Zdenek Andrew Bocek

January 4, 2024

# My Project Proposal: Look Up Book

1. I will use NodeJS, Express, React, React Router, Babel, HTML, CSS, and possibly some other tech stacks to complete my project.
2. The front-end UI will be the focus of my project since I will be using React and React router. All requests will go through the front end in the browser, instead of retrieving requests from the server side.
3. This will be an app that should auto fit any screen on which it appears.
4. I will make an API and call it "Look Up Book". This API will hold details on authors of books, how many literary prizes they have won, details of the books they have written, what year the books came out, etc. It will be a recommendation engine for people looking up genres of books, or regions of the world, or authors of books, or what sort of award they want the book they read to have won. It will not feature authors who have been on "bestseller lists", because all an author needs to do to end up on a bestseller list is have most American consumers in a certain city read their book. The New York Times bestseller list simply tells you that most people who live in New York City have read that author’s book, but IT DOES NOT tell you whether that author won any literary prizes at all for their book. You do not need to win any awards whatsoever to end up on a bestseller list as an author. My API will serve as a recommendation engine for people who are looking to read books by award-winning authors. These awards will be serious awards that require real achievements in writing to win, including the Nobel Prize in Literature, the Pulitzer Prize, the Booker Prize, the International Booker Prize, PEN America Literary Awards, the National Book Award, the Costa Book Award, or the PEN Faulkner Award. They can search for books using multiple metrics, and they can post the books they choose to their own account or put them on a shopping list. They can also update their lists and delete books from their lists. They will also be able to post a like or not-like regarding the book. They can also recommend a book that has not won an award but is by an award-winning author. People can post a like or dislike for a recommendation from others. Also, to describe the books that people suggest, they can select from keywords to describe it, rather than giving their subjective opinions. People will not be able to post entire comments on what they think of a book, nor will they be able to comment on each other’s comments. This will not be like YouTube. People will also be able to withdraw or delete a suggestion if they suggested it. If, however, it becomes a verified book that is from an award-winning author, they will no longer be able to withdraw it. Also, I do not count a song as having the same level of literary complexity as a book, therefore, people will not be able to recommend song lyrics, just books. Someday, I might have this app go public and people will be able to recommend award-nominated books, because there are many authors that are still nominated for one of these critically acclaimed awards but who did not win.
5. I predict that the demographic that will use my app will be from every age group, every ethnicity, every race, and every gender. It is up to readers if they want to read a book that other literary professionals have rated as being beyond in terms of creativeness and intelligence over a book that has simply appeared on a bestseller list.
6. I will create my own API and populate it with data. I will try to collect data from libraries, as well as any other place where I do not infringe on copyright.
7. a. My database schema will have the user table at the center, from which the user will be able to send a get request to multiple tables from my API. My database will have several many-to-many as well as one-to-many relationships.

b. I want to avoid a race condition between data in different tables. I also will want not more than one piece of data in one bucket. If I must store multiple data pieces in one bucket, I will want to make sure that my app does not get confused about which piece of data to send.

c. I will secure all users’ passwords, as well as the secret key to connect to my API.

d. My app will use POST, GET, DELETE, and PUT. In short, it should use every aspect of the CRUD cycle.

e. The user will land on the account access page, where he/she will need to either log in or create a new account. After that, he/she will go to the homepage. He/she will be able to request a new search for a book he/she wishes to read from the navbar on the homepage. The user will be able to select a new book based on the author, country of origin, professional awards that the author has won, genre of book, the year of the book, etc. The homepage will also show 10 random books that have won awards as a suggestion to the user. The navbar will also link to the user’s profile page. The profile page will share their account information, such as their date of birth, their username, e-mail address, and they might be able to show what genre of book they like best.

f. This goes beyond being a simple CRUD app in terms of its creativity and originality. I do not think anyone has tried to offer readers an app where they can look for books from authors that have done more than just being popular.