**Introduction:**

Our project is titled *The BookCooper - Virtual Book Exchange Platform*. Our team consists of Alex Liu (EE’25), Aidan Cusa (EE’26), and Isaac Schertz (EE’26).

The goal of the project is to create a simple marketplace for readers, while not having to purchase a new book everytime and get no value for their old purchases. Here, users have the ability to earn points for sending in undesired books and then use points to purchase new ones. These points, called “Book Bucks” (B-Bucks for short), can also be purchased with real money. This Facebook-like online marketplace exchange is tailored and made for all types of books. *The BookCooper* will start small, being an exclusive marketplace for Cooper affiliated individuals, though will eventually be integrated in every book lover’s daily life!

**Project Scope:**

Frontend:



Block diagram for the frontend

Written in React, the web application can be traversed through a navigation bar which redirects users to the home page, search page, create a listing page, point system for book exchanges, and account page. In addition there will be a cart icon on the corner for user to checkout. The users will first create an account using an username, an email, and a password. Once logged in, users will be directed to the home page. There, users have the option to search for a specific book using the search bar. A filter feature will aid users to navigate through our platform efficiently to find a book of their choice.

Selling your book is another option on the home page. Users will be asked several questions about their book, ranging from name to quality. Then our proprietary algorithm will evaluate the book in points.

Frontend Development (React):

* Landing Page: First time site visitors will be brought to this page.
* Account Creation and Management: Users will create accounts and manage their profiles.
* Home Page: Integration of a navigation bar, search functionality, book listing feature, and user account access.
* Search and Filter: A dynamic search bar with filtering options for efficient book discovery.
* Book Listing: Interface for users to list books, including a form to input book details.
* Point System Integration: For book exchanges, reflecting the user’s B-Bucks balance.
* Risks: User interface may not be intuitive for all users; Potential issues with real-time data updates.

Backend:



Block diagram for the backend

Backend Development (Spring Boot, Postgres):

* User Account Management: Secure storage and management of user data.
* Book Listings Database: Storing book details (title, author, condition, etc.), managing listings, and transaction records.
* Point System Algorithm: Creating an algorithm to assign value to books in B-Bucks.
* Risks: Data security concerns; ensuring robustness and scalability of the server. Potential issues with real-time data updates.
* API Integration: When a user types in a book to search, use the Libgen API to retrieve the book’s ISBN number and then the BooksRun API to retrieve the price of the book.

| **Features** | **Task** | **Estimated Effort** |
| --- | --- | --- |
| **Frontend** |  |  |
| Landing Page | Create an engaging landing page | 1 week |
| Account Creation | Develop account creation and login pages | 1 week |
| Home Page | Implement home page with navigation bar | 1 week |
| Search | Create search bar and page for searching listings | 2 weeks |
| Book Listing | Build form for book listings | 2 weeks |
| Point System | Integrate points balance and transactions | 2 weeks |
| **Backend** |  |  |
| User Authentication | Set up user authentication | 2 weeks |
| Database Management | Set up PostgreSQL database | 2 weeks |
| Listing Service | Develop book listing management | 3 weeks |
| Points Service | Implement points system | 3 weeks |
| Cloud Hosting | Host the website on Azure | 1 week |

**Teamwork:**

Our team employs a holistic collaboration approach, where every member is actively involved in all aspects of the project, ensuring equal responsibility and skill development across tasks. We break down large tasks into smaller, micro-tasks for manageability, using GitHub Issues for effective tracking and communication. We will also have regular in-person meetings to have discussions, knowledge sharing, and code reviews. By employing this collaborative approach, our team aims to build a solid, well-rounded skill set in each member, fostering a strong, adaptable, and efficient team dynamic. This strategy not only benefits the current project but also prepares each member for future collaborative and multidisciplinary projects.