EE 461L Term Project Report

Team : The Bookish People
Project : Booklopedia
Canvas Group : morning-2
Github Repository : https://github.com/VishruthiR/EE461L_IDB
Website Link: http://booklopedia.appspot.com/

Meet the Team :

Kumaran Arulmani
Email: kumaranarulmani@utexas.edu
Github Username: @kumosumo

Vishruthi Ramaswamy
Email: vishruthi.ramaswamy@utexas.edu
Github Username: @vishruthir

Jaino Vennatt
Email: vennatt.jaino@gmail.com
Github Username: @jainovennatt

David Day
Email: david.day@utexas.edu
Github Username: @davidday99

Matthew Jiang
Email: jiang.matthew22@gmail.com
Github Username: @myj99

Siddhartha Shetkar
Email: sshetkar@gmail.com
Github Username: @sshetkar3858

Motivation :

Our motivation for making this website was to combine a lot of our favorite book models from different websites (google, amazon, barnes & noble) and turn it into one cohesive website to find all the information. We also want to implement a user profile for the website where users can personalize their experience to fit their interests.

Users :

We expect our primary users to be online book lovers (users that would usually engage with Google Books or Amazon Books). Primarily, we want to target book lovers who want to be a part of an online community to share their interests in the form of personal favorites and personal recommendations.

Phase Leads :

Phase I - Matthew Jiang
Phase II - Kumaran Arulmani

User Stories :

Name : **1**
Genre Description
Assignee :
Siddhartha Shetkar
Description :
As a user, I want to be able to see descriptions of various genres and so I can further explore different areas of literature.
Development Process :
Create descriptions to put in a database with corresponding type of genre. On the genre page display the description.
Estimated Time to Completion :
2 hours
Actual Time to Completion :
2 hours

Name : **2**
Editor’s Choice Carousel
Assignee :
David Day
Description :
As a user, I want to be able to see the "Editor's Choice" of books by a favorite author or of a certain genre so I can further explore books that may fit my interest and likings.
Development Process :
Create a carousel of images of books pulled from a database by the same author (for author page), or classified as the same genre (for genre page).
Estimated Time to Completion :
2 hours
Actual Time to Completion :
2 hours

Name : **3**
Book Summary
Assignee :
Vishruthi Ramaswamy
Description :
As a user, I want to be able to read short descriptions (blurbs) of books so I can get a sense of what the book is about and gage my interest.
Development Process :
Using API store data on book summaries in the database. Display corresponding summary to book on page.
Estimated Time to Completion :
2 hours
Actual Time to Completion :
2 hours

Name : **4**

Book Images

Assignee :

Jaino Vennatt

Description :

As a user, I want to be able to visualize different books that I am searching, so when I am looking to purchase a book I have an idea of what the book looks like.

Development Process :

Store images from API of books. Display book images on corresponding pages.

Estimated Time to Completion :

2 hours

Actual Time to Completion :

2 hours

Name : **5**

Navigation Bar

Assignee :

Matthew Jiang

Description :

As a user, I want to be able to easily access the different pages the site has to offer so I can navigate through the website with ease and find information quickly.

Development Process :

Have links to different pages on a side bar available on all pages.

Estimated Time to Completion :

2 hours

Actual Time to Completion :

2 hours

Name : **6**
About Page
Assignee :
Kumaran Arulmani
Description :
As a developer, I want to be able to see the different contributions that went into the website and GitHub statistics on interactions/commits with the project so I can better understand everyone's contribution.
Development Process :
Have a dynamic webpage pulling data from the GitHub regarding each team member's profile (picture, biography, commits, interactions) and overall project statistics.
Estimated Time to Completion :
2 hours
Actual Time to Completion :
2 hours

Name : **7**
Search Functionality
Assignee :
Kumaran Arulmani
Description :
As a user, I want to be able to search by Book Title, Author, Genre and Publisher so I can further explore literary works related to one another.
Development Process :
Have search (input) fields for each model, create a query based on input and display the results from the database onto the Results page (with Pagination functionality).
Estimated Time to Completion :
15 hours
Actual Time to Completion :
20 hours

Name : **8**

Accessibility & Design

Assignee :

Vishruthi Ramaswamy

Description :

As a user, I want to be able to view information easily and quickly, and the website should be inviting to use.

Development Process :

Use Material UI along with React to create an aesthetically pleasing and easy to use website design. Elements should be easy to read and can traverse through the website with no difficulty.

Estimated Time to Completion :

17 hours

Actual Time to Completion :

19 hours

Name : **9**

Pagination for Search Results

Assignee :

Jaino Vennatt

Description :

As a user, I want to be able to view my search results in an organized manner (only showing 10 results per page) and sift through the pages to be able to read information quickly and easily.

Development Process :

Use Material UI along with React to create a pagination bar (using the template provided by Material UI). Each linked results page should send a new query to the database to pull corresponding results of the query (Page 1: Results 0-9, Page 2 : Result 10-19).

Estimated Time to Completion :

12 hours

Actual Time to Completion :

15 hours

Name : **10**

Dynamically Created Model Pages

Assignees :

David Day & Sid Shetkar

Description :

As a user, I want to be able to access a page for every search result (if I search for a book, I am able to click on the book to be redirected to a page with more information - same for other models) to further organize and get more information about each search result.

Development Process :

Create a general template for each model page (Book, Genre, Author) using Material UI & React, make the parameters for information variables that can be passed in from a query (for example: If I click on the Book Title of "The Great Gatsby" the corresponding author, ISBN, book summary, etc. should populate on the Book page).

Estimated Time to Completion :

15 hours

Actual Time to Completion :

17 hours

Name : **11**

Navigation Bar (Part 2)

Assignee :

Matthew Jiang

Description :

As a user, I want to be able to easily access the different pages the site has to offer so I can navigate through the website with ease and find information quickly.

Development Process :

Have links to different pages on a side bar available on all pages. Recreate using Material UI & React, also add search functionality to Navigation Bar. Should be available on all pages.

Estimated Time to Completion :

5 hours

Actual Time to Completion :

7 hours

Name : **12**
Author Biographies
Assignee :

Description :
As a user, I want to be able to read short biographies of authors so I can further explore my favorite authors and their stories.
Development Process :

Estimated Time to Completion :
5 hours
Actual Time to Completion :

Name : **13**
Author's Popular Books
Assignee :

Description :
As a user, I want to be able to receive suggestions on other popular books by an author I am interested in so I can explore other literary works by the same author.
Development Process :

Estimated Time to Completion :
3 hours
Actual Time to Completion :

Name : **14**
User Reviews
Assignee :

Description :
As a user, I want to be able to post and read reviews on books so I can gather further information about the book and help share my thoughts with other readers as well.
Development Process :

Estimated Time to Completion :
6 hours
Actual Time to Completion :

Name : **15**
Book Purchase Links
Assignee :

Description :
As a user I want to be able to view/receive multiple links to websites where I can purchase a particular book online so I can compare prices and deals between sites easier.
Development Process :

Estimated Time to Completion :
5 hours
Actual Time to Completion :

Name : **16**

User Profile

Assignee :

Description :

As a user, I want to be able to create a user profile, so I can have a custom wishlist and like/follow threads to stay updated with the newest releases by my favorite authors.

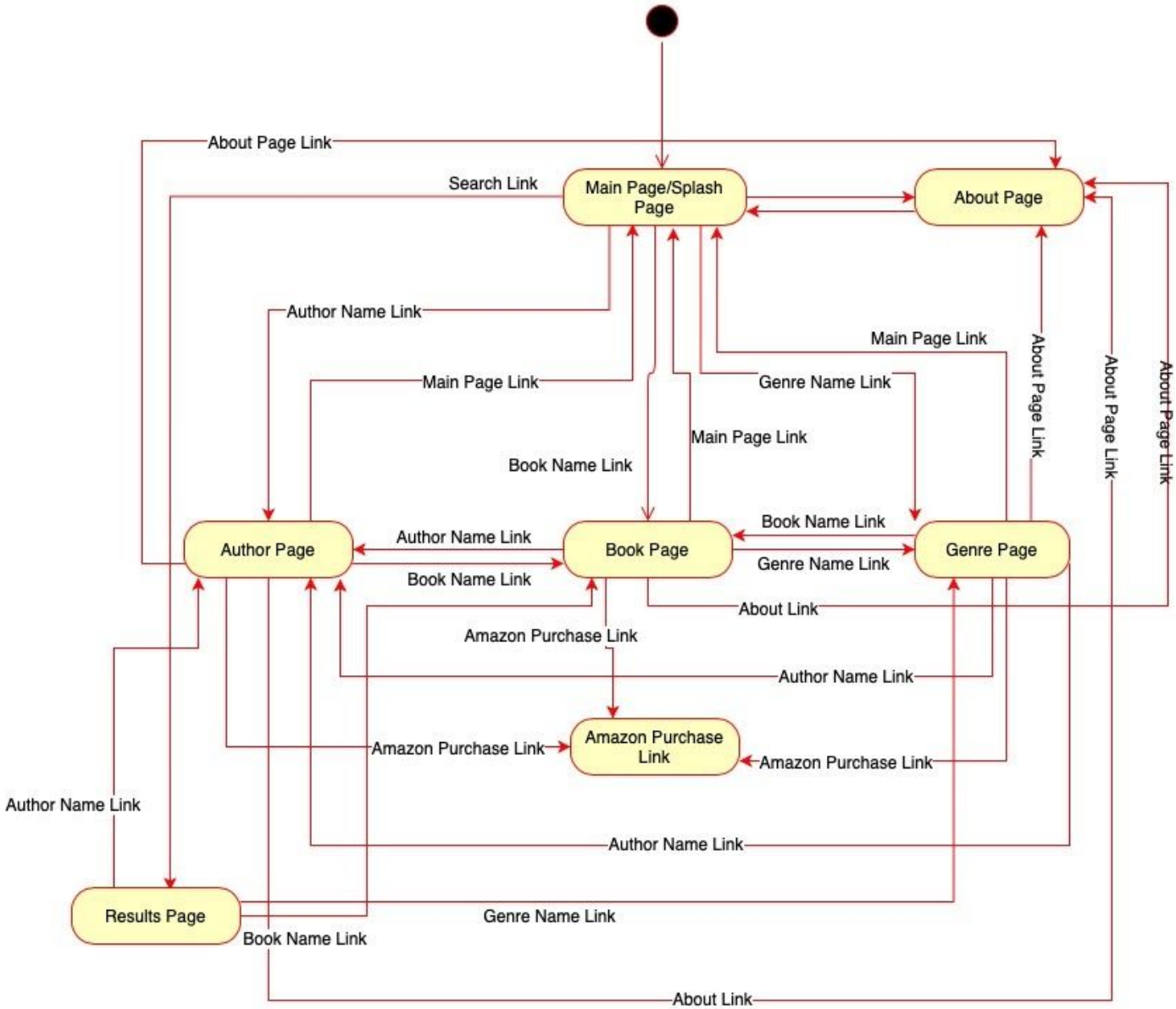
Development Process :

Estimated Time to Completion :

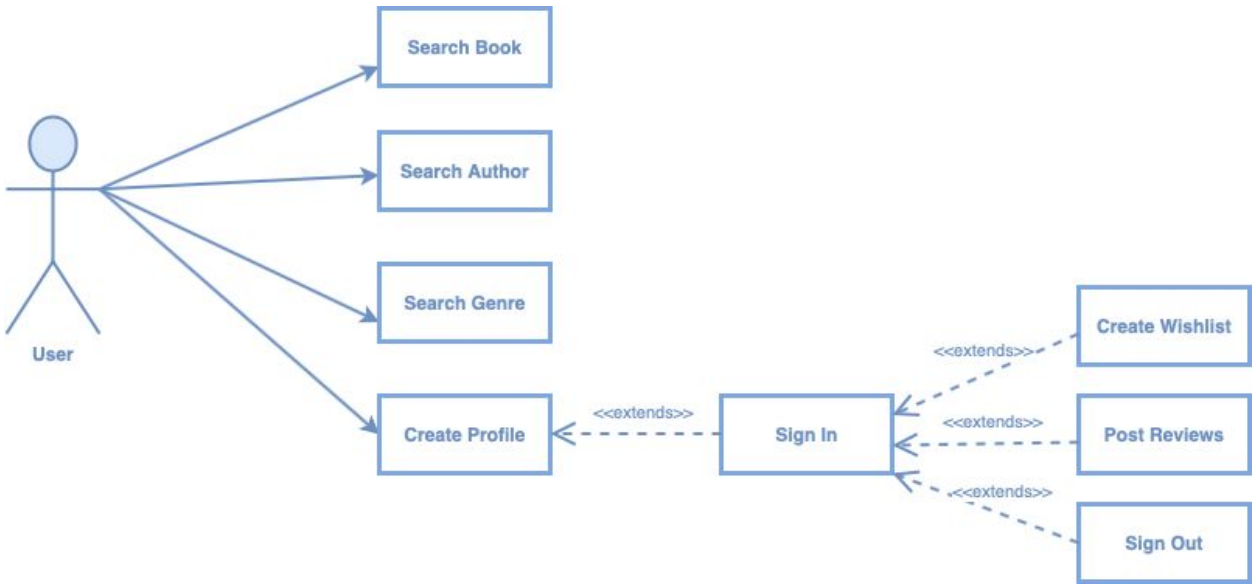
10 hours

Actual Time to Completion :

UML Diagram :


About Page Link
Search Link
Main Page/Splash Page
About Page
Author Name Link
Main Page Link
Main Page Link
Genre Name Link
About Page Link
About Page Link
About Page Link
Main Page Link
Book Name Link
Book Name Link
Author Page
Author Name Link
Book Page
Genre Page
Book Name Link
Genre Name Link
About Link
Amazon Purchase Link
Author Name Link
Amazon Purchase Link
Amazon Purchase Link
Amazon Purchase Link
Author Name Link
Author Name Link
Results Page
Genre Name Link
Book Name Link
About Link

Use Case Diagram :



Design :

Our website starts on the splash page, where we plan on including the following attributes: a navigation bar (links to the home page and about page), and explore links (to different books, genres, and authors). We plan to have the navigation bar available on every page. Other pages we also have are book, genre, and author pages. Each of these pages will be dynamically generated from a base template and offer detailed information on each particular instance. Our base template for the author, genre, and book pages include author biography description, reviews, purchase links, genre description, popular books, popular authors, and book summaries. These pages are designed to be accessed through links on the homepage (editor's choice and explore) as well as through the search functionality. Our search functionality (which will be available through the Navigation Bar) will lead to a Results page - which will display the results of a query in a formatted order (pagination will be available on this page). Each result displayed will be a link to that particular model's page (a search for "J.K. Rowling" should return a link to the "J.K. Rowling" Page where the user may obtain more information about the specific model).

Implementation :

Phase I

We implemented static web pages for each of our models and had all overlapped information link to the corresponding webpages. For example, an author's name on a book web page would link to the web page for that author. We also queried the Github API for commit/issue/unittest information about each of our team members. Finally, we implemented a sidebar for easy access between the web pages of our models.

Phase II

We implemented dynamically generated websites (based on a generic template for each model). We also implemented query calls to the database to get information about a specific model instance. These query calls were paginated on the server side because it would slow down the website to send back all the data in one response. For example, to query all books would return almost 25000 books which severely impacted load time. We filled the MongoDB database with information about our models using our REST APIs through python scripts. We also implemented a lot of new intensive testing for our Front End, Back End, & Middleware.

Testing:

Phase I

All of our testing for Phase I was completed by hand and user testing. All pages implemented thus far have been static (apart from the "About" page) so to test the feasibility of this page we viewed the pages on different screen sizes (laptop, tablet, phone) to make sure content was resized correctly (and dynamically). We also monitored the About page throughout the course of our working sessions to see the real time update of information. We hope to do more intensive testing in the upcoming phases once our pages are dynamically produced.

Phase II

With our pages being dynamically produced, we were able to implement more intensive testing in this Phase from the back end (database) to middle end (java/javascript that allows us to query the database) to front end (web pages/ui design). We tested both our front and middle end using a combination of Jest and Enzyme. Combining the libraries helped us test queries for the database to see if we could pull stored data as well as simulate accessing web pages. We additionally used Postman to test the backend to make sure database entries were valid and exceptions were handled appropriately.

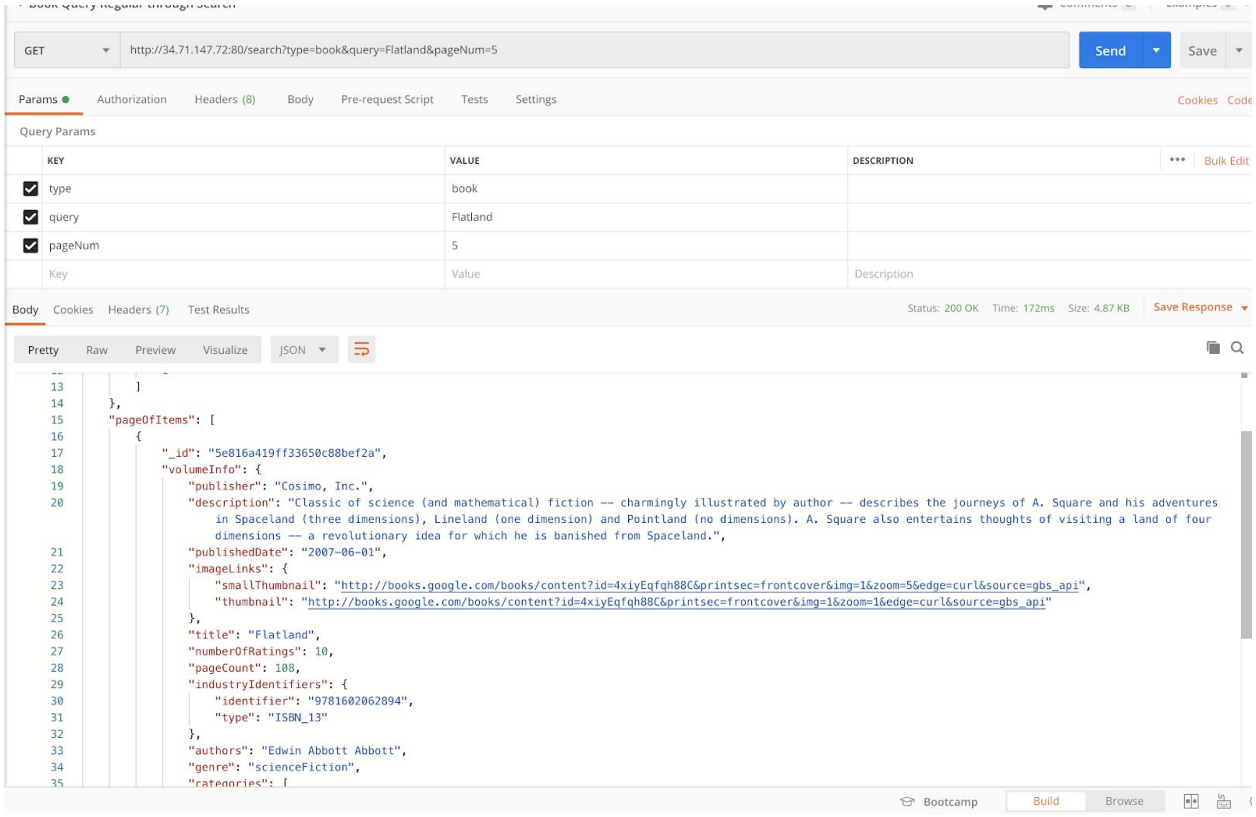
Jest/Enzyme:

- Search Bar
- Re-rendering page correctly updates components after additional server call
- Improper URLs do not break the website
- Individual React components work as intended
- Server calls can give us dedicated information based on book/author/genre
- Server side pagination

Postman:

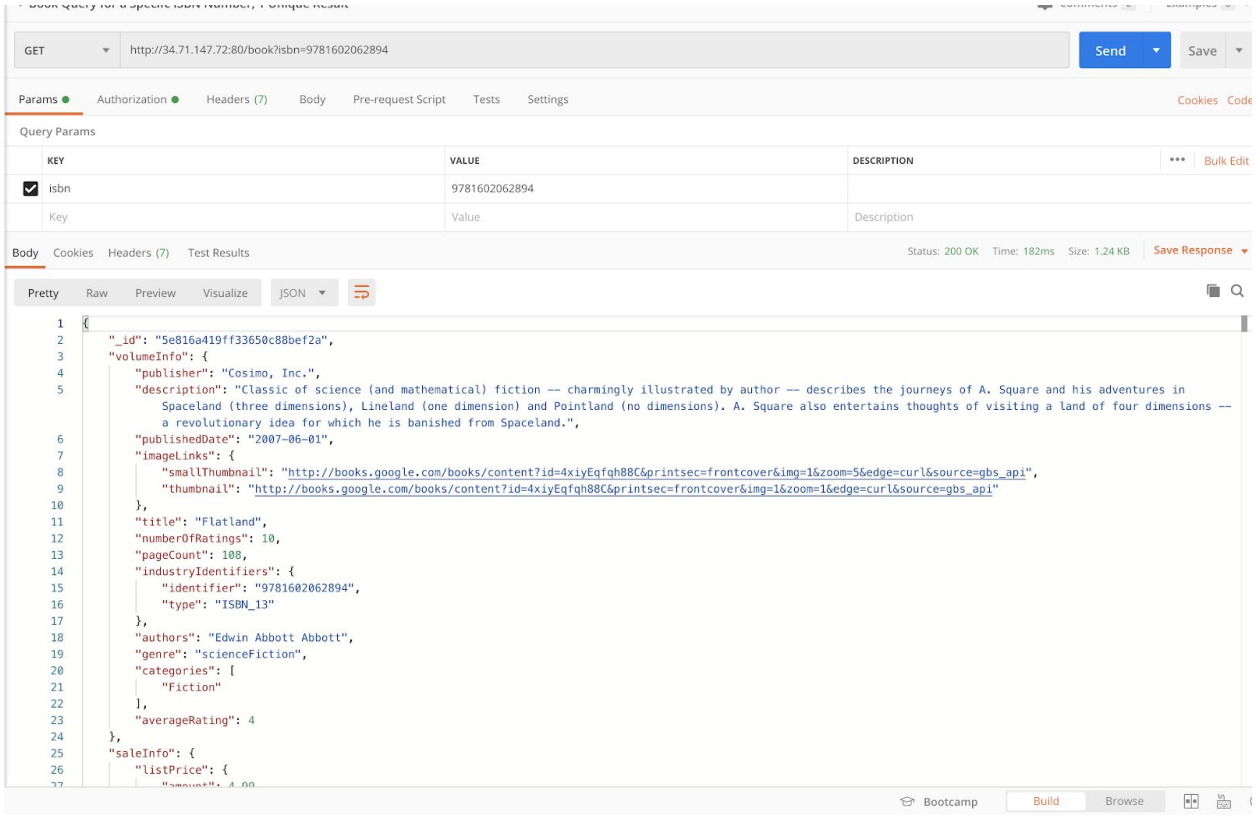
- Database Testing Phase 1
    - Put entry in database, check if entry is actually in the database
    - If there is no such entry in database, errors handled appropriately
    - Test for duplicate entries in database
- Backend API Testing Phase 2
    - Test every back-end API Call possible and check returned values
    - Double check that correct values are returned for random search
    - Double check that correct values are returned for specific search
    - Double check values for calls regarding books, authors, and genres

Back End API Test Example 1:



- Inputs:
  - Type - specifies model
  - Query - string to search
  - Page Number - specifies which page of results/ pagination
- Outputs:
  - JSON Object - array of all books pertaining to search as well as all fields regarded to each book
- Goals:
  - Check if our query returns appropriate book results for the sent query

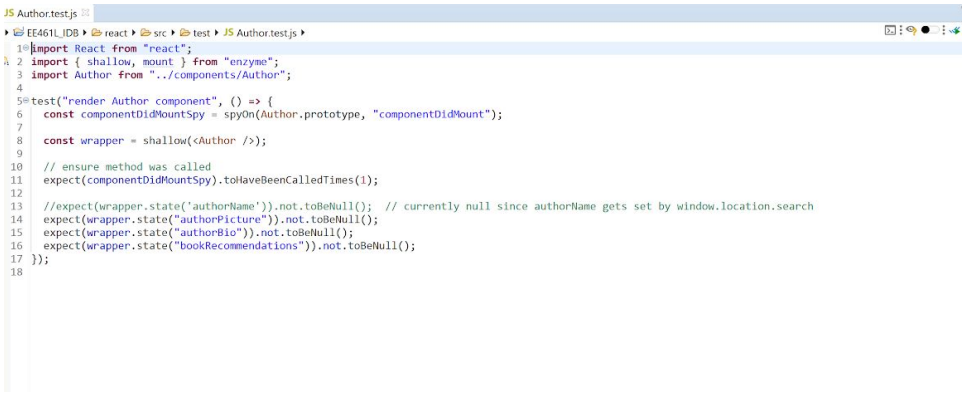
Back End API Test Example 2:



- Inputs:
    - ISBN - specifies unique ID of book being searched

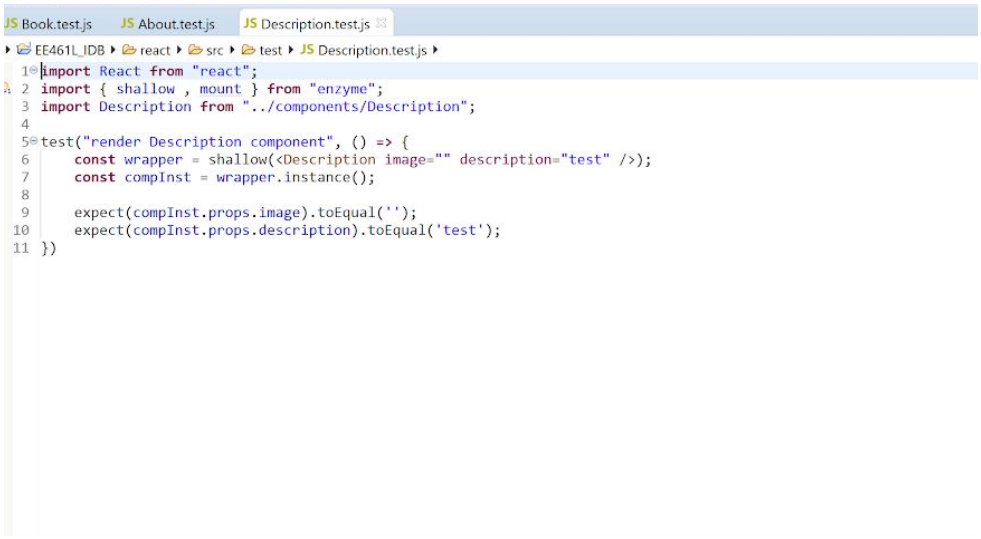
- Outputs:
    - JSON Object - returns the book(singular) that matches the unique ISBN as well as all the values associated with this book
- Goals:
    - Check if our query returns appropriate book result for the sent query

Front End API Test Example 1:



- Goals:
    - Check if Author component states are initialized properly.

Front End API Test Example 2:



- Goals:
    - Check properties passed into the Description component are set correctly.

Models:

1. Books
    - Number of Instances: ~25000
    - Attributes: Author, Publisher, Genre, Cover, Year
    - Examples: The Road, Kite Runner, The Book Thief
2. Authors
    - Number of Instances: ~1000
    - Attributes: Name, Biography, Most Popular Publications
    - Examples: J.K. Rowling, Cormac McCarthy, Markus Zusak
3. Genre
    - Number of Instances: ~5
    - Attributes: Description, New Releases, Most Popular Publications, Most Popular Authors
    - Examples: Non-Fiction, Romance

Tools/Software/Frameworks:

We used the Google Cloud Platform to deploy the website. We used React (implemented with JavaScript) for our front end design and the Material-UI library to handle styling aspects of the website. MongoDB has been set up and filled with data using a combination of PyMongo and our REST APIs (Google Books, GoodReads, ISBN). We later pulled this data dynamically into our website by integrating NodeJS and Express with our frontend. The database will additionally hold user information (login, profile, wishlist, preferences) which we aim to complete in the next phase. For testing we used Jest, Enzyme, Postman.

Reflection:

Phase I

Overall, our team got together with a great mindset and was able to hold a long productive working session that resulted in the majority completion of Phase 1. It was helpful to sit together and work in the same room, so many team members could pair program pages, ask questions freely, and bounce ideas off of each other. Our slack also helped us stay extremely organized throughout this process - with the ability to view git commits instantly from your phone, talk to teammates about pressing issues, create polls to vote on any team decisions. We were able to work independently after our joint working session, however productivity seemed to lag since members had to wait for the response of others through slack to make any design decisions/move forward with their assignments. For the upcoming phases, it would be beneficial for us to reduce the amount of independently worked time on the project and hold more joint working sessions to keep increased productivity.

Phase II

Overall, our team was able to pull together all the requirements specified for Phase II. We ran into some unprecedented issues (a World-Wide Pandemic) but our team was able to make the best of the situation. Through the use of Zoom, we hosted daily standup meetings, and longer working sessions (specific to our divided teams: BackEnd/MiddleEnd & FrontEnd). We had a great deal of work to be done on both teams and through the use of virtual meetings we were able to collaborate (screen sharing, chats). Our slack also helped us stay organized by giving us immediate updates on any issues, commits, and progress. Unfortunately with the situation at hand, we were not able to hold as many joint working sessions (virtually) as we would've liked, but had to each remotely complete our assigned tasks. As a result of this, our progress seemed to be slower than our previous Phase. In addition, our assigned tasks for this Phase were a great deal harder and complicated compared to the previous Phase. Switching the UI design (using React & Material UI) and querying the database were tasks that we were all not well versed in, so the learning curve was steeper. In conclusion, our team was able to meet the requirements for Phase II but lacked efficiency. We plan to improve this issue in future phases by planning a more strict timeline with clearly assigned tasks from the beginning that allows room for unprecedented issues and hiccups along the way.