

Kumaran Arulmani

Classification: Junior

Major: ECE

Tracks: Software Engineering & Academic Enrichment

- Created Back-end API
- Created About Us Page



David Day

Classification: Junior

Major: ECE and Math

Tracks: Software Engineering & Academic Enrichment

- React component development
- Unit testing



Matthew Jiang

Classification: Junior

Major: ECE

Tracks: Software Engineering & Academic Enrichment

Role: Front-End Developer

Primary Contributions:

Designed navbar

Implemented sorting/filtering options



Vishruthi Ramaswamy

Classification: Junior

Major: ECE

Tracks: Software Engineering & Academic Enrichment

Role: Front-End Developer

- Unit Tests
- Designed Book & Author pages



Siddhartha Shetkar

Classification: Junior

Major: ECE

Tracks: Software Engineering & Academic Enrichment

Role: Front-End Developer

- Designed results & instance web pages
- Setup routing between web pages
- Implemented database fetches/populated webpages



Jaino Vennatt

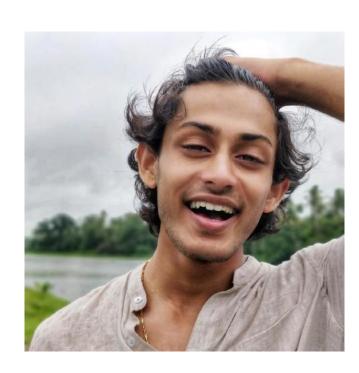
Classification: Senior

Major: ECE

Tracks: Software Engineering & Academic Enrichment

Role: Back-End Developer

- Created database and migrated data from APIs
- Worked on Back-End Server
- Did back-end testing



Booklopedia: The Internet Book Database

Purpose:

- Curates information about modern English books
- Allows for robust search capabilities through authors, books, genres, ratings, etc.
- Centralized location for all information regarding authors, books, and genres

Users:

- Online book lovers
- Book lovers who want a dedicated place to browse recent popular books

Frameworks Used

- Front-end
 - React
 - Material-Ul
 - React-Router
 - Jest/Enzyme unit testing
 - WebDriverIO integration testing
- Back-end
 - Node.js
 - Express
- Database
 - MongoDB
 - Postman database testing

Technology Stack

MongoDB	Express	React
 Classroom experience let us be more familiar with MongoDB Popular NoSql framework Straightforward API 	 Similar to Javascript, so Node.js was easy to pick up Native support for MongoDB 	 Many popular frameworks support React React is a very popular front-end framework The React Component system let us reuse code and decrease duplication.

Self-Critique

What we did do well?

- Intuitive UI
- Substantial Refactorization of Code
- Full Functionality

What we could improve on?

- Add more sorting/filter options
- Better Visuals, UI Design
- Display more statistics/data on instance pages
- Add user profiles for a more personalized experience

What did we learn?

- Integrating front end & back end in web development
- How to use design patterns to make more adaptable and readable code

Developer Critique

What did they do well?

- Lot of options for the users
- Visually aesthetic design

What they could improve on?

- Add loading icons, sometimes website rendering lags (dark mode)
- Slight UI bugs (carousel shows sliders during videos, grid display spacing looks inconsistent)
- Make all clickable links look clickable (link to homepage doesn't indicate visually that it is clickable)

What did we learn from their website?

- We could improve our databasing, they have lots of images that load quickly, while ours does have some lag time
- We would benefit from using smaller fonts and a more consistent theming

What was confusing about their website?

- The lite/dark mode button seemed a little misleading; might have been intended to work that way
- There is not good distinction in the two sections for the home page



SINCE 2020 ____

