

Booknook

Mengwen Li
CS department
Columbia University
New York, USA
ml4643@columbia.edu

Abstract — BookNook is a cloud-based web application and recommender system that addresses the challenges faced by readers in today's digital era. In a world inundated with an overwhelming abundance of books, the platform seeks to simplify the discovery process and enhance the reading experience by leveraging cloud computing and advanced recommendation algorithms.

I. INTRODUCTION

Readers encounter difficulties in finding personalized book recommendations amidst the vast sea of available titles. The sheer volume of options can be daunting, hindering users from discovering new and intriguing books. Additionally, the inconvenience of manually searching for nearby bookstores or libraries compounds the challenge. BookNook aims to alleviate these issues by providing a solution that not only offers personalized book suggestions but also facilitates easy access to physical or online book retailers.

The exponential growth in the number of available books has created a need for a personalized recommendation system to cater to individual preferences. BookNook recognizes this need and strives to enhance the reading experience by guiding users to books that align with their tastes. Furthermore, the platform endeavors to foster a connection between readers and both local and online bookstores, promoting the love of reading and supporting the literary community.

While existing solutions like Goodreads.com provide book recommendations and user reviews, they often suffer from cluttered interfaces and usability challenges. BookNook distinguishes itself by integrating some recommendation algorithms to deliver more accurate and tailored suggestions. The platform also emphasizes a social aspect, enabling users to connect with like-minded readers and share their literary interests. Additionally, the incorporation of geolocation services sets BookNook apart, offering users valuable information about the nearest physical bookstores.

In essence, BookNook stands as an innovative solution that not only simplifies book discovery but also enriches the reading journey by fostering a sense of community and connecting readers with the diverse world of literature.

For this paper, we are going to discuss the Architecture, Key design details, the implementation, results and conclusions. Each of the following sections provide details for the mentioned topics.

II. ARCHITECTURE

BookNook is designed to seamlessly operate on the AWS cloud infrastructure, leveraging various services to create a robust and scalable system for delivering personalized book recommendations and facilitating access to nearby bookstores. The architecture comprises frontend and backend components, with key services such as AWS Lambda, API Gateway, Amazon S3, DynamoDB, and AWS Location Services forming the backbone of the system.

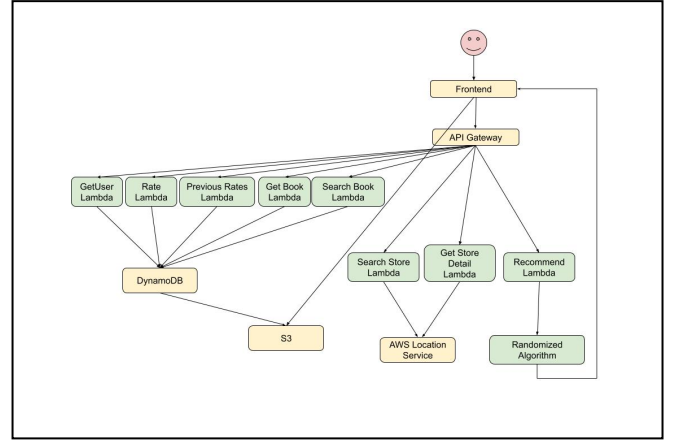


Fig. 1. The graph shows the service and archetecure for BookNook .

The user interacts with the BookNook frontend, initiating requests for home page, book recommendations, search queries, or location-based services. The frontend sends requests to the API Gateway, specifying the required actions or data. API Gateway directs the requests to the appropriate AWS Lambda functions, triggering the backend logic. Lambda functions interact with DynamoDB to retrieve or update user data, book information, and other relevant data. For geolocation services, Lambda functions communicates with saved user info and AWS Location Services to determine the user's location and provide information about nearby bookstores. The backend processes the requests, generates personalized book recommendations, and returns the results to the frontend through API Gateway. The frontend displays the recommendations, search results, or other relevant information to the user.

III. KEY DESIGN DETAILS

A. Frontend

Due to unforeseen circumstances, including the absence of the original partner and limited familiarity with frontend development on the part of the current team, BookNook currently lacks an elaborate and sophisticated user interface.

The UI design is simplified, focusing on functionality rather than intricate visual elements. This decision was made to prioritize the core features and functionalities supported by the backend, ensuring that the platform remains functional and user-friendly despite the absence of a visually intricate UI.

The complete frontend of BookNook is hosted on Amazon S3. This cloud-based storage service is utilized to store and serve static web content, including HTML, CSS, and JavaScript files that collectively form the user interface of the application. By leveraging S3, BookNook benefits from a scalable, secure, and reliable infrastructure for delivering the frontend content to users.

B. API gateways and Lambda functions

In the architecture of BookNook, the frontend communicates with the backend by making service calls through AWS API Gateway. This gateway serves as the centralized entry point for managing and routing requests from the client-side to the corresponding backend Lambda functions. By utilizing API Gateway, BookNook ensures a structured and secure means of interaction between the user interface and the backend services.

Within API Gateway, resources are defined and categorized to align with the primary entities in the BookNook ecosystem. The main categories include "books," "bookstores," and "users." This categorization reflects the core functionalities and data entities that the application manages. Each category corresponds to specific API endpoints, facilitating a logical and organized structure for handling various user requests.

API Gateway endpoints related to the defined resources trigger corresponding Lambda functions in the backend. These Lambda functions are designed to execute specific tasks based on the nature of the user request. The categorization into "books," "bookstores," and "users" allows for a modular and maintainable structure, where each Lambda function handles a specific set of actions.

The backend logic in BookNook is implemented using AWS Lambda functions, and these functions are written in Python. Python is chosen for its readability, flexibility, and ease of development. The Lambda functions serve as the executable units for processing user requests, enabling the seamless execution of backend logic in a serverless environment.

a) User Actions

Functions related to user profiles, such as retrieving book history and rating books, fall under this category. These functions are responsible for managing and providing information about user interactions with books.

b) Bookstore Features

Functions in this category utilize the location information stored in user profiles. They leverage AWS Location Service to search for nearby bookstores, enhancing the user experience by connecting users with physical book retail locations.

c) Book Details

Lambda functions associated with books provide detailed information about books. These functions handle requests for book details, enabling users to access comprehensive information about specific titles.

C. DynamoDB and S3

BookNook's primary data source is Kaggle datasets, which encompass information about available books, user profiles (with limited user details), and user ratings assigned to different books. These datasets are stored in a structured manner, in CSV formats, and are hosted on Amazon S3. Kaggle datasets provide a diverse and extensive collection of information that serves as the foundation for the BookNook application.

To make the datasets accessible and manipulable within the BookNook application, They are imported into Amazon DynamoDB. DynamoDB is chosen for its scalability, flexibility, and seamless integration with serverless architectures like AWS Lambda. The datasets, now residing in S3, are efficiently migrated into DynamoDB tables, ensuring that the data is structured and easily retrievable for various functionalities.

The image shows three screenshots of DynamoDB tables. The first table, 'Users', has columns: User-ID (String), Age, and Location. It contains 5 rows of data. The second table, 'Books', has columns: ISBN (String), Book-Author, Book-Title, Image-URL-L, Image-URL-M, Image-URL-S, Publisher, and Year-Of-Publication. It contains 3 rows of data. The third table, 'Rates', has columns: User-ID (String), ISBN (String), and Book-Rating. It contains 5 rows of data.

User-ID (String)	Age	Location
271880		new york, new york, usa
265330	18	newport, england, united kingdom
220632	34	stuttgart, baden-wuerttemberg, germany
138124	27	stuttgart, baden-wuerttemberg, germany
135462	32	boston, massachusetts, usa

ISBN (String)	Book-Author	Book-Title	Image-URL-L	Image-URL-M	Image-URL-S	Publisher	Year-Of-Publication
315116386	Alexandra Bl...	Texas Kiss L...	http://images...	http://images...	http://images...	Jove Books	1995
316100368	Marc Tolon Br...	Arthur Hake...	http://images...	http://images...	http://images...	Little Brow...	1998
689113617	Susan Pardy	AS EASY AS ...	http://images...	http://images...	http://images...	Scribner	1984

User-ID (String)	ISBN (String)	Book-Rating
265330	0676973183	8
230078	0375706856	0
230078	05381428X	0
230078	0767903385	0

Fig. 2. The graph shows Users table, Books table and Rates table from above to bottom.

a) Users Table

The Users table contains limited user information, possibly including user IDs, names, and basic profile details. This table serves as the repository for user profiles, allowing BookNook to identify and differentiate individual users.

b) Books Table

The Books table comprises book-related information, such as ISBN, title, author, and other relevant details. It acts as the central repository for book-related data, enabling BookNook to provide comprehensive information about available books.

c) Rates Table

The Rates table records the ratings given by users to various books. This table includes user IDs, book ISBNs, and corresponding user ratings. It plays a crucial role in building the recommendation system by capturing user preferences and interactions with specific books.

IV. IMPELEMENTATION

A. Frontend

BookNook's frontend is developed solely in JavaScript, without the use of advanced frontend technologies such as Node.js and React. While Node.js and React are commonly employed to enhance the development of dynamic and interactive user interfaces, the BookNook team has chosen a more straightforward approach by relying solely on JavaScript. This decision aligns with the team's familiarity with JavaScript and allows them to capture the essential features supported by the backend without introducing additional complexity.

B. Backend

Lambda functions in BookNook are programmed to respond to specific events, such as user requests for personalized book recommendations, search queries, or geolocation services. Each Lambda function encapsulates a specific piece of functionality, allowing for modular and maintainable backend code.

In cases where geolocation services are required, specific Lambda functions handle the communication with AWS Location Services. These functions determine the user's location and retrieve information about nearby bookstores to enrich the user experience.

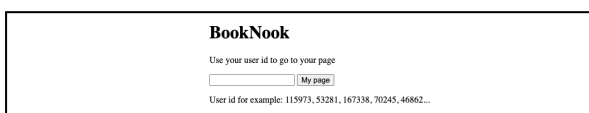
Lambda functions contribute to the scalability and resource efficiency of BookNook by automatically scaling up or down based on demand. This ensures that the application remains responsive even during peak usage periods.

C. Recommendation

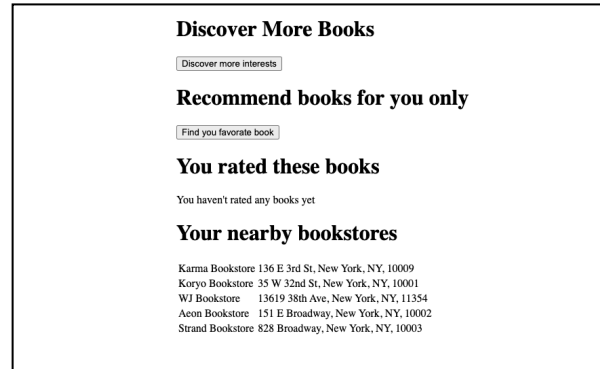
As my partner is not present and has not fulfilled her commitment to complete the recommendation engine, the current implementation replaces the sophisticated recommendation algorithm with a randomized selection from our existing book collections in the system.

V. RESULTS

The implementation underwent slight changes from the initial design. During the integration of the backend with the frontend, it became apparent that certain existing endpoints were not as helpful as anticipated. Simultaneously, the need for additional endpoints arose to enhance the overall user experience. As a result, adjustments were made to the original design to optimize the integration and address specific requirements for a more user-friendly interface.



The image shows a login page for BookNook. It has a title "BookNook" and a subtitle "Use your user id to go to your page". Below this is a text input field for the user ID and a button labeled "My page". At the bottom, there is a note: "User id for example: 115973, 53281, 167338, 70245, 46862..."

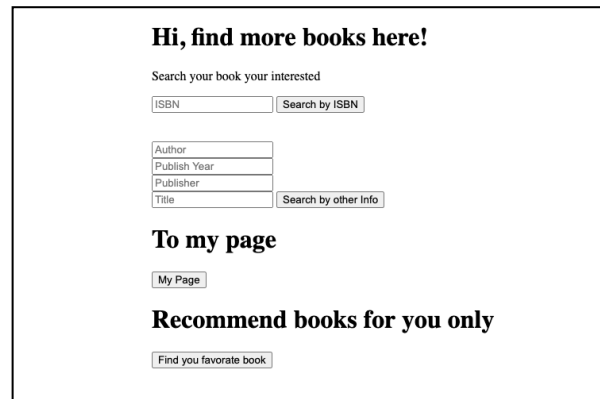


The image shows a user home page. It has a title "Discover More Books" and a button "Discover more interests". Below this is a section "Recommend books for you only" with a button "Find you favorite book". Then there is a section "You rated these books" with the text "You haven't rated any books yet". Finally, there is a section "Your nearby bookstores" with a list of bookstores: Karma Bookstore (136 E 3rd St, New York, NY, 10009), Koryo Bookstore (35 W 32nd St, New York, NY, 10001), WJ Bookstore (13619 38th Ave, New York, NY, 11354), Aeon Bookstore (151 E Broadway, New York, NY, 10002), and Strand Bookstore (828 Broadway, New York, NY, 10003).

Fig. 3. The main page and example user home page.

Upon visiting the BookNook main webpage, users are prompted to enter their user ID. Following this, they are directed to a personalized page, which not only displays all the books the user has rated before but also provides information about nearby available bookstores. This personalized page serves as a central hub for the user, offering a comprehensive view of their reading history and convenient access to local bookstore information.

Additionally, the main page links to two essential sections: the search page and the recommendation page.



The image shows a search page. It has a title "Hi, find more books here!". Below this is a section "Search your book your interested" with an input field for ISBN and a button "Search by ISBN". Then there is a section "To my page" with a button "My Page". Finally, there is a section "Recommend books for you only" with a button "Find you favorite book".

Fig. 4. The search page.

The search page, accessible from the user's main page, facilitates flexible exploration. Users can search using a singular criterion, such as ISBN, or employ a combination of parameters including title, author, publish year, and publisher. This versatility empowers users with a comprehensive and customizable search experience tailored to their preferences.

On the recommendation page, users are presented with direct system-generated recommendations. The system showcases a curated list of books and invites users to provide their ratings. This interactive element not only enhances user engagement but also contributes to refining the recommendation algorithm, ensuring that future suggestions align more closely with individual preferences. The recommendation page thus serves as a dynamic

platform for discovering new and tailored reading experiences.

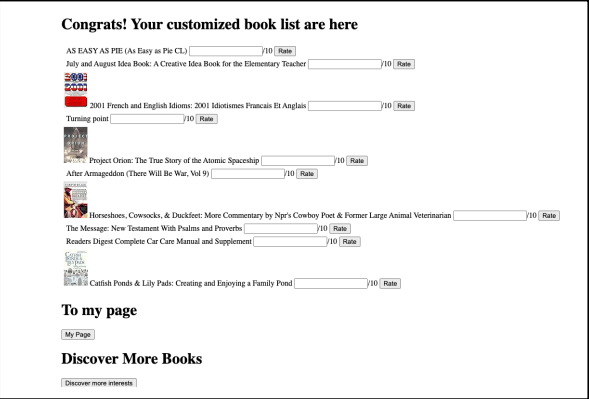


Fig. 5. The Recommendation page.

VI. CONCLUSION

BookNook stands as a dynamic and user-centric platform that aims to revolutionize the reading experience. By integrating a personalized semi-recommendation system, flexible search functionalities, and extensive information about nearby bookstores, BookNook offers readers a comprehensive and engaging environment.

The emphasis on user personalization is evident in features such as the user-specific main page, showcasing previously rated books and nearby bookstore details. Though the recommendation system doesn’t leverage advanced algorithms yet to deliver tailored book suggestions, it leaves the space to do such improvement. Additionally, the search page provides users with the flexibility to explore books based on various criteria.

The integration of AWS services, including Lambda functions, API Gateway, S3, and DynamoDB, ensures a scalable, reliable, and efficient infrastructure. The cloud-based architecture enhances the platform's responsiveness and adaptability to user demands.

Despite encountering challenges, such as the absence of a team member and subsequent adjustments to the initial design, BookNook perseveres in delivering a valuable reading companion. The commitment to providing a user-friendly interface, fostering community engagement through shared interests, and supporting local bookstores contributes to BookNook's vision of enhancing the literary journey for readers around the globe.

As BookNook evolves, continual refinement of its recommendation algorithms and the expansion of features will contribute to a richer and more satisfying reading experience. In essence, BookNook strives to be more than a platform—it aspires to be a community that connects readers, fosters a love for literature, and simplifies the discovery of the next captivating page-turner.

ACKNOWLEDGMENTS

Due to my partner's decision to withdraw from the class after Thanksgiving, Recommendation engine features in this project is missing.

The following chart describes the effort for this project

Contributor	Efforts	Contribute%
bt2509	Initial proposal First draft Figma	2.5%
ml4643	First draft Swagger, Entire architecture design, Entire second checkpoint, Entire frontend(original assign to bt2509), Entire backend, All submission material such paper, demo, YouTube video and github	97.5%

REFERENCES

[1] Amazon Lambda Service. (n.d.). <https://aws.amazon.com/pm/lambda/>
[2] Amazon Dynamo Service. (n.d.). <https://aws.amazon.com/dynamo/>
[3] Amazon S3 Service. (n.d.). <https://aws.amazon.com/s3/>
[4] Amazon Location Service. (n.d.). <https://aws.amazon.com/location/>
[5] Amazon API Gateway. (n.d.). <https://aws.amazon.com/api-gateway/>