

WADL MINI PROJECT REPORT

ON

**“BookHive – A social platform for Bibliophiles”**

SUBMITTED BY

Rudra Kadam – 33330

Mugdha Kulkarni – 33338

Vaishnavi Mahale – 33342

Under the guidance of  
Mrs. Deepali Salapurkar



DEPARTMENT OF INFORMATION TECHNOLOGY  
**PUNE INSTITUTE OF COMPUTER TECHNOLOGY, PUNE**

Sr. No. 27, Trimurti Chowk, Dhankawadi, Pune-43

AY 2023-24

## **ABSTRACT**

BookHive is a React application developed as a mini project for the Web Application Development subject. The project aims to provide a user-friendly platform for book enthusiasts to keep track of their reading progress and manage their book lists efficiently.

The application is built using React, a popular JavaScript library for building user interfaces. React's component-based architecture and virtual DOM implementation ensure a smooth and responsive user experience. The frontend is hosted on Azure App Services, a fully managed platform as a service (PaaS) offering from Microsoft Azure.

BookHive allows users to create personalized accounts and maintain three distinct book lists: "Read," "Currently Reading," and "Want to Read." Users can easily add books to their respective lists, providing details such as title, author, cover image, and brief descriptions. The application leverages a backend API to fetch and store book data, ensuring data persistence and scalability.

One of the key features of BookHive is its seamless integration with cloud services. The backend API is hosted on a separate cloud platform, decoupling the frontend and backend components for better scalability and maintainability. Additionally, the application leverages cloud storage services to store and retrieve book cover images, ensuring efficient content delivery.

The cloud-native architecture of BookHive enables high availability, scalability, and ease of deployment. The project showcases the implementation of a modern SaaS application leveraging cloud technologies, demonstrating the benefits of cloud computing in delivering robust and scalable solutions.

Overall, BookHive demonstrates the application of cloud computing concepts, React development, and software engineering principles in building a functional and user-friendly SaaS application for book lovers.

## INTRODUCTION

In today's digital age, reading habits have evolved, and book enthusiasts seek convenient and efficient ways to manage their literary pursuits. BookHive, a React Web application, is designed to cater to the needs of avid readers by providing a centralized platform for tracking their reading progress and managing their book collections.

The primary objective of BookHive is to simplify the process of organizing and maintaining personal book lists. With its user-friendly interface and robust features, the application aims to enhance the reading experience by offering a seamless solution for keeping track of books that users have read, are currently reading, or plan to read in the future.

BookHive leverages the power of cloud computing technologies to deliver a scalable and highly available solution. By hosting the frontend on Azure App Services and integrating with a separate backend API hosted on a different cloud platform, the application demonstrates a decoupled and modular architecture, enabling independent scaling and maintenance of individual components.

One of the key advantages of BookHive is its ability to provide a personalized experience for each user. Upon creating an account, users can maintain three distinct book lists: "Read," "Currently Reading," and "Want to Read." These lists can be easily updated by adding or removing books, allowing users to have a comprehensive overview of their reading progress and future reading plans.

The application streamlines the process of managing book information by enabling users to input details such as title, author, cover image, and brief descriptions. BookHive also incorporates a rating system, allowing users to rate and review books they have read, fostering a community-driven platform for book recommendations and discussions.

Moreover, BookHive leverages cloud storage services to store and retrieve book cover images, ensuring efficient content delivery and enhancing the overall user experience. The cloud-native architecture of the application ensures high availability, scalability, and ease of deployment, showcasing the benefits of cloud computing in delivering robust and scalable solutions.

## MOTIVATION

The idea for developing BookHive emerged from a deep understanding of the challenges faced by avid readers in efficiently managing their book collections and reading lists. Despite the abundance of digital tools and platforms, there remained a need for a user-friendly and comprehensive solution tailored specifically for book enthusiasts.

One of the primary motivations behind BookHive was to provide a centralized platform where readers could easily keep track of their reading progress. Many book lovers struggle with maintaining a coherent record of the books they have read, are currently reading, or plan to read in the future. This lack of organization often leads to confusion, duplicated purchases, or missed opportunities to explore new literary works.

Additionally, the rise of cloud computing technologies and the increasing demand for scalable and highly available applications prompted the development of BookHive as a cloud-native SaaS solution. By leveraging the power of cloud services, BookHive aims to deliver a seamless and reliable experience to users, regardless of their location or device.

Furthermore, the project was driven by the desire to foster a community-driven platform for book discussions and recommendations. By incorporating features such as rating systems and user reviews, BookHive aims to create a vibrant ecosystem where readers can share their thoughts, discover new authors, and engage with like-minded individuals.

Another significant motivating factor was the need for a personalized and user-friendly interface. BookHive was designed with the goal of providing a seamless and intuitive experience, allowing users to effortlessly manage their book lists, add or remove entries, and access relevant information with minimal effort.

Moreover, the project aimed to demonstrate the application of modern software development practices, such as decoupled architectures, cloud integration, and scalable design principles. By implementing these concepts, BookHive serves as a practical example of how cloud computing technologies can be leveraged to build robust and scalable applications.

Overall, the motivation behind BookHive stems from a deep understanding of the challenges faced by book enthusiasts, the recognition of the potential of cloud computing technologies, and the desire to create a user-friendly and community-driven platform that enhances the reading experience.

## IMPLEMENTAION DETAILS

### **Problem Statement clearly stating Input and Output:**

Book enthusiasts often face challenges in organizing and keeping track of their reading progress, book collections, and future reading plans. The lack of a centralized and user-friendly platform can lead to confusion, duplicated purchases, and missed opportunities to explore new literary works. Additionally, the absence of a community-driven platform limits the ability to share book recommendations, reviews, and engage with like-minded individuals. The primary objective of the BookHive project is to develop a React web-application that addresses these challenges by providing a comprehensive and user-friendly solution for managing personal book lists and fostering a vibrant community for book discussions and recommendations.

### **Input:**

The input to the BookHive application consists of the following:

1. **User Registration and Authentication:** Users must provide their personal information, such as name, email address, and a secure password, to create an account and authenticate themselves.
2. **Book Information:** Users can input details about books, including title, author, cover image, brief description, and optionally, a personal rating or review.
3. **List Management:** Users can add books to three distinct lists: "Read," "Currently Reading," and "Want to Read," allowing them to organize their reading progress and future plans.
4. **User Interactions:** Users can engage with the community by leaving reviews, ratings, and comments on books, as well as interacting with other users' reviews and recommendations.

### **Output:**

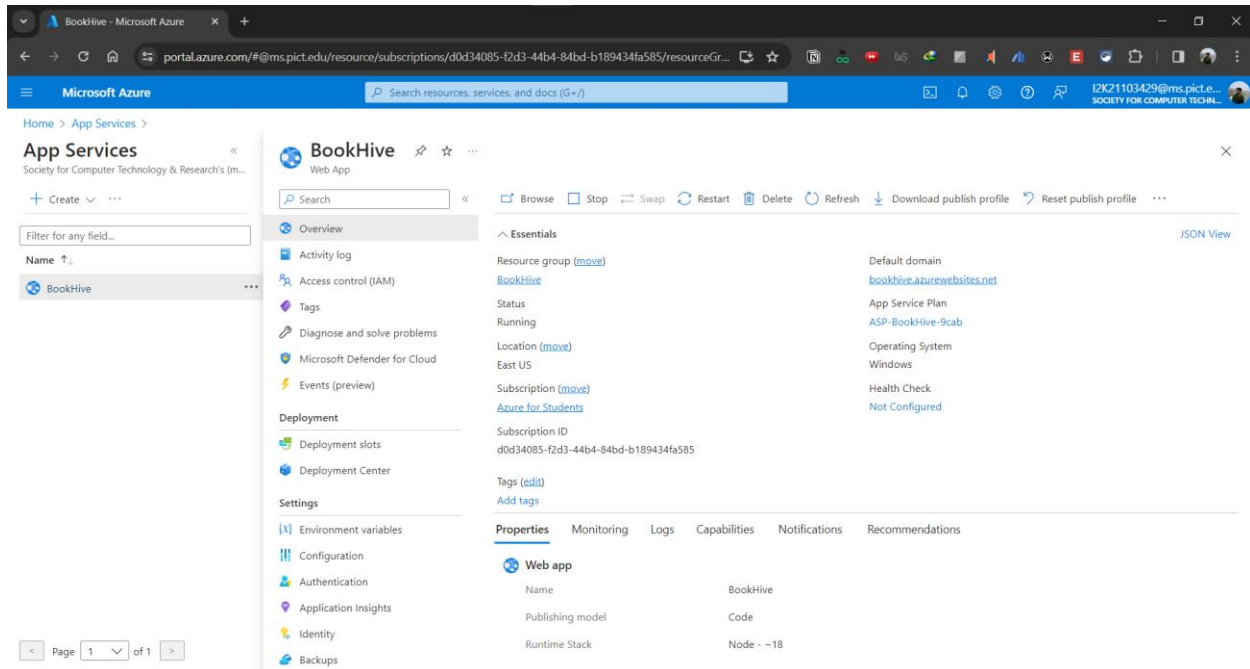
The output of the BookHive application includes the following:

1. **Personalized Book Lists:** Users can view and manage their personal "Read," "Currently Reading," and "Want to Read" book lists.
2. **Book Details:** Users can access detailed information about each book, including title, author, cover image, description, average rating, and reviews from the community.
3. **Community Recommendations:** Users can explore book recommendations based on reviews, ratings, and popularity within the BookHive community.
4. **User Profiles:** Users can view their own profiles, displaying their personal book lists, reviews, and ratings, as well as interact with other users' profiles and shared content.

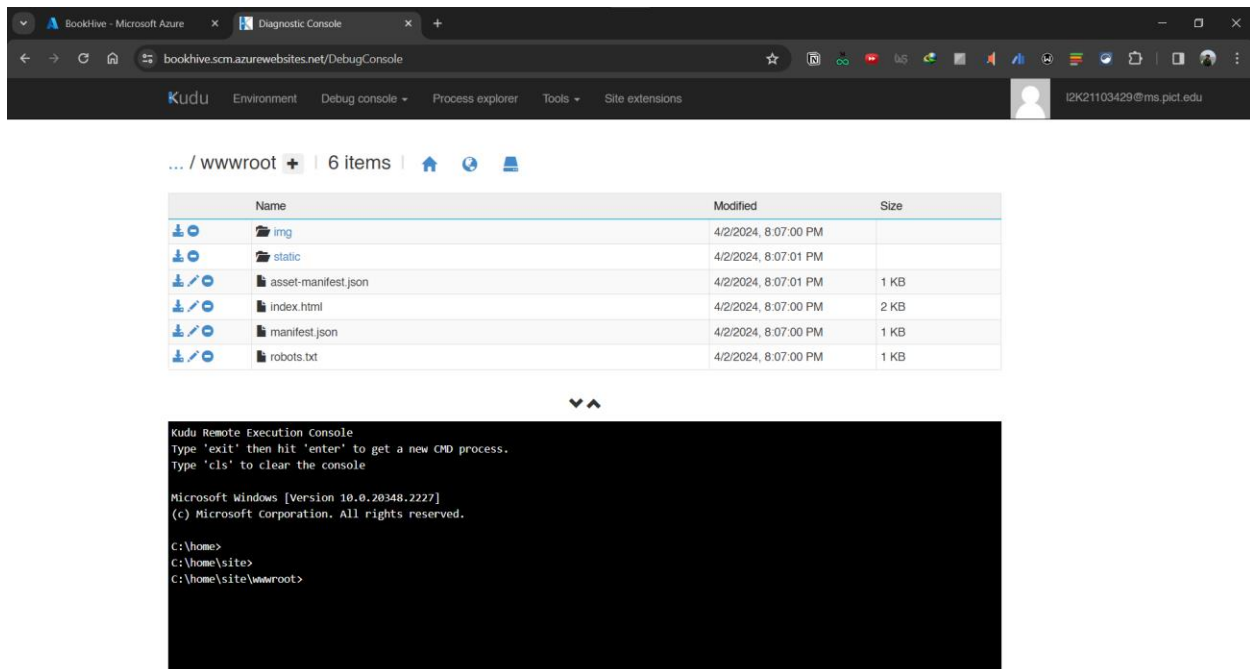
**Workload distribution among members:**

Sr. No	Member name	Tasks carried out	List of resources used / technology used
1	Rudra Kadam	Back-end and Deployment	React, Azure
2	Mugdha Kulkarni	Front-end	Figma, React
3	Vaishnavi Mahale	Front-end	Figma, React

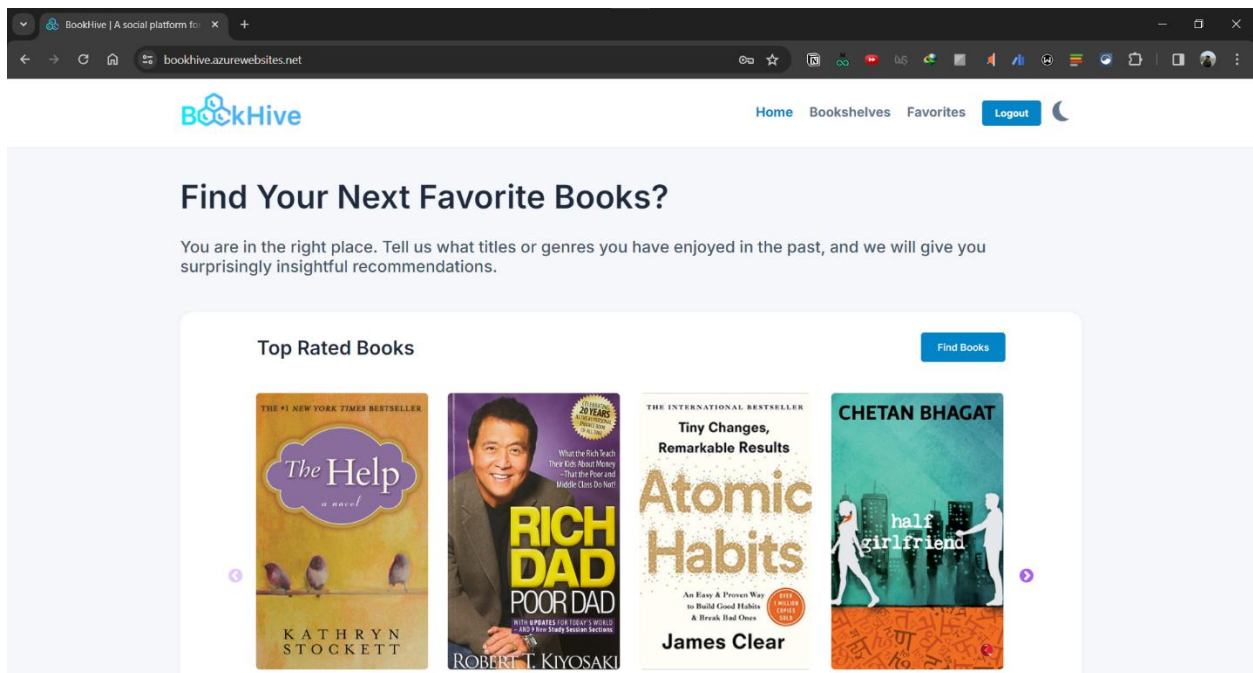
## OUTPUT (Screenshots)



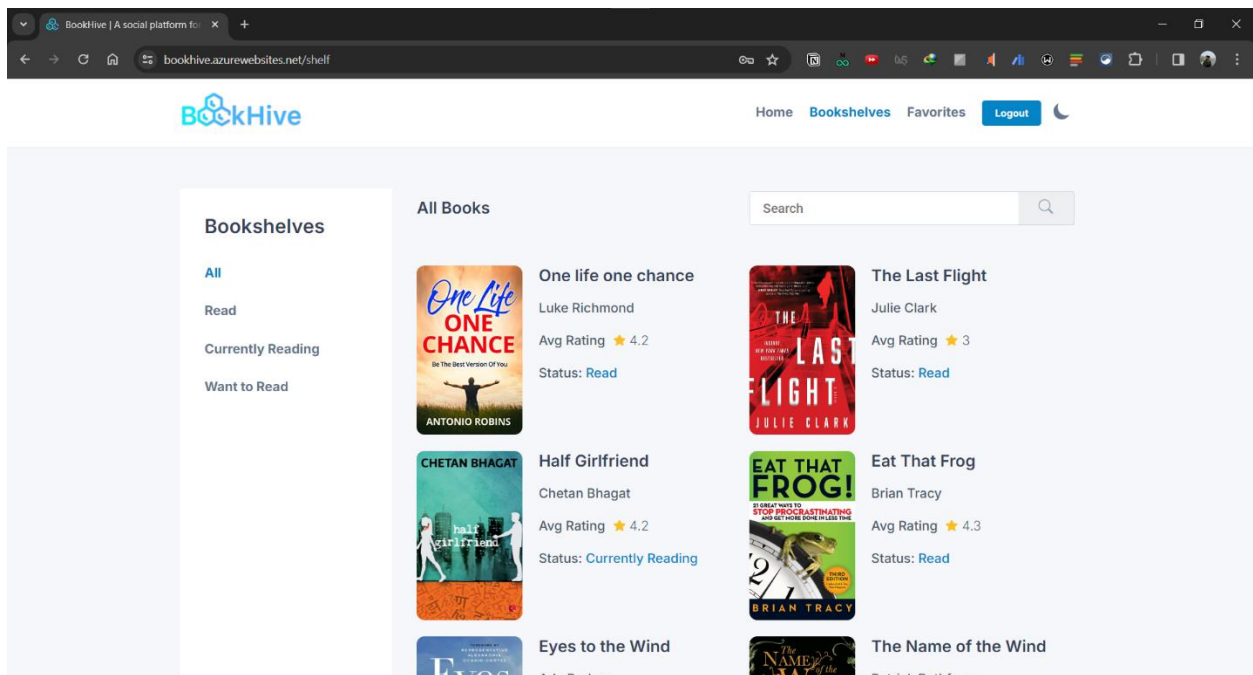
### 2. Azure App Services Overview



### 1. Azure Kudu Website Code Files

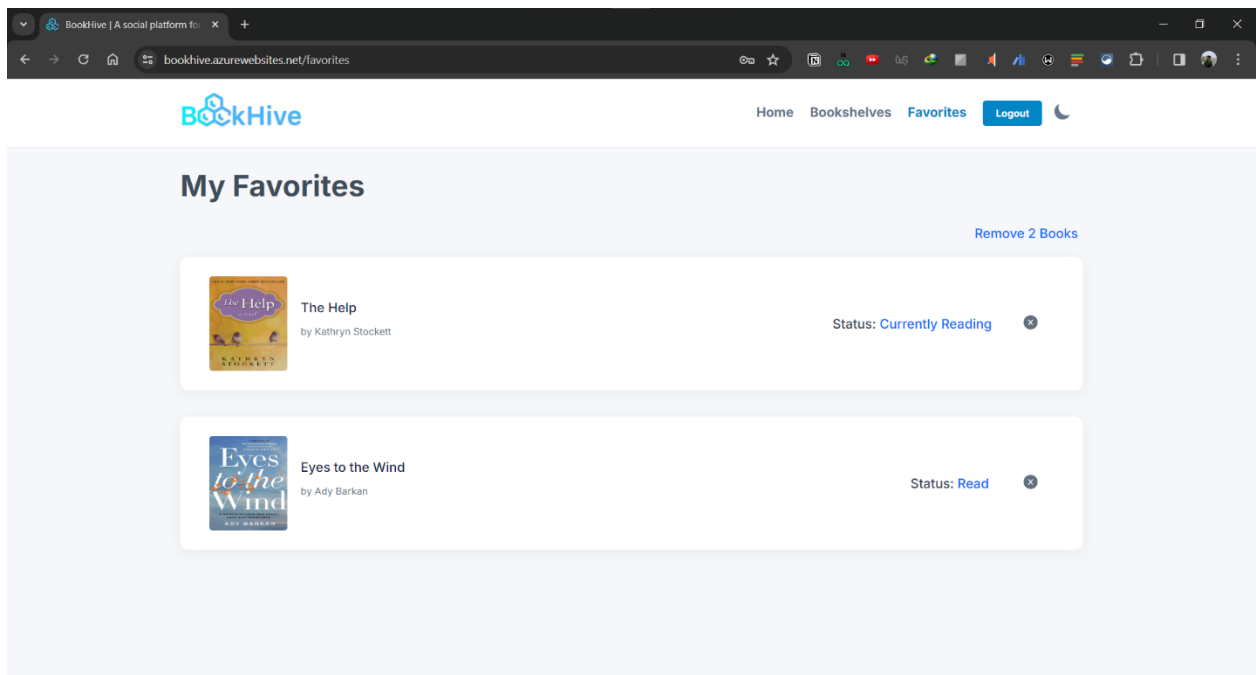


3. Home Page

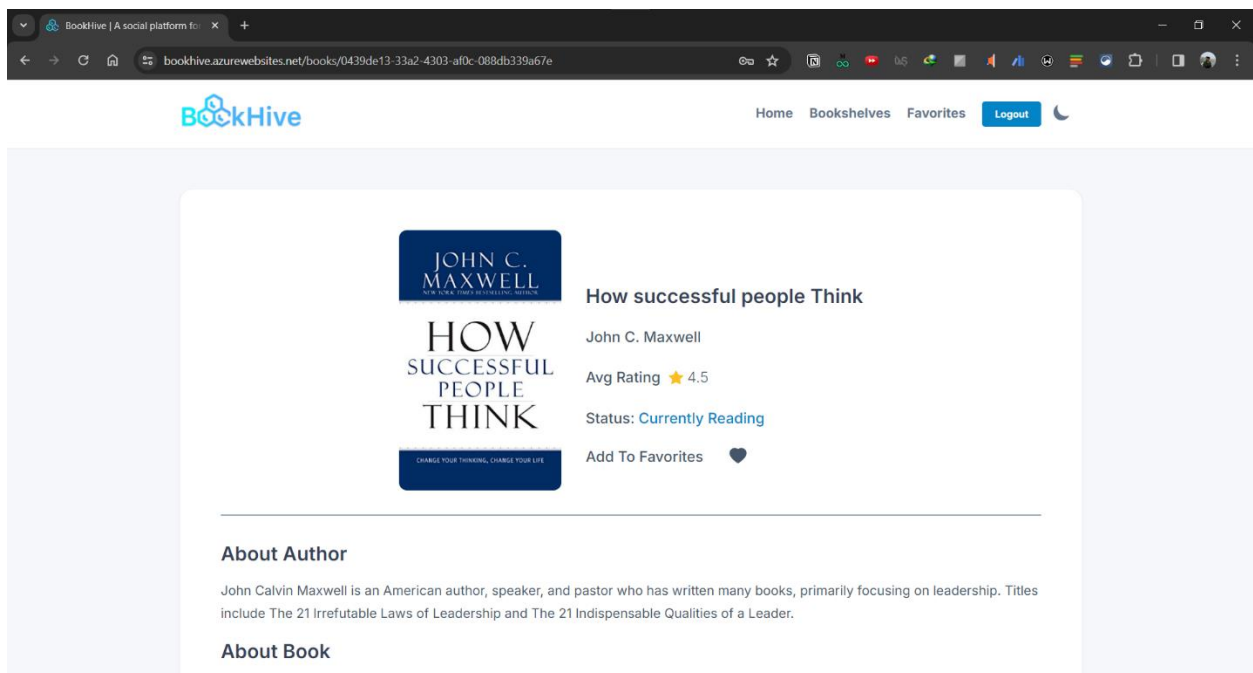


4. Bookshelves

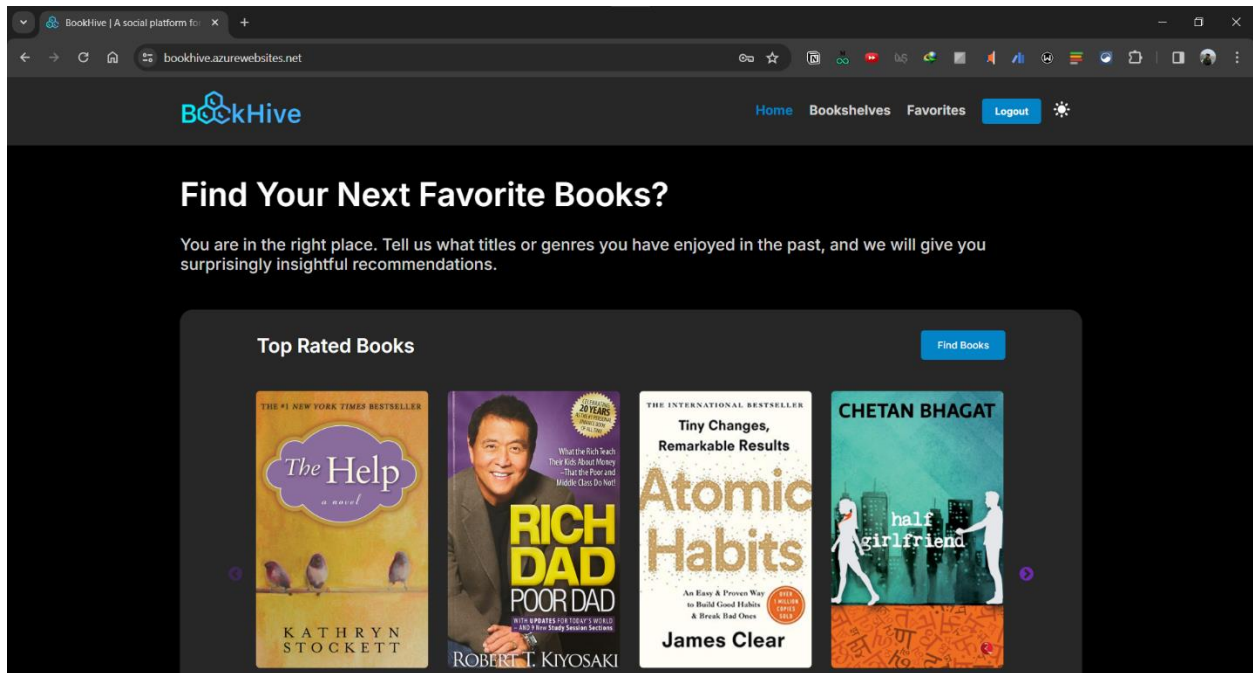




5. Favorites Page



6. Book Details



7. Home Page (in Dark Mode)

## CONCLUSION

The development and implementation of BookHive, a React web-application, has successfully addressed the challenges faced by book enthusiasts in managing their reading progress, book collections, and fostering a community-driven platform for book discussions and recommendations.

Through its user-friendly interface and robust features, BookHive has provided a centralized platform where users can effortlessly maintain three distinct book lists: "Read," "Currently Reading," and "Want to Read." This organization system has enabled users to have a comprehensive overview of their reading journey, eliminating confusion and duplicated purchases.

BookHive has demonstrated the benefits of a cloud-native architecture, ensuring high availability, scalability, and efficient content delivery. The decoupled architecture, with the frontend hosted on Azure App Services and the backend API hosted on a separate cloud platform, has facilitated independent scaling and maintenance of individual components, enhancing the application's overall performance and reliability.

Furthermore, BookHive has fostered a vibrant community by incorporating features such as rating systems, user reviews, and book recommendations. Users can now engage with like-minded individuals, share their thoughts on literary works, and discover new authors, enriching their reading experiences.

Throughout the development process, BookHive has exemplified the application of modern software development practices, such as decoupled architectures, cloud integration, and scalable design principles. The project serves as a practical example of how cloud computing technologies can be leveraged to build robust and scalable applications, paving the way for future innovations in the SaaS domain.

Looking ahead, BookHive has the potential for further enhancements and feature additions, such as integration with online book retailers, personalized recommendations based on user preferences, and the creation of virtual book clubs or discussion forums. These advancements would further solidify BookHive's position as a comprehensive and innovative solution for book lovers worldwide.

In conclusion, the BookHive project has successfully achieved its objectives by providing a user-friendly and community-driven platform for managing personal book lists, fostering literary discussions, and leveraging the power of cloud computing technologies to deliver a scalable and highly available solution.