

CS 410 - Group Project Fall 2023

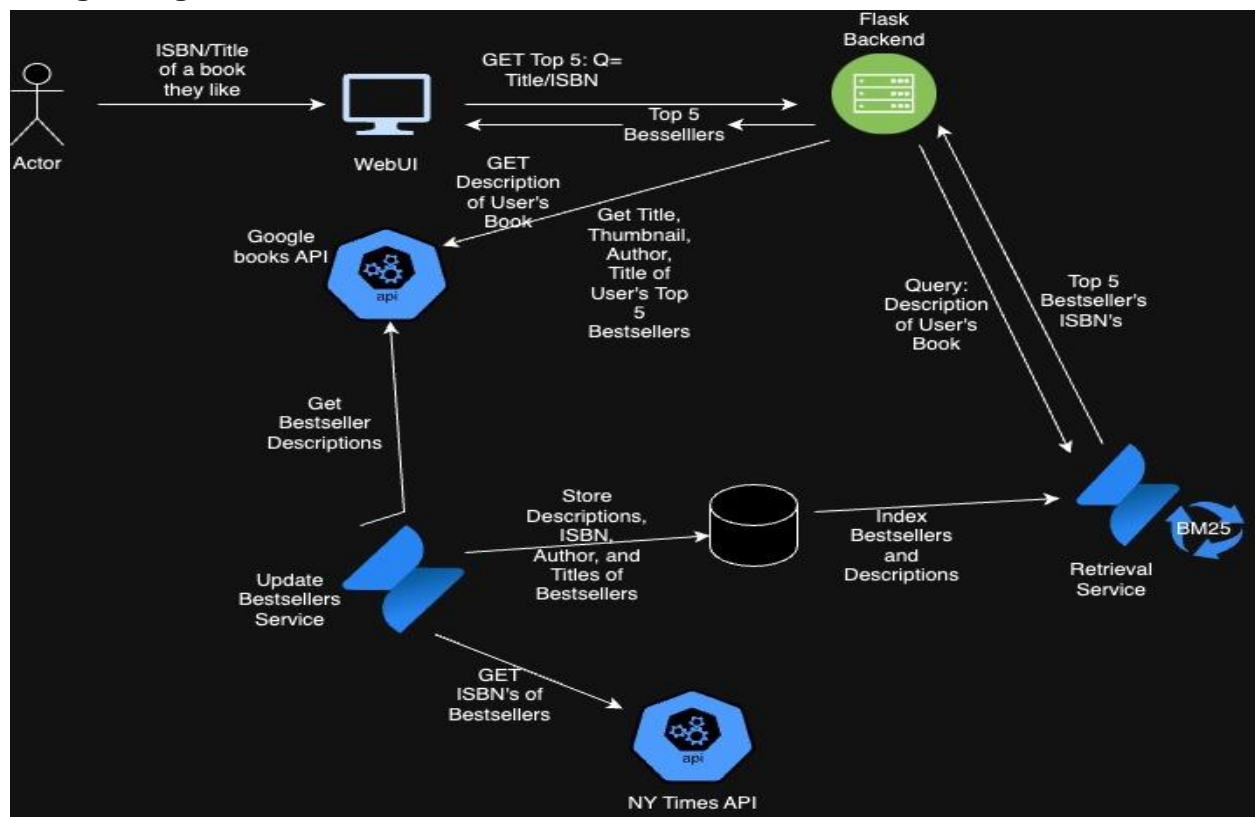
Team: EMKW

Overview of Code Function

The purpose of this code is to take an input of a book that the user is interested in and return a ranked list of 5 books on the current New York Times BestSeller List that are similar to that book. It is intended for use by those who want to find their next book to read and want something that is well liked and easy to find in a bookstore as most New York Times BestSeller books are.

Software Methodology

Design Diagram



The diagram above shows the Design Diagram for this software. The user will input the ISBN or Title of a book that they like into the Web User Interface(UI). The Web UI will then reach to the Flask Backend which gets a description of the book from the Google Books API and will reach to the Retrieval Service that ranks the books returned from the Update BestSellersService which is run separately prior to launching the WebUI currently. The Update BestSellers Service reaches out to the New York Times Books

API when it is run to get a list of all books currently on the New York Times BestSeller List (approximately 230 books).

Software Use

Usage of APIs and Functions:

1. *Django WebApp*

- We chose to use this framework since it runs on python and we could easily apply the functions that we have used in the MPs to the project. We later changed to flask.

2. *Flask Webapp*

- The flask webapp holds our backend and calls the google api and calls the retrieval service which is the communication connection between our bestsellers service. It also holds the frontend where a user can go to their local port and enter a book title or isbn.

3. *New York Times API*

- We chose to use this API to get a list of popular books to recommend to readers.
- Requires an API key that can be obtained through creating an account with NYT (<https://developer.nytimes.com/get-started>) and adding it to the config.py file. Only the Books API needs to be enabled for the key to work for this project.

4. *Google Books API*

- We chose to use this API to get additional information about books that the user can search for because it does not require additional API Keys, it is free to use, and it contains all the necessary information to run this application, including title, ISBN, description, and book cover image.

5. *NLTK Function*

- We used this function for preprocessing book descriptions, involving:
 - a. Converting words to lowercase
 - b. Removing punctuations
 - c. Removing stop words
 - d. Lemmatizing words
 - e. Stemming words

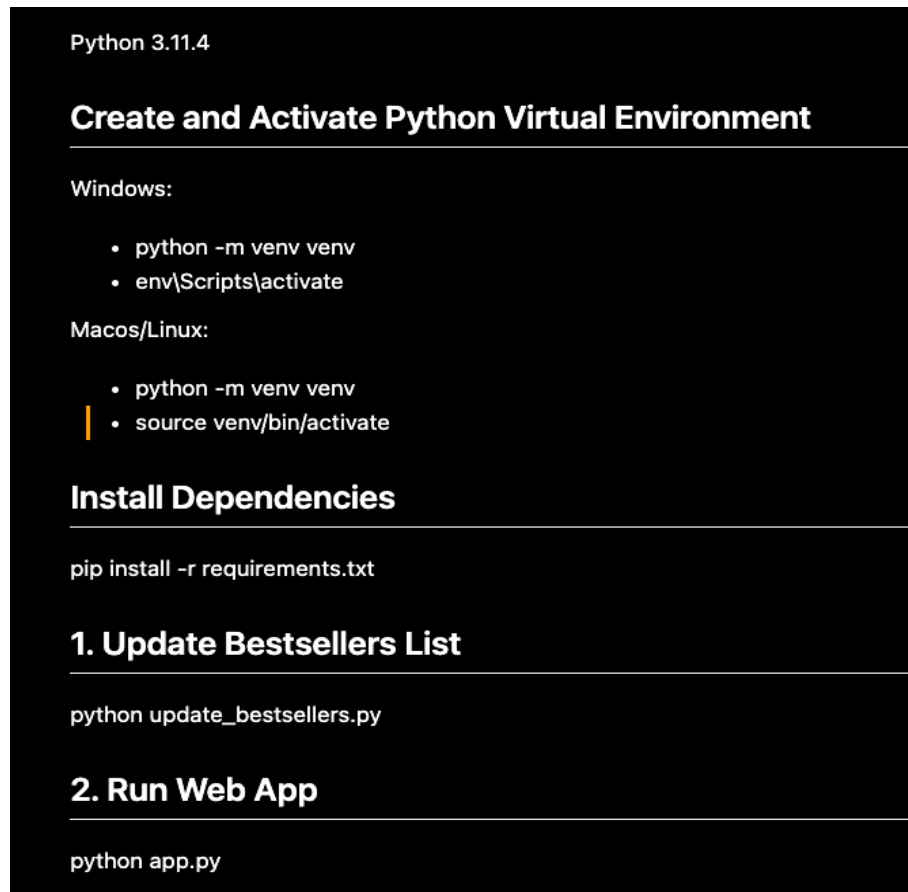
6. *Inverted Index Function*

- We initially considered utilizing the MetaPy Inverted Index to efficiently index book descriptions. However, we did not proceed with its implementation due to challenges encountered in integrating the inverted index with ranking functions using the MetaPy library.

7. *BM25 Function*

- We used this function to rank the books in the New York Times bestseller list based on similarity to the description of any book from user input.

How to use this application:



Use the Read Me within the BestSellerNudge Folder in the github for instructions on how to launch the application. The image above is a screenshot of how that Read Me looks. Once the application is launched the user can input either the title or isbn number of a book that they recently enjoyed to find other books that are similar to that one.

Team Member Contributions

Table of the contributions from each team member.

| Tasks | Time (hours) | Responsible | Language |
|--|--------------|--------------|---------------------|
| Get information from Best Seller List | 10 | | |
| Extract information from NYT API | | Emily Meyer | Python (pass JSON) |
| Formatting Information once extracted | | Emily Meyer | Python |
| Get description information from Google Books | 15 | | |
| Write a function to return a description from Google Books | | Emily Meyer | Python (pass JSON) |
| Preprocess descriptions pulled from Google Books | | Kin Yik Lam | Python |
| Creation of Inverted Index | 10 | | |
| Code Inverted Index algorithm function | | Kin Yik Lam | Python |
| Store inverted index per book for ranking | | Kin Yik Lam | Python |
| Ranking Algorithm Implementation | 10 | | |
| Code the BM25 Function | | Kin Yik Lam | Python |
| Implementing the function on an imported list | | Warren Dietz | Python |
| Back End | 10 | | |
| Exporting the results to the Results Table | | Warren Dietz | Python (pass JSON) |
| Create Endpoints | | Warren Dietz | Python |
| User Interface | 25 | | |
| Set up Framework | | Mark Falcone | HTML/CSS/Javascript |
| Write Instructions for User | | Warren Dietz | HTML/CSS/Javascript |
| Implement Search Bar (search google books) | | Mark Falcone | HTML/CSS/Javascript |
| Include drop down to pick from books? | | Mark Falcone | HTML/CSS/Javascript |
| Implement Table with Results (ranked NYT list) | | Warren Dietz | HTML/CSS/Javascript |