

Capstone 2 Proposal: My Booklist

Mandy Blott

Project Goals and Overview: The goal of this web app would be to provide a convenient and efficient way for users to search for popular books and create lists of books they're interested in reading, to reference later when they're ready to read a new book. Users would be able to search for either fiction or nonfiction books that made the New York Times Bestseller list during various time periods, and view detailed information about those books. For books that pique the user's interest, they could either add the book to an existing list on their account, or create a new list (e.g. "Leisure/Fiction," "Programming", "Cookbooks"). The user would also have the option to later delete a book from their lists entirely, or to move the book to a different list (e.g. "Read and Loved", "Unfinished"). This project would use the New York Times Bestseller API (<https://developer.nytimes.com/docs/books-product/1/overview>).

Approach: On the backend, I would use Node/Express to create my own custom API, which would include GET and POST routes for signup and login (including user authentication using JSON Web Tokens), a GET route for getting new books (which will itself make an API call to the NYT Bestseller API), and GET and POST routes for accessing and modifying the user's booklists. I would create model classes for users and for booklists, which would include custom SQL query methods that could be used in the routes to query the database via Node pg. To validate request data before querying the database, I would use JSON schema to define what data is expected and/or required. On the frontend, I would use React router to implement client-side routing. This would allow me to create a website with single-page app functionality, using AJAX requests to the backend to update the database and the user view. I would use React's useState hook to store and update the user's profile information and booklist data, and the useEffect hook to re-render components if/when that data changes. Finally, I would create a suite of tests for both the backend and the frontend using jest.

User Flow: On the home page, the user would first see a description of the web app, as well as forms for login and sign up. Upon logging in, the user would first see the My Lists page, which gives them a form for creating new lists, as well as a view of the lists they've already created. They could click on any book in a list to view that book's detail page. If the user wished to search for new books, they could click on the link on the nav bar to go to the Search Books page. This page would provide a form where they could search for either fiction or nonfiction books that were on the bestseller list on a particular date. Below the search form, there would be a default list of 10 books from the current fiction bestseller list, which would update upon submission of the search form. Again, the user could click on any book displayed to view the book's detail page. On the book detail page, the user could view the book cover and description, click on a link to check out the book on Amazon, and add or remove the book from one of their lists.

Stretch goals: If I'm able to implement stretch goals, one idea would be for users to have the option to rank books on a list, so they could put books at the top that they're most interested in. Another possible stretch goal would be to provide a form on the book detail page, so that users could write and save their notes about a book after reading it.