personalized reading dashboard that allows users to monitor their reading habits and progress. As shown in Fig. 7, the dashboard displays important information such as yearly reading goals, preferred genres, and the number of books completed.

Fig. 7: Reading analytics and activity tracking

The system also visualizes reading activity through graphical analytics, including daily reading time trends and an activity heatmap that highlights user engagement patterns. These visual indicators help users track their reading consistency and maintain motivation through measurable progress. By combining reading analytics with goal tracking, the dashboard transforms the platform into an interactive environment that encourages long-term reading engagement.

The performance evaluation of the proposed system indicates that it operates efficiently under typical usage conditions. The average search latency was measured at approximately 134 ms, enabling quick retrieval of results. Book recommendations based on selected genres were generated within 0.77 s, while the mood-to-genre mapping process completed in about 2.32 s. The analytics dashboard, which displays user reading activity and insights, loaded within 4 s, ensuring smooth visualization without noticeable delays.

Furthermore, the system maintained stable performance during repeated interactions, with an average image rendering time of 16 ms across 124 requests. These observations suggest that the platform is capable of handling real-time user interactions effectively, while providing a responsive and seamless experience for book discovery and reading activity tracking.

VII. CONCLUSION AND FUTURE WORK

A. Conclusion

The development of ReadSphere aims to improve the digital book discovery experience by providing a modern web-based platform for exploring books, receiving personalized mood-aware recommendations, and tracking reading activity. The system is built using technologies including Next.js, React, Tailwind CSS, Supabase (PostgreSQL), and Clerk authentication, ensuring a secure and responsive user experience. By combining rule-based mood mapping with content-based filtering, ReadSphere generates relevant book recommendations without relying on computationally heavy machine learning models. Additionally, the integration of reading analytics and an AI chatbot for instant book summarization simplifies book discovery and encourages consistent reading habits. Overall, the platform provides a balanced solution that focuses on usability, efficiency, and user engagement, making digital reading more interactive and accessible.

B. Future Work

To further enhance the functionality and user experience of the ReadSphere platform, the following improvements can be considered for future iterations:

- **Gamified Reading Features:** Introducing badges, leaderboards, and point-based systems alongside reading streaks to reward regular readers and motivate habit building.
- **Multi-Language Support:** In future, the platform can support multiple languages, allowing users to receive recommendations in their preferred language and improving accessibility.
- **Cross-Platform Synchronization:** Enabling syncing with e-readers, mobile applications, and browser extensions to provide a seamless, multi-device reading experience.
- **Social and Community Features:** Adding interactive elements such as reading circles, book clubs, and discussion boards to foster a community of engaged readers.

REFERENCES

[1] E. Hasan, M. Rahman, C. Ding, J. X. Huang, and S. Raza, "Based recommender systems: a survey of approaches, challenges and future perspectives," *ACM Computing Surveys*, vol. 58, no. 1, pp. 1–41, 2025.
[2] B. V. Umakant, B. V. Srinivasulu *et al.*, "Ai-powered library services: Personalized recommendations, enhanced search, and virtual assistance for information accessibility," *International Journal of Data Science and IoT Management System*, vol. 4, no. 3, pp. 375–385, 2025.
[3] A. I. Alomran and I. Basha, "An ai-based classification and recommendation system for digital libraries," *Scalable Computing: Practice and Experience*, vol. 25, no. 4, pp. 3181–3199, 2024.
[4] C. Tegetmeier, A. Johannssen, and N. Chukhrova, "Artificial intelligence algorithms for collaborative book recommender systems," *Annals of Data Science*, vol. 11, no. 5, pp. 1705–1739, 2024.
[5] K. Sabateen and J. A. Itmazi, "Recommender system for delivering learning contents from a digital library to a learning management system," *Ahliya Journal of Business Technology and MEAN Economies*, vol. 2, no. 2, pp. 43–60, 2025.
[6] J. Huang, S. Wang, L. Ning, W. Fan, S. Wang, D. Yin, and Q. Li, "Towards next-generation recommender systems: A benchmark for personalized recommendation assistant with llms," in *Proceedings of the Nineteenth ACM International Conference on Web Search and Data Mining*, 2026, pp. 217–226.
[7] W. Zhang, B. Liu, and S.-B. Tsai, "Analysis and research on digital reading platform of multimedia library by big data computing in internet era," *Wireless Communications and Mobile Computing*, vol. 2022, no. 1, p. 5939138, 2022.
[8] J. Banerji, "From textbook to screen: Using online authentic materials to develop reading comprehension skills," in *Sustainable Global Societies Initiative*, vol. 1, no. 2. Vibrasphere Technologies, 2026.
[9] T. Barnett, "Read in browser: Reading platforms, frames, interfaces, and infrastructure," *Participations*, vol. 16, no. 1, pp. 306–19, 2019.
[10] Y. Lin and Z. Yu, "Extending technology acceptance model to higher-education students' use of digital academic reading tools on computers," *International Journal of Educational Technology in Higher Education*, vol. 20, no. 1, p. 34, 2023.
[11] H. Mark and D. BOLLEN, "Student perceptions of an online extensive reading platform," , vol. 44, pp. 145–152, 2019.
[12] G. Nuriddinova *et al.*, "Cultivating the engaged reader in books and libraries," *European Review of Contemporary Arts and Humanities*, vol. 2, no. 1, pp. 17–24, 2026.
[13] E. Cho and M. Han, "Ai powered book recommendation system," in *Proceedings of the 2019 ACM Southeast Conference*, 2019, pp. 230–232.
[14] A. Kolhatkar, "Ai powered book recommendation system using the uml diagrams," 2025.
[15] W. Chen, Y. Zhao, J. Huang, Z. Ye, C. M. Ju, T. Zhao, N. Shah, L. Chen, and Y. Zhang, "Memrec: Collaborative memory-augmented agentic recommender system," *arXiv preprint arXiv:2601.08816*, 2026.