Environment:

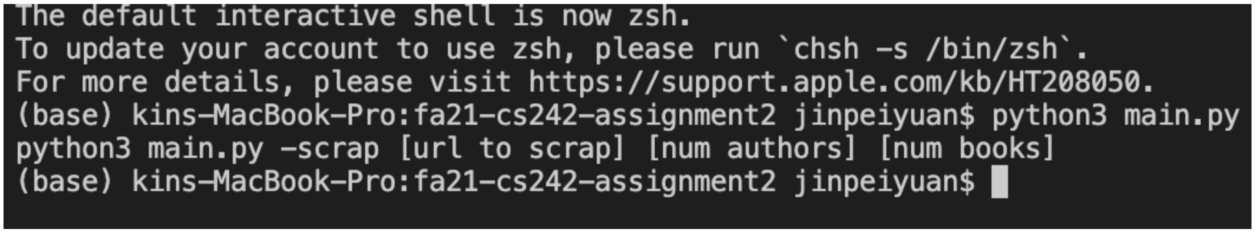
Code editor: Visual Studio code

Language: Python 3.8.5

Machine: macOS

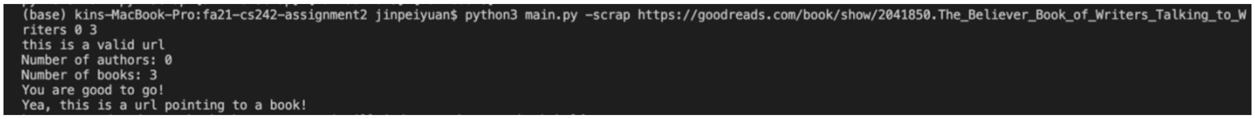
Table of contents:

1. Author information scraper file: author_scraping.py
2. Book informaiton scraper file: book_scraping.py
3. Data from database retrival and upload file: database.py
4. Python unit test file: test.py
5. File for parsing user inputs: main.py



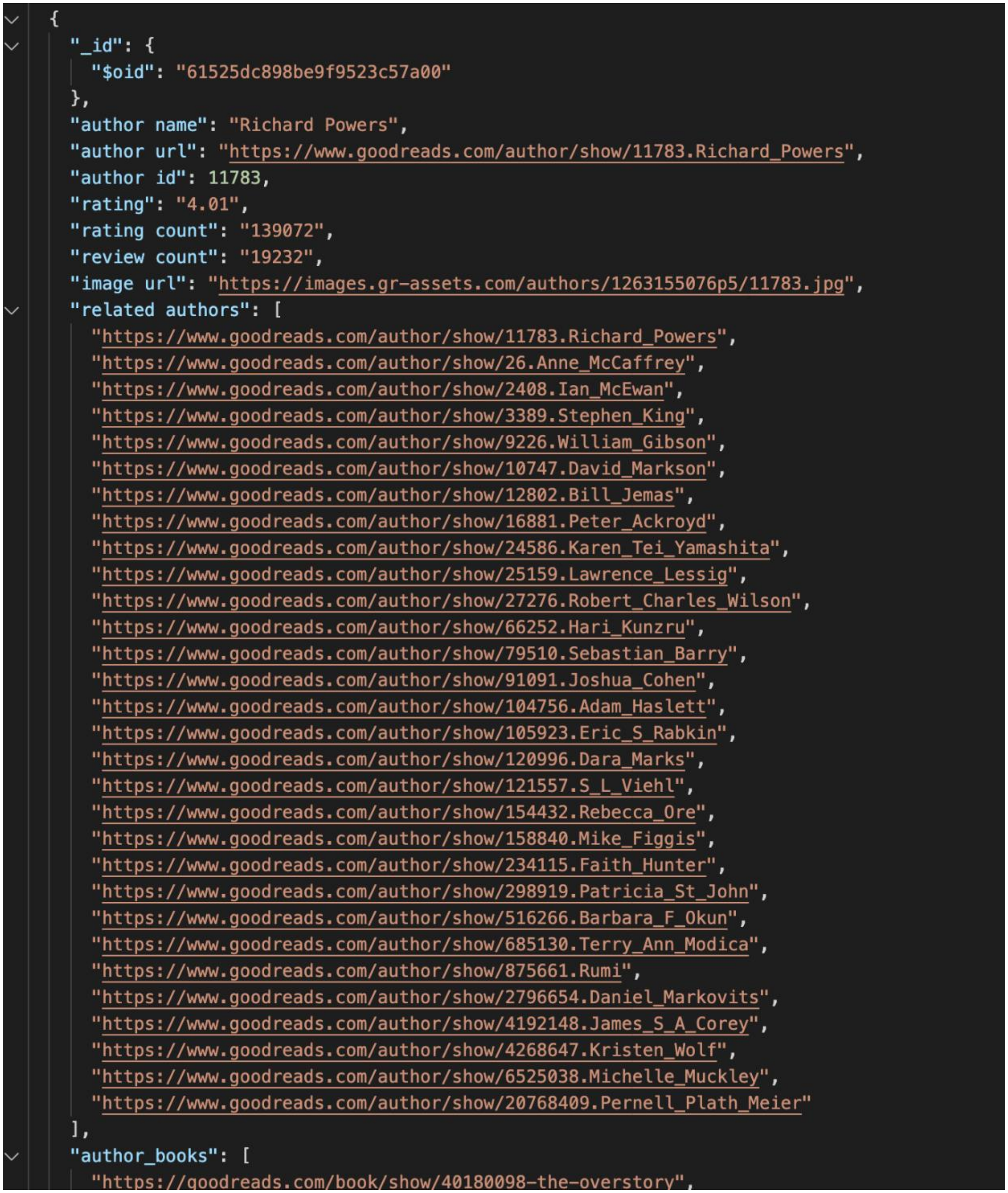
```
The default interactive shell is now zsh.
To update your account to use zsh, please run `chsh -s /bin/zsh`.
For more details, please visit https://support.apple.com/kb/HT208050.
(base) kins-MacBook-Pro:fa21-cs242-assignment2 jinpeiyuan$ python3 main.py
python3 main.py -scrap [url to scrap] [num authors] [num books]
(base) kins-MacBook-Pro:fa21-cs242-assignment2 jinpeiyuan$ 
```

This is the user command line interface that accepts user inputs and it is very user friendly: if users not correctly input stuff, it will print usage() (like CS241 would do). And if user inputted in the format, it will begin to parse the arguments.



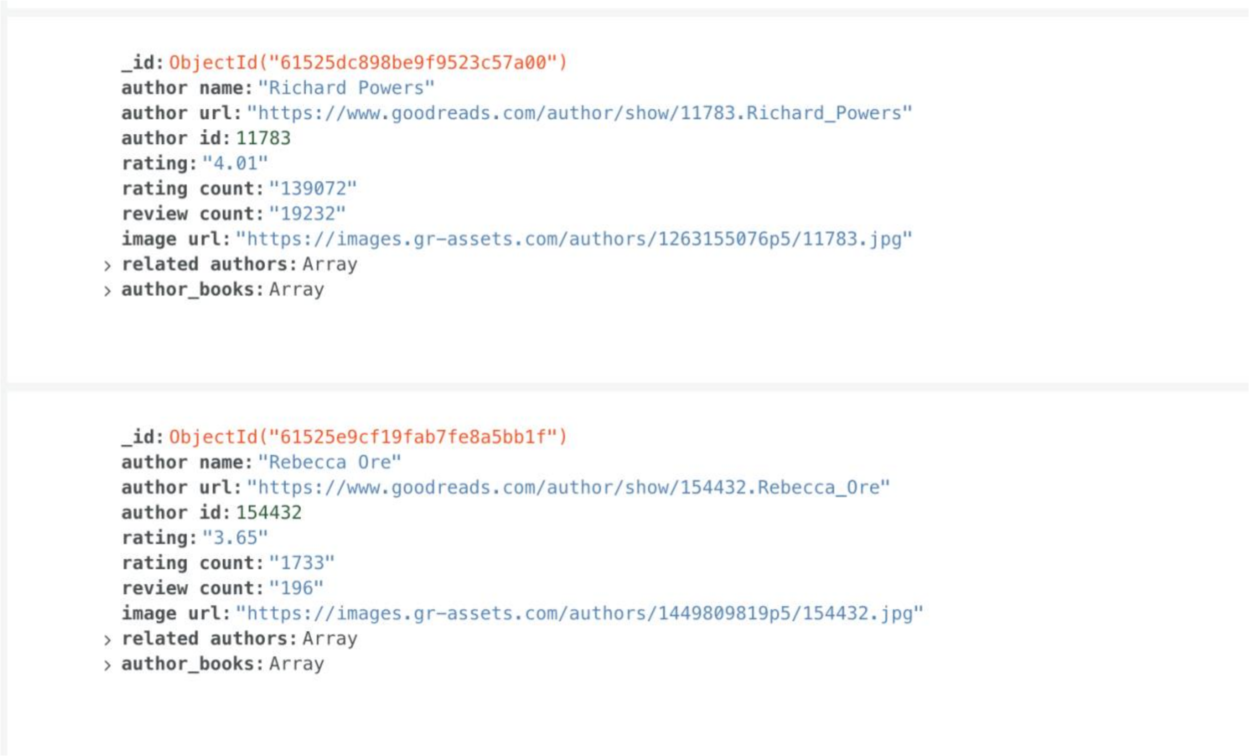
```
(base) kins-MacBook-Pro:fa21-cs242-assignment2 jinpeiyuan$ python3 main.py -scrap https://goodreads.com/book/show/2041850.The_Believer_Book_of_Writers_Talking_to_W
riters 0 3
this is a valid url
Number of authors: 0
Number of books: 3
You are good to go!
Yea, this is a url pointing to a book!
```

And, on success, it will pop up several information about the stuff user entered.



```
{
  "_id": {
    "$oid": "61525dc898be9f9523c57a00"
  },
  "author name": "Richard Powers",
  "author url": "https://www.goodreads.com/author/show/11783.Richard_Powers",
  "author id": 11783,
  "rating": "4.01",
  "rating count": "139072",
  "review count": "19232",
  "image url": "https://images.gr-assets.com/authors/1263155076p5/11783.jpg",
  "related authors": [
    "https://www.goodreads.com/author/show/11783.Richard_Powers",
    "https://www.goodreads.com/author/show/26.Anne_McCaffrey",
    "https://www.goodreads.com/author/show/2408.Ian_McEwan",
    "https://www.goodreads.com/author/show/3389.Stephen_King",
    "https://www.goodreads.com/author/show/9226.William_Gibson",
    "https://www.goodreads.com/author/show/10747.David_Markson",
    "https://www.goodreads.com/author/show/12802.Bill_Jemas",
    "https://www.goodreads.com/author/show/16881.Peter_Ackroyd",
    "https://www.goodreads.com/author/show/24586.Karen_Tei_Yamashita",
    "https://www.goodreads.com/author/show/25159.Lawrence_Lessig",
    "https://www.goodreads.com/author/show/27276.Robert_Charles_Wilson",
    "https://www.goodreads.com/author/show/66252.Hari_Kunzru",
    "https://www.goodreads.com/author/show/79510.Sebastian_Barry",
    "https://www.goodreads.com/author/show/91091.Joshua_Cohen",
    "https://www.goodreads.com/author/show/104756.Adam_Haslett",
    "https://www.goodreads.com/author/show/105923.Eric_S_Rabkin",
    "https://www.goodreads.com/author/show/120996.Dara_Marks",
    "https://www.goodreads.com/author/show/121557.S_L_Viehl",
    "https://www.goodreads.com/author/show/154432.Rebecca_Ore",
    "https://www.goodreads.com/author/show/158840.Mike_Figgis",
    "https://www.goodreads.com/author/show/234115.Faith_Hunter",
    "https://www.goodreads.com/author/show/298919.Patricia_St_John",
    "https://www.goodreads.com/author/show/516266.Barbara_F_Okun",
    "https://www.goodreads.com/author/show/685130.Terry_Ann_Modica",
    "https://www.goodreads.com/author/show/875661.Rumi",
    "https://www.goodreads.com/author/show/2796654.Daniel_Markovits",
    "https://www.goodreads.com/author/show/4192148.James_S_A_Corey",
    "https://www.goodreads.com/author/show/4268647.Kristen_Wolf",
    "https://www.goodreads.com/author/show/6525038.Michelle_Muckley",
    "https://www.goodreads.com/author/show/20768409.Pernell_Plath_Meier"
  ],
  "author_books": [
    "https://goodreads.com/book/show/40180098-the-overstory",
```

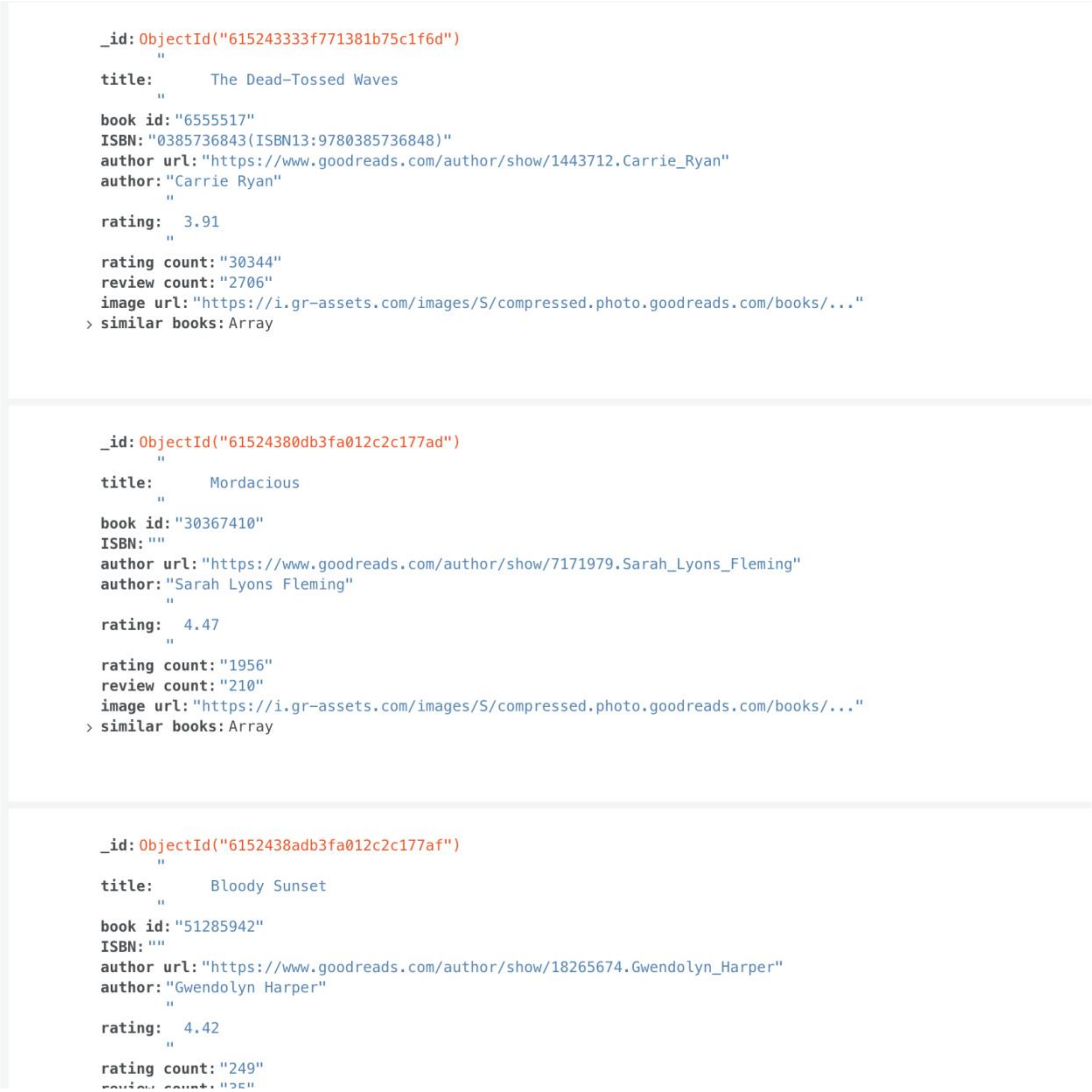
Also, my program can output the data from database as user wants into json file, and this screen shot is the prove of it in the data.json I outputted as an example.



```
_id: ObjectId("61525dc898be9f9523c57a00")
author name: "Richard Powers"
author url: "https://www.goodreads.com/author/show/11783.Richard_Powers"
author id: 11783
rating: "4.01"
rating count: "139072"
review count: "19232"
image url: "https://images.gr-assets.com/authors/1263155076p5/11783.jpg"
> related authors: Array
> author_books: Array
```

```
_id: ObjectId("61525e9cf19fab7fe8a5bb1f")
author name: "Rebecca Ore"
author url: "https://www.goodreads.com/author/show/154432.Rebecca_Ore"
author id: 154432
rating: "3.65"
rating count: "1733"
review count: "196"
image url: "https://images.gr-assets.com/authors/1449809819p5/154432.jpg"
> related authors: Array
> author_books: Array
```

This is the database interface on pymongo of authors

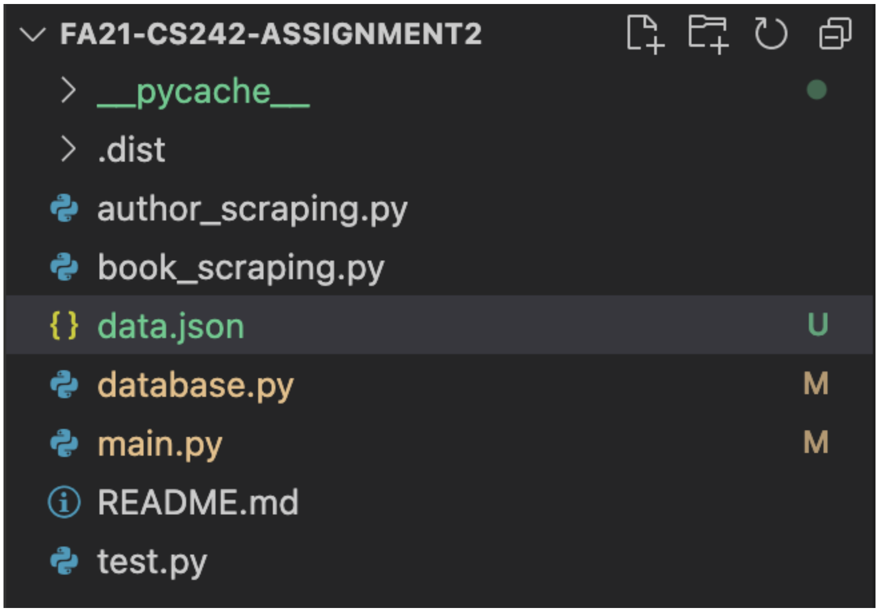


```
  _id: ObjectId("615243333f771381b75c1f6d")
        "
  title:       The Dead-Tossed Waves
        "
  book id: "6555517"
  ISBN: "0385736843(ISBN13:9780385736848)"
  author url: "https://www.goodreads.com/author/show/1443712.Carrie_Ryan"
  author: "Carrie Ryan"
         "
  rating:   3.91
         "
  rating count: "30344"
  review count: "2706"
  image url: "https://i.gr-assets.com/images/S/compressed.photo.goodreads.com/books/..."
> similar books: Array


  _id: ObjectId("61524380db3fa012c2c177ad")
        "
  title:       Mordacious
        "
  book id: "30367410"
  ISBN: ""
  author url: "https://www.goodreads.com/author/show/7171979.Sarah_Lyons_Fleming"
  author: "Sarah Lyons Fleming"
         "
  rating:   4.47
         "
  rating count: "1956"
  review count: "210"
  image url: "https://i.gr-assets.com/images/S/compressed.photo.goodreads.com/books/..."
> similar books: Array


  _id: ObjectId("6152438adb3fa012c2c177af")
        "
  title:       Bloody Sunset
        "
  book id: "51285942"
  ISBN: ""
  author url: "https://www.goodreads.com/author/show/18265674.Gwendolyn_Harper"
  author: "Gwendolyn Harper"
         "
  rating:   4.42
         "
  rating count: "249"
  review count: "35"
```

And this is the data interface for books on pymongo

As for my testing of invalid URL input, (like I tested in my test.py) any invalid URL inputted to my functions that accept a URL will return “invalid url” or sort, which will ensure that the user input url will be good to fetch some data from. I will not show this part because it is very obvious in my code.

Last but not least, my whole coding structure looks like this:



```
FA21-CS242-ASSIGNMENT2
  > __pycache__
  > .dist
  author_scraping.py
  book_scraping.py
  data.json            U
  database.py          M
  main.py              M
  README.md
  test.py
```