Overview

It is a program used to get information and create new articles from and to a database.

User guide

Users will be given the choice of entering 1 to search for an article, 2 to search for an author, 3 to list most popular venues, and 4 to add an article
If user inputs 1, they will be asked to search multiple keywords to which articles containing those keywords will match and show the results. The user will then have the option to get more information about a certain result.
If user inputs 2, they will be asked to search an author's name to which the articles with a matching author name will come out with results. The user will then have the option to get more information about a certain author.
If user inputs 3, they will be asked how many of the most popular venues they'd like to see, and then those venues will be listed.
If user inputs 4, they will be asked to enter details required for a new article to be made and an article with those details will be inserted into the database

Testing Strategy:

We used the 1k json file posted on the eClass forum to test our program. We made random inputs and tried destroying the program. Whenever we would get a result, we would check the actual json file to make sure that it matched the results we had hoped for.

Group Break-Down Strategy

Rayan:

-loaded the json file in (30 minutes)
-created the searchArticle() function (5 hours)
-created the searchAuthors() function (4 hours)
-created main function (30 minutes)
-debugging (20 minutes)

Yushu:

-created listVenues() function (6 hours)
-create addArticle() function (3 hours)
-manage the closing of MongoClient (30 minutes)
-debuggin (1 hour)

Coordination strategy:

Discord server:

- Messaged each other back and forth consistently everyday to make sure we kept updating each other on our progress

Meetings after class:

- We share the same 3 classes every morning, so we would also talk about anything that happened over the night to keep updated on the progress and what is happening. Also tried discussing things that were really hard

Github:

- Created a repository for the project and committed over 50 times with many comments on what we fixed and what we updated. This means we made sure to break down the work into meaningful small problems so that figuring out bugs and later fixes would be very easy.

Design

We utilized PyMongo to incorporate MongoDB in Python
load_json() establish MongoClient, creates collection; read from the json file and insert into the collection
listVenues() list the top n venues as the user inputs, the venue name, count of articles and number of articles that reference an article in this venue will be displayed
searchArticle() finds articles containing all keywords provided and prints details on them, then more information can be displayed on a certain article and all articles that reference that article
searchAuthors() finds amount of articles written by each person from the keyword name and displays short information, and a specific author can be chosen to show all the articles they have written.

Bugs

-using “or” semantics instead of “and” semantics. Went online to find that I needed to separate each word by quotations to instead use and semantics
-wasn’t getting results for year when searching it up, fixed it by turning the year into a string instead of an integer which allowed the search index feature to work
-conflict in indexes created: indexes were created in load_json.py but listVenues.py also created index for search
-after ”$unwind” in listVenues, “$size” operator does not work because “referenced” would not be an array after unwind, fixed by having “$group” before any operation that changes referenced