University of Colorado Boulder

BOOKHIVE:

Your Gateway for Book Reviews and Recommendations

Zach Dyre | Abdirahman Ebiso | Zariyat Hossain | Ryan O'Leary | Isaias Perez

CSCI 3308: Software Development Methods and Tools

Prof. Sreesha Nath

9 Dec. 2024

Contributors

Zach Dyre (zdyre), Abdirahman Ebiso(abdi0101), Zariyat Hossain (zariyat15), Ryan O'Leary (Ryol7118) and Isaias Perez (IP-04) contributed to BookHive.

Description of BookHive

BookHive is a web application that is designed to enhance the book discovery and review process through creating a social media like community where readers can share insights and find recommendations tailored to their preferences. Our platform enables users to explore a vast collection of books, rate them, and leave detailed reviews that are visible to others within the community. Leveraging user feedback and preferences, BookHive aims to deliver personalized suggestions helping our users discover new books and genres that align with them.

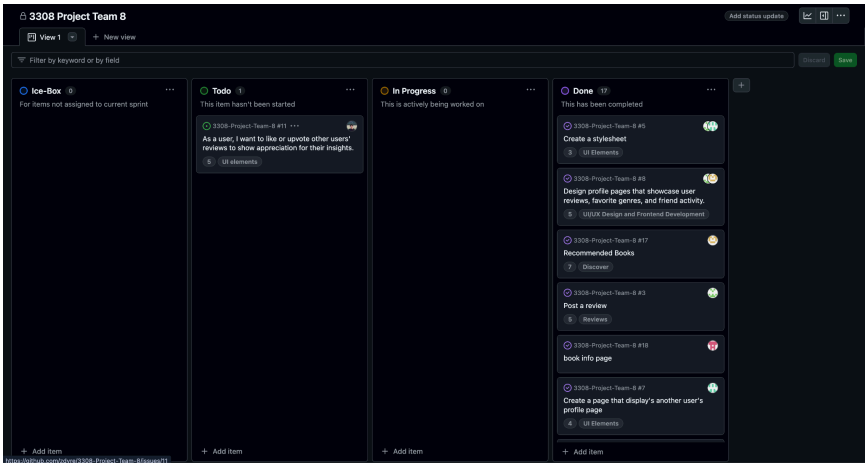
Our system is built with a robust set of technologies to ensure seamless user experience and functionality. The backend utilizes Node.js and Express for server-side operations, supported by PostgreSQL for efficient database management. The frontend employs HTML, Handlebars, and Bootstrap to provide an intuitive interface. BookHive also integrates external API's like google books for fetching book data and Render for cloud hosting.

Some key features include user-generated reviews, a recommendation algorithm, a discover page, a streamlined search bar, and a user-friendly interface for book browsing. The development process allowed the team to quickly iterate and refine features, overcoming challenges such as database setup, Git command management, and learning to collaborate effectively.

BookHive aspires to evolve with upcoming advanced features, such as an enhanced algorithm and community driven insights, to further enrich the reading experience.

Project Tracker - GitHub Project Board

GitHub Project Board Link: https://github.com/zdyre/3308-Project-Team-8



Video

https://drive.google.com/file/d/1p50G0Petip_emCf4ziceLJ2motJQkQ16/view?usp=sharing

VCS

GitHub Repository: https://github.com/zdyre/3308-Project-Team-8

Contributions

Zach Dyre:

In this project, I helped develop much of the home page and implemented a book details page as well as a user profile page. Additionally, I handled creating the API request to Google Books and populating our database with its information. For the home page, featured and trending books are determined based on friend preferences and book rating respectively. The

profile page has features implemented to allow one to edit their own description, view other profiles, add/remove friends, and view books another user has liked. Finally, the book details page displays all relevant information including title, author, description, and user reviews as well as a link to purchase the book. Both the profile page and book details page have DOM implementation to update without refreshing.

Abdirahman Ebiso:

I developed the Discover Page, allowing users to explore other profiles, view new book releases, and receive personalized book recommendations. On the backend, I used Node.js, Express, and PostgreSQL. I created a PostgreSQL function to handle partial dates (e.g., year or year-month) and ensure consistent date formatting. Additionally, I designed the wishlist table to track books users wish to read in the future. On the frontend, I used Handlebars to dynamically render data, integrating the UI Avatars API to generate profile images based on usernames. The page allows users to click on profiles to view recommendations (randomly selected from the profiles database) and save books to their Future Reads list.

Zariyat Hossain:

I played a role in developing and establishing the framework for critical components of BookHive. I built the initial structure of the profile page and discover page, laying out their core functionalities, including rendering user-specific data like reviews, favorite genres, and personalized book recommendations using Handlebars. These pages showcase the user-centric features that form the heart of BookHive's functionality. I also configured the .env file for secure

session handling and database integration. Additionally, I implemented dynamic navigation bar logic, seamlessly adapting buttons based on user authentication status to improve usability and provide a personalized user experience. I also developed and incorporated middleware to enable secure, reliable, and efficient session-based authentication, ensuring that users could interact with the platform safely and smoothly.

Ryan O'Leary:

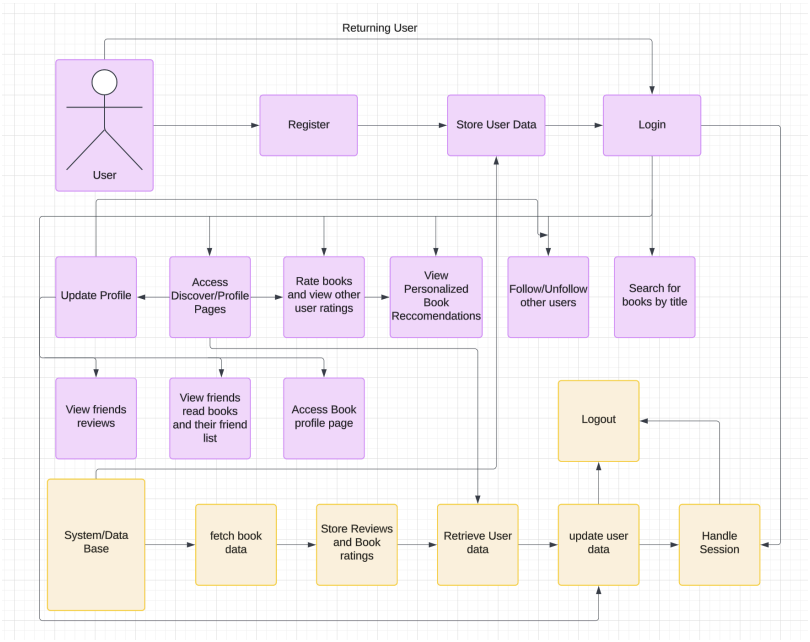
I played a large role in front-end development as well as creation of javascript code that provided functions for the proper functioning of the website. I was responsible for the login and registration pages as well as contributing for the logic in adding new users to our PostgreSQL database. I played a key role in the initialization of our back-end systems: the database, the docker and the creation of API routes for navigation of the website. Towards the end of the project, I helped with feature polishing and bug fixing. And finally I was the creator of our final presentation.

Isaias Perez:

My contributions to the BookHive project were diverse and important to its success. I was fully responsible for developing the search bar feature and the search results page, ensuring that users could easily find books in our database. I also contributed significantly to the rating and review system, helping to implement functionality that allows users to rate books and view top-rated reviews. Aside from my own tasks, I assisted teammates in resolving issues with Docker setup, login and registration bugs, and improving the CSS design to make the website

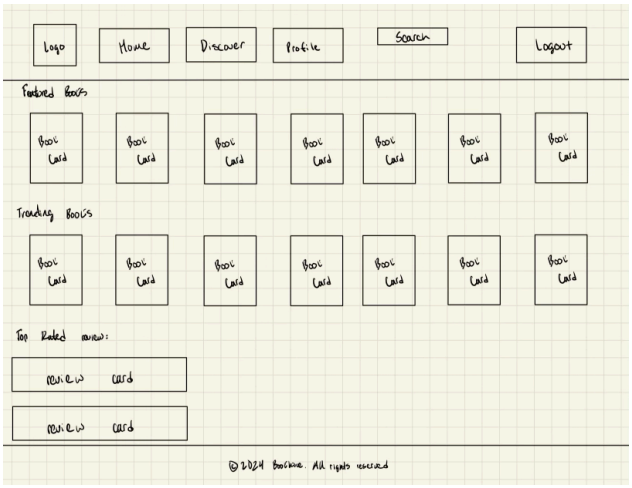
more visually appealing. In addition, I helped manage the Scrum system, tracked meetings, and helped keep the team organized and on track. Overall, my work in both development and team support was significant to the project's overall success.

Use Case Diagram

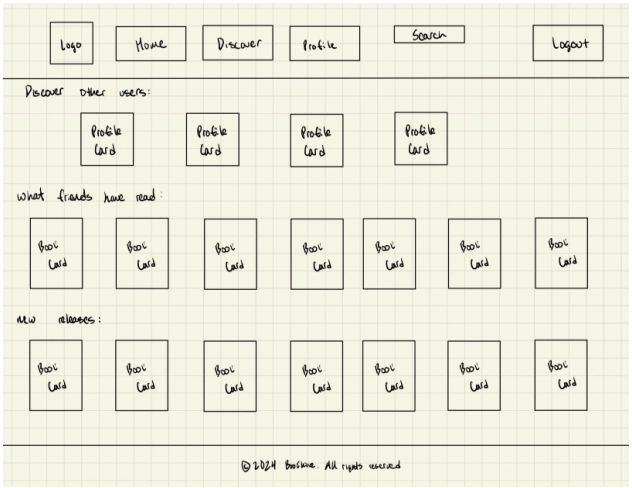


Wireframes

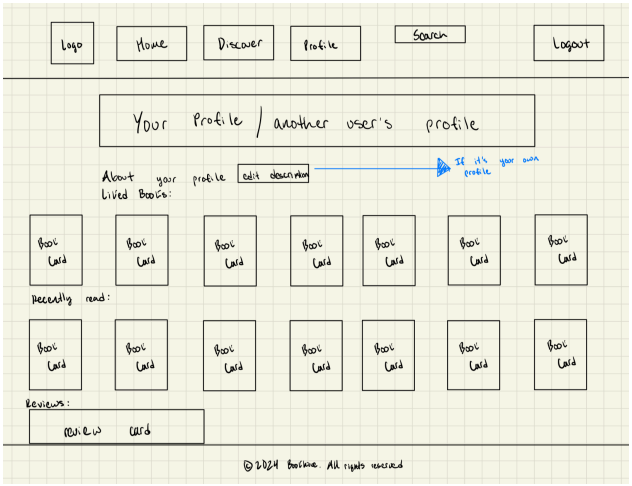
Home Page:



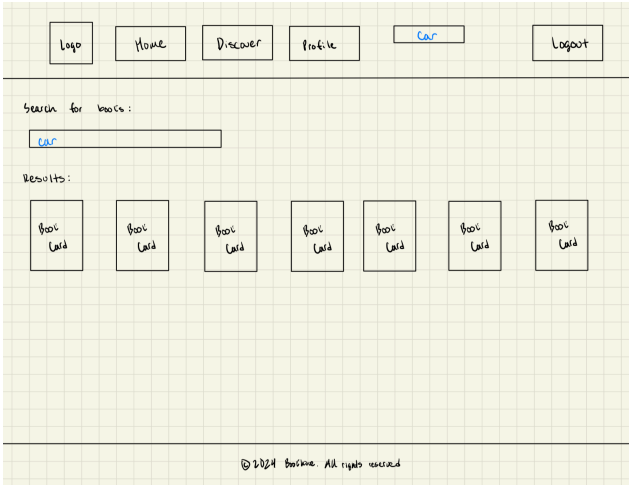
Discover Page:



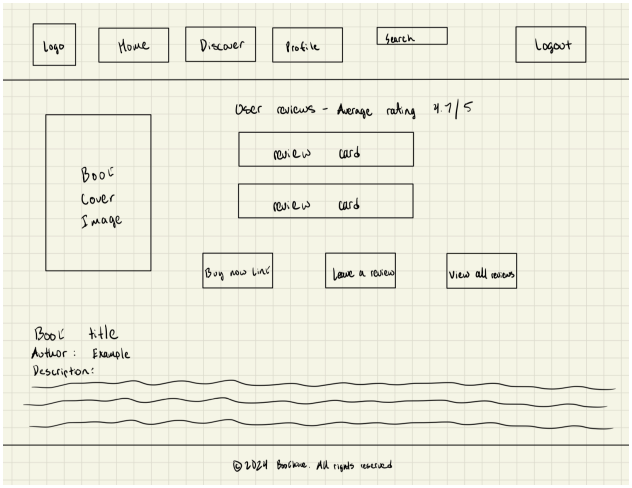
Profile Page:



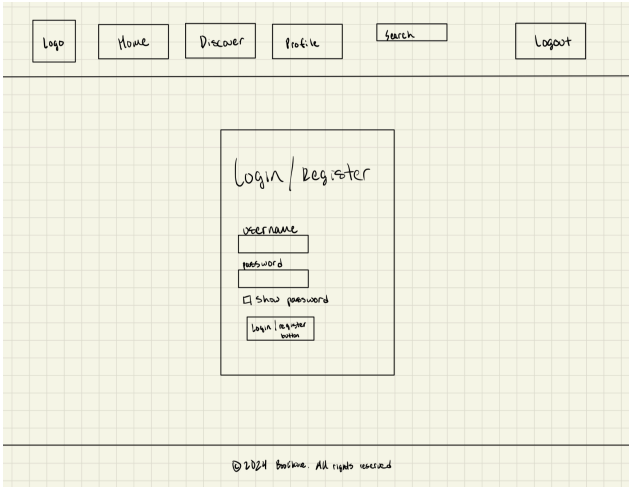
Search Results Page:



Book Details Page:



Login/Register Page:



Test Results:

During our user testing, we were able to identify many bugs and navigation issues that were patched prior to our final project. Most of our bugs had to do with the profile and book details page improperly updating the DOM or database. For example, when submitting a review, it would only update the DOM and not properly adjust the database. Our user testing also revealed issues with our input field validation, allowing users to leave overly lengthy

descriptions or rating higher than 5/5. Additionally, user testing revealed that data wasn't being properly rendered when accessing certain pages through anchor tags.

Beyond all of the bugs we found and fixed, we also optimized the UI and navigation flow between pages. Prior to user testing, book reviews would not link back to the book they were for. However, this was changed because it made more sense for user navigation. Additionally, the profile page was updated to clarify when a user has no friends by displaying "No Friends Yet!" instead of being an empty list. Finally, the book details page was updated to show a popup error if a user has already left a review and attempts to leave another one. These changes likely would not have been made without user testing because we as the creators of the website already had full knowledge of how to navigate the site.

Deployment

Our web application, Bookhive, is live and accessible at the link: https://three308-project-team-8.onrender.com/. Our app was deployed by using Render, a cloud platform, for hosting web services and databases. Our backend was configured for deployment with our own environmental variables (such as "SESSION_SECRET") and was built with Node.js and Express. Our database credentials are also securely stored in Render's environment configuration. Using PostgreSQL, we provisioned a database using Render's managed database service, with proper scheme setup during deployment. Our frontend was created using HTML, Handlebars, and Bootstrap. These tools were deployed alongside the backend along with static files through Express's middleware.

Render provided us with a public URL for anyone with an internet connection to access. Our project is integrated using GitHub for continuous and seamless deployment, ensuring

updates are automatically reflected when changes are pushed to our main branch. Before we officially deployed we conducted testing on our local devices using Docker to ensure the app's functionality. To access BookHive, users can visit the link, register an account, or log in to explore our features such as book discovery, reviews, and personalized recommendations.