

# **Manual Test Plan**

## **Prerequisites:**

This project requires

- Python3
- PyCharm
- MongoDB

Make sure you have all of those installed before beginning.

This project was developed with PyCharm Professional, but you can run the command line without it.

## **Environment Setup and Configurations:**

```
git clone https://gitlab.engr.illinois.edu/yiyins2/sp21-cs242-assignment2.git  
cd sp21-cs242-assignment2
```

## **For the API and Command Line:**

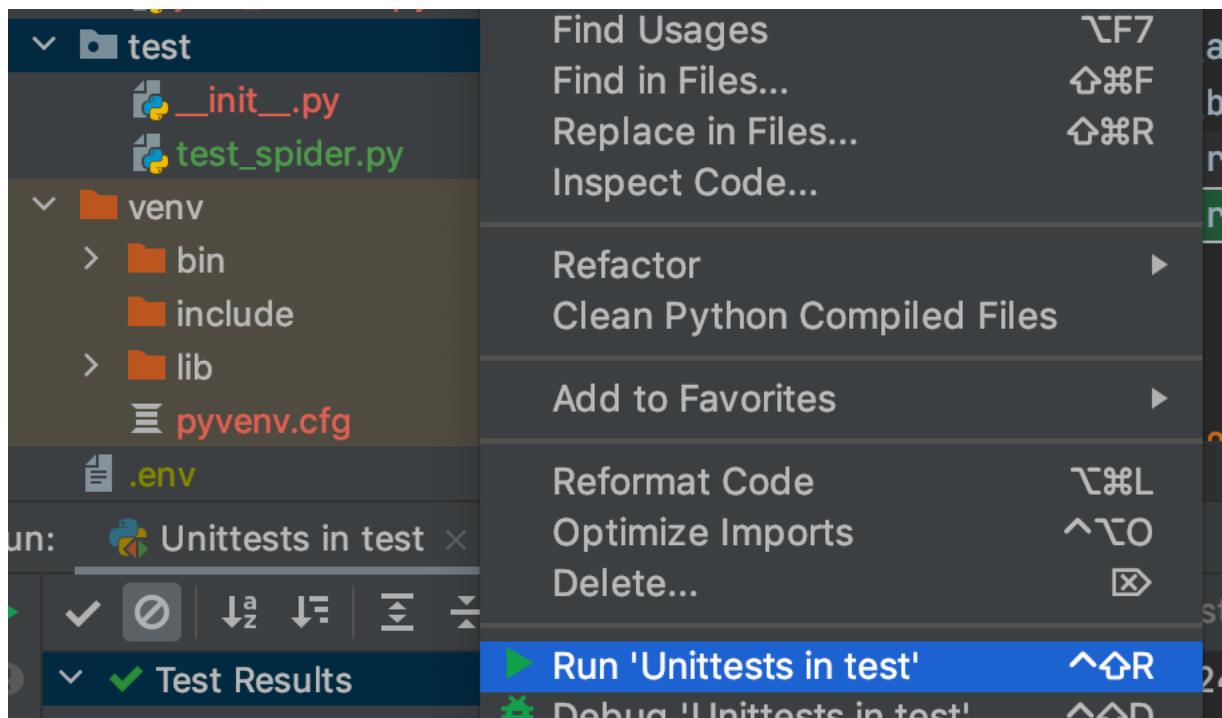
Before running the test and command line, you need to create a '.env' file that includes:

MONGODB\_KEY = your mongodb URI  
MONGODB\_DB = your database name

You also need to create two collections inside your database: one is “authors” and the other is “books” to store the scrapped authors and books.

For running the test cases:

Right click the “test” package and hit “Run ‘Unitests in test’”



For running the command line:

In the terminal, run

```
python goodreads/command_line.py
```

with the following flags

- `--start_url`: the URL to start scrapping (must be a goodreads book URL), default is <https://www.goodreads.com/book/show/3735293-clean-code>
- `--max_authors`: max num of authors to scrap (must be a positive int and better not exceed 50), default is 50
- `--max_books`: max num of books to scrap (must be a positive int and better not exceed 200), default is 200
- `--import_json`: import the given json into the database (must be an existing file and correct format), default is none
- `--export_json`: export the database into the given file (must be a valid path)

For running the API:

```
python goodreads/api.py
```

With default settings, the database will look like:

It scrapped 53 authors and 204 books.

## goodreads.authors

COLLECTION SIZE: 126.22KB TOTAL DOCUMENTS: 53 INDEXES TOTAL SIZE: 36KB

Find Indexes Schema Anti-Patterns 0 Aggregation Search Indexes •

INSERT DOCUMENT

FILTER {"filter": "example"} Find Reset

QUERY RESULTS 1-20 OF MANY

`_id: ObjectId("60466209d9f013abd87a4bb")  
author_url: "https://www.goodreads.com/author/show/1003655.K_W_Jeter"  
author_id: 1003655  
rating: 3.57  
rating_count: 27088  
review_count: 1734  
image_url: "https://images.gr-assets.com/authors/1363305061p5/1003655.jpg"  
> author_books: Array  
> similar_authors: Array`

`_id: ObjectId("60466209d9f013abd87a4bf9")  
author_url: "https://www.goodreads.com/author/show/4845.Kevin_J_Anderson"  
author_id: 4845  
rating: 3.77  
rating_count: 446239  
review_count: 17903  
image_url: "https://images.gr-assets.com/authors/1215310030p5/4845.jpg"  
> author_books: Array  
> similar_authors: Array`

`_id: ObjectId("60466278d9f013abd87a8b34")  
author_url: "https://www.goodreads.com/author/show/45372.Robert_C_Martin"  
author_id: 45372  
rating: 4.34  
rating_count: 29148  
review_count: 1016`

## goodreads.books

COLLECTION SIZE: 527.03KB TOTAL DOCUMENTS: 204 INDEXES TOTAL SIZE: 36KB

Find

Indexes

Schema Anti-Patterns 0

Aggregation

Search Indexes •

INSERT DOCUMENT

FILTER {"filter": "example"}

Find

Reset

QUERY RESULTS 1-20 OF MANY

```
_id: ObjectId("60466209d9f013abd87a4ba5")
book_url: "https://www.goodreads.com/book/show/9222475-infernal-devices"
title: "Infernal Devices"
book_id: 9222475
ISBN: "0857660977"
author_url: "https://www.goodreads.com/author/show/1003655.K_W_Jeter"
author: "K.W. Jeter"
rating: 3.36
rating_count: 2702
review_count: 386
image_url: "https://i.gr-assets.com/images/S/compressed.photo.goodreads.com/books/..."
> similar_books: Array
```

```
_id: ObjectId("60466209d9f013abd87a4bd7")
book_url: "https://www.goodreads.com/book/show/337116.The_Difference_Engine"
title: "The Difference Engine"
book_id: 337116
ISBN: "055329461"
author_url: "https://www.goodreads.com/author/show/9226.William_Gibson"
author: "William Gibson"
rating: 3.46
rating_count: 20384
review_count: 1117
image_url: "https://i.gr-assets.com/images/S/compressed.photo.goodreads.com/books/..."
```



We can also see that there are no duplicates in the database:

## goodreads.books

COLLECTION SIZE: 527.03KB TOTAL DOCUMENTS: 204 INDEXES TOTAL SIZE: 36KB

Find

Indexes

Schema Anti-Patterns 0

Aggregation

Search Indexes •

INSERT DOCUMENT

FILTER {"title": "Clean Code: A Handbook of Agile Software Craftsmanship"}

Find

Reset

QUERY RESULTS 1-1 OF 1

```
_id: ObjectId("60466277d9f013abd87a8acf")
book_url: "https://www.goodreads.com/book/show/3735293-clean-code"
title: "Clean Code: A Handbook of Agile Software Craftsmanship"
book_id: 3735293
ISBN: "0132350882"
author_url: "https://www.goodreads.com/author/show/45372.Robert_C_Martin"
author: "Robert C. Martin"
rating: 4.4
rating_count: 15967
review_count: 950
image_url: "https://i.gr-assets.com/images/S/compressed.photo.goodreads.com/books/..."
> similar_books: Array
```

The screenshot shows the MongoDB Compass interface with the 'goodreads.authors' collection selected. At the top, it displays 'COLLECTION SIZE: 126.22KB', 'TOTAL DOCUMENTS: 53', and 'INDEXES TOTAL SIZE: 36KB'. Below the header are navigation tabs: 'Find' (highlighted in green), 'Indexes', 'Schema Anti-Patterns (0)', 'Aggregation', and 'Search Indexes (1)'. On the right side of the header are buttons for 'INSERT DOCUMENT', 'Find' (green), and 'Reset'. A search bar at the top contains the filter '{ "author\_id": 45372 }'. The main area is titled 'QUERY RESULTS 1-1 OF 1' and shows a single document. The document details are as follows:

```

_id: ObjectId("60466278d9f013abd87a8b34")
author_url: "https://www.goodreads.com/author/show/45372.Robert_C_Martin"
author_id: 45372
rating: 4.34
rating_count: 29148
review_count: 1916
image_url: "https://images.gr-assets.com/authors/1490470967p5/45372.jpg"
> author_books: Array
> similar_authors: Array

```

On the right side of the document preview, there are four small icons for edit, copy, delete, and refresh.

## For the interactive front-end web page:

Make sure you are running the API:

```
python goodreads/api.py
```

To run the interactive web page, go to the “app” folder and run:

```
npm start
```

Then access the web page at <http://localhost:3000/>

## Manual Tests:

### For Command Line Testing:

#### Test start\_url:

With command “`python goodreads/command_line.py --start_url https://www.goodreads.com/book/show/9222475-infernal-devices --max_authors 1 --max_books 1`”, the database indeed updated with the book and the corresponding author and it stop scrapping after reaching the max\_authors and max\_books:

## goodreads.books

COLLECTION SIZE: 7.37KB TOTAL DOCUMENTS: 3 INDEXES TOTAL SIZE: 4KB

Find

Indexes

Schema Anti-Patterns 0

Aggregation

Search Indexes 0

INSERT DOCUMENT

FILTER {"filter": "example"}

Find Reset

QUERY RESULTS 1-3 OF 3

```
> _id: ObjectId("60466209d9f013abd87a4ba5")
  book_url: "https://www.goodreads.com/book/show/9222475-infernal-devices"
  title: "Infernal Devices"
  book_id: 9222475
  ISBN: "0857660977"
  author_url: "https://www.goodreads.com/author/show/1003655.K_W_Jeter"
  author: "K.W. Jeter"
  rating: 3.36
  rating_count: 2702
  review_count: 386
  image_url: "https://i.gr-assets.com/images/S/compressed.photo.goodreads.com/books/..."
  > similar_books: Array
```



```
_id: ObjectId("60466209d9f013abd87a4bd7")
book_url: "https://www.goodreads.com/book/show/337116.The_Difference_Engine"
title: "The Difference Engine"
book_id: 337116
ISBN: "055329461"
author_url: "https://www.goodreads.com/author/show/9226.William_Gibson"
author: "William Gibson"
rating: 3.46
rating_count: 20384
review_count: 1117
image_url: "https://i.gr-assets.com/images/S/compressed.photo.goodreads.com/books/..."
> similar_books: Array
```



## goodreads.authors

COLLECTION SIZE: 3.98KB TOTAL DOCUMENTS: 2 INDEXES TOTAL SIZE: 20KB

Find

Indexes

Schema Anti-Patterns 0

Aggregation

Search Indexes ●

INSERT DOCUMENT

FILTER {"filter": "example"}

Find

Reset

QUERY RESULTS 1-2 OF 2

▶ \_id: ObjectId("60466209d9f013abd87a4bba")  
author\_url: "https://www.goodreads.com/author/show/1003655.K\_W\_Jeter"  
author\_id: 1003655  
rating: 3.57  
rating\_count: 27088  
review\_count: 1734  
image\_url: "https://images.gr-assets.com/authors/1363305061p/1003655.jpg"  
> author\_books: Array  
> similar\_authors: Array



\_id: ObjectId("60466209d9f013abd87a4bf9")  
author\_url: "https://www.goodreads.com/author/show/4845.Kevin\_J\_Anderson"  
author\_id: 4845  
rating: 3.77  
rating\_count: 446239  
review\_count: 17903  
image\_url: "https://images.gr-assets.com/authors/1215310030p5/4845.jpg"  
> author\_books: Array  
> similar\_authors: Array

It will raise an error saying invalid URL if the URL is invalid:

```
(venv) Yiyins-MacBook-Pro-2:sp21-cs242-assignment2 suophia$ python goodreads/command_line.py --start_url https://illinois.edu/  
Traceback (most recent call last):  
  File "goodreads/command_line.py", line 21, in <module>  
    raise argparse.ArgumentError(url_arg, 'Invalid URL.')  
argparse.ArgumentError: argument --start_url: Invalid URL.
```

## Test max\_authors and max\_books:

Values are not int will not be accepted:

```
^C(venv) Yiyins-MacBook-Pro-2:sp21-cs242-assignment2 suophia$ python goodreads/command_line.py --max_authors asdbjwkefd  
usage: command_line.py [-h] [--start_url START_URL]  
                      [--max_authors MAX_AUTHORS] [--max_books MAX_BOOKS]  
                      [--import_json IMPORT_JSON] [--export_json EXPORT_JSON]  
command_line.py: error: argument --max_authors: invalid int value: 'asdbjwkefd'
```

```
(venv) Yiyins-MacBook-Pro-2:sp21-cs242-assignment2 suophia$ python goodreads/command_line.py --max_books asdbjwkefd  
usage: command_line.py [-h] [--start_url START_URL]  
                      [--max_authors MAX_AUTHORS] [--max_books MAX_BOOKS]  
                      [--import_json IMPORT_JSON] [--export_json EXPORT_JSON]  
command_line.py: error: argument --max_books: invalid int value: 'asdbjwkefd'
```

**Negative values are not accepted:**

```
(venv) Yiyins-MacBook-Pro-2:sp21-cs242-assignment2 suophia$ python goodreads/command_line.py --max_authors -20
Traceback (most recent call last):
  File "goodreads/command_line.py", line 24, in <module>
    raise argparse.ArgumentError(max_authors_arg, 'Max number of authors cannot be negative.')
argparse.ArgumentError: argument --max_authors: Max number of authors cannot be negative.
```

```
(venv) Yiyins-MacBook-Pro-2:sp21-cs242-assignment2 suophia$ python goodreads/command_line.py --max_books -20
Traceback (most recent call last):
  File "goodreads/command_line.py", line 30, in <module>
    raise argparse.ArgumentError(max_books_arg, 'Max number of books cannot be negative.')
argparse.ArgumentError: argument --max_books: Max number of books cannot be negative.
```

**When max\_books exceeds 200, a warning will be raised:**

```
(venv) Yiyins-MacBook-Pro-2:sp21-cs242-assignment2 suophia$ python goodreads/command_line.py --max_books 2000
goodreads/command_line.py:33: UserWarning: Max number of books is too large (> 200).
  warnings.warn('Max number of books is too large (> 200).')
```

**When max\_authors exceeds 50, a warning will be raised:**

```
^C(venv) Yiyins-MacBook-Pro-2:sp21-cs242-assignment2 suophia$ python goodreads/command_line.py --max_authors 2000
goodreads/command_line.py:27: UserWarning: Max number of authors is too large (> 50).
  warnings.warn('Max number of authors is too large (> 50).')
```

## Test export json:

Currently we have three documents in “books”:

## goodreads.books

COLLECTION SIZE: 7.5KB TOTAL DOCUMENTS: 3 INDEXES TOTAL SIZE: 36KB

Find Indexes Schema Anti-Patterns 0 Aggregation Search Indexes •

INSERT DOCUMENT

FILTER {"filter": "example"} Find Reset

QUERY RESULTS 1-3 OF 3

\_id: ObjectId("604631b0d9f013abd85c4a00")  
book\_url: "https://www.goodreads.com/book/show/7716140-married-with-zombies"  
title: "Married with Zombies"  
book\_id: 7716140  
ISBN: "0316102865"  
author\_url: "https://www.goodreads.com/author/show/3369676.Jesse\_Petersen"  
author: "Jesse Petersen"  
rating: 3.74  
rating\_count: 6472  
review\_count: 993  
image\_url: "https://i.gr-assets.com/images/S/compressed.photo.goodreads.com/books/..."  
> similar\_books: Array

\_id: ObjectId("604631b1d9f013abd85c4a38")  
book\_url: "https://www.goodreads.com/book/show/52899703-always-be-my-banshee"  
title: "Always Be My Banshee"  
book\_id: 52899703  
ISBN: ""  
author\_url: "https://www.goodreads.com/author/show/2503028.Molly\_Harper"  
author: "Molly Harper"  
rating: 4.23  
rating\_count: 1540  
review\_count: 160  
image\_url: "https://i.gr-assets.com/images/S/compressed.photo.goodreads.com/books/..."  
> similar\_books: Array

id: ObjectId("604631b1d9f013abd85c4a56")

And two documents in “authors”:

## goodreads.authors

COLLECTION SIZE: 3.54KB TOTAL DOCUMENTS: 2 INDEXES TOTAL SIZE: 24KB

Find

Indexes

Schema Anti-Patterns 0

Aggregation

Search Indexes •

INSERT DOCUMENT

FILTER {"filter": "example"}

Find Reset

QUERY RESULTS 1-2 OF 2

```
> _id: ObjectId("604631b0d9f013abd85c4a1f")
author_url: "https://www.goodreads.com/author/show/3369676.Jesse_Petersen"
author_id: 3369676
rating: 3.84
rating_count: 13740
review_count: 1794
image_url: "https://images.gr-assets.com/authors/1282232065p5/3369676.jpg"
> author_books: Array
> similar_authors: Array
```



```
_id: ObjectId("604631b1d9f013abd85c4a75")
author_url: "https://www.goodreads.com/author/show/298438.Kerstin_Gier"
author_id: 298438
rating: 4.1
rating_count: 360001
review_count: 28624
image_url: "https://images.gr-assets.com/authors/1276520835p5/298438.jpg"
> author_books: Array
> similar_authors: Array
```

With command “`python goodreads/command_line.py --export_json test.json`”, we indeed got three books and two authors in “`test.json`”:

```
"books": [
  {
    "_id": {"$oid": "604631b0d9f013abd85c4a00"},  

    "book_url": "https://www.goodreads.com/book/show/7716140-married-with-zombies",  

    "title": "Married with Zombies",  

    "book_id": 7716140,  

    "ISBN": "0316102865",  

    "author_url": "https://www.goodreads.com/author/show/3369676.Jesse_Petersen",  

    "author": "Jesse Petersen",  

    "rating": 3.74,  

    "rating_count": 6472,  

    "review_count": 993,  

    "image_url": "https://i.gr-assets.com/images/S/compressed.photo.goodreads.com/books/1265814659L/7716140.jpg",  

    "similar_books": [...]
  },
  {
    "_id": {"$oid": "604631b1d9f013abd85c4a38"},  

    "book_url": "https://www.goodreads.com/book/show/52899703-always-be-my-banshee",  

    "title": "Always Be My Banshee",  

    "book_id": 52899703,  

    "ISBN": "",  

    "author_url": "https://www.goodreads.com/author/show/2503028.Molly_Harper",  

    "author": "Molly Harper",  

    "rating": 4.23,  

    "rating_count": 1540,  

    "review_count": 160,  

    "image_url": "https://i.gr-assets.com/images/S/compressed.photo.goodreads.com/books/1585868410L/52899703._SX318_.jpg",  

    "similar_books": [...]
  },
  {
    "_id": {"$oid": "604631b1d9f013abd85c4a56"},  

    "book_url": "https://www.goodreads.com/book/show/29902146-jed-had-to-die",  

    "title": "Jed Had to Die",  

    "book_id": 29902146,  

    "ISBN": "1682304329",  

    "author_url": "https://www.goodreads.com/author/show/6422281.Tara_Sivec",  

    "author": "Tara Sivec",
  }
]
```

```
"authors": [
  {
    "_id": {"$oid": "604631b0d9f013abd85c4a1f"},  

    "author_url": "https://www.goodreads.com/author/show/3369676.Jesse_Petersen",  

    "author_id": 3369676,  

    "rating": 3.84,  

    "rating_count": 13740,  

    "review_count": 1794,  

    "image_url": "https://images.gr-assets.com/authors/1282232065p5/3369676.jpg",  

    "author_books": [...],  

    "similar_authors": [...]
  },
  {
    "_id": {"$oid": "604631b1d9f013abd85c4a75"},  

    "author_url": "https://www.goodreads.com/author/show/298438.Kerstin_Gier",  

    "author_id": 298438,  

    "rating": 4.1,  

    "rating_count": 360001,  

    "review_count": 28624,  

    "image_url": "https://images.gr-assets.com/authors/1276520835p5/298438.jpg",  

    "author_books": [...],  

    "similar_authors": [...]
  }
]
```

If we start with an empty database, the json file returned contains no books and authors:

```
> sophia > cs > 242 > s|  
{|  
  "books": [],  
  "authors": []  
}|
```

## Test import json:

We first test importing a file that does not exist, it raises an argument error saying json file does not exist.

```
(venv) Yiyins-MacBook-Pro-2:sp21-cs242-assignment2 suophia$ python goodreads/command_line.py --import_json xx.json
Traceback (most recent call last):
  File "goodreads/command_line.py", line 36, in <module>
    raise argparse.ArgumentError(import_json_arg, 'JSON file does not exist.')
argparse.ArgumentError: argument --import_json: JSON file does not exist.
```

We then test import an empty json, and indeed nothing changes in the database:

The screenshot shows the MongoDB Compass interface with two collection panes. The top pane is for the collection **goodreads.books**. It displays the following information:

- Collection Size: 0B
- Total Documents: 0
- Indexes Total Size: 12KB

Below this, there are tabs for **Find**, **Indexes**, **Schema Anti-Patterns** (0), **Aggregation**, and **Search Indexes**. A **FILTER** button with the query `{"filter": "example"}` is present. The bottom section is labeled **QUERY RESULTS 0**. The bottom pane is for the collection **goodreads.authors**, showing similar statistics and a matching filter query. Both panes have a **QUERY RESULTS 0** message at the bottom.

We then test import a json with the wrong format, without book\_id:

```
{  
  "books": [  
    {  
      "book_url": "https://www.goodreads.com/book/show/7716140-married-with-zombies",  
      "title": "Married with Zombies"  
    }  
  ],  
  "authors": [  
    {  
      "author_url": "https://www.goodreads.com/author/show/298438.Kerstin_Gier",  
      "rating": 4.1,  
      "rating_count": 360001  
    }  
  ]  
}
```

It raises an error saying no book\_id:

```
(venv) Yiyins-MacBook-Pro-2:sp21-cs242-assignment2 suophia$ python goodreads/command_line.py --import_json wrong.json  
Traceback (most recent call last):  
  File "goodreads/command_line.py", line 39, in <module>  
    import_json(args.import_json)  
  File "/Users/suophia/cs/242/sp21-cs242-assignment2/venv/lib/python3.7/site-packages/goodreads/json_handler.py", line 21, in import_json  
    raise ValueError('No book id.')  
ValueError: No book id.
```

Similarly, an error will be raised if no author\_id:

```
(venv) Yiyins-MacBook-Pro-2:sp21-cs242-assignment2 suophia$ python goodreads/command_line.py --import_json wrong.json  
Traceback (most recent call last):  
  File "goodreads/command_line.py", line 39, in <module>  
    import_json(args.import_json)  
  File "/Users/suophia/cs/242/sp21-cs242-assignment2/venv/lib/python3.7/site-packages/goodreads/json_handler.py", line 25, in import_json  
    db['authors'].replace_one({'author_id': author['author_id']}, author, upsert=True)  
KeyError: 'author_id'
```

Now, we test a correct formatted nonempty json (“test.json” we exported when testing exporting) into an empty database:

We can see that the three documents and two authors in “test.json” successfully imported into the database:

## goodreads.books

COLLECTION SIZE: 7.5KB TOTAL DOCUMENTS: 3 INDEXES TOTAL SIZE: 24KB

Find

Indexes

Schema Anti-Patterns 0

Aggregation

Search Indexes •

INSERT DOCUMENT

FILTER {"filter": "example"}

Find

Reset

QUERY RESULTS 1-3 OF 3

```
> _id: ObjectId("604631b1d9f013abd85c4a56")
book_url: "https://www.goodreads.com/book/show/29902146-jed-had-to-die"
title: "Jed Had to Die"
book_id: 29902146
ISBN: "1682304329"
author_url: "https://www.goodreads.com/author/show/6422281.Tara_Sivec"
author: "Tara Sivec"
rating: 4.17
rating_count: 4083
review_count: 608
image_url: "https://i.gr-assets.com/images/S/compressed.photo.goodreads.com/books/..."
> similar_books: Array
```



```
_id: ObjectId("604631b0d9f013abd85c4a00")
book_url: "https://www.goodreads.com/book/show/7716140-married-with-zombies"
title: "Married with Zombies"
book_id: 7716140
ISBN: "0316102865"
author_url: "https://www.goodreads.com/author/show/3369676.Jesse_Petersen"
author: "Jesse Petersen"
rating: 3.74
rating_count: 6472
review_count: 993
image_url: "https://i.gr-assets.com/images/S/compressed.photo.goodreads.com/books/..."
> similar_books: Array
```



```
_id: ObjectId("604631b1d9f013abd85c4a38")
```

**goodreads.authors**

COLLECTION SIZE: 3.54KB TOTAL DOCUMENTS: 2 INDEXES TOTAL SIZE: 24KB

**Find** Indexes Schema Anti-Patterns 0 Aggregation Search Indexes • INSERT DOCUMENT

FILTER {"filter": "example"} Find Reset

QUERY RESULTS 1-2 OF 2

```
_id: ObjectId("604631b0d9f013abd85c4a1f")
author_url: "https://www.goodreads.com/author/show/3369676.Jesse_Petersen"
author_id: 3369676
rating: 3.84
rating_count: 13740
review_count: 1794
image_url: "https://images.gr-assets.com/authors/1282232065p5/3369676.jpg"
> author_books: Array
> similar_authors: Array
```

```
_id: ObjectId("604631b1d9f013abd85c4a75")
author_url: "https://www.goodreads.com/author/show/298438.Kerstin_Gier"
author_id: 298438
rating: 4.1
rating_count: 360001
review_count: 28624
image_url: "https://images.gr-assets.com/authors/1276520835p5/298438.jpg"
> author_books: Array
> similar_authors: Array
```

If we import “test.json” again, the database stays the same (no duplicates are made).

## For Interactive Web Page Testing:

When you enter the web page, there are three links:



Table will lead users to a table of books and a table of authors, where users can perform GET, PUT, POST, and DELETE operations.

Authors Visualization will lead users to a bar plot of user-given top k authors' rating.

Books Visualization will lead users to a bar plot of user-given top k books' rating.

When you go to the Table's page, a list of all books in the database will be shown:

React App    localhost:3000/api

- Table
- Authors Visualization
- Books Visualization

## Books

Search Books

Book ID	Title	Book URL
ISBN	Image URL	
Author	Author URL	
Rating	Rating Count	Review Count
Add Book		

	Book ID	Title	ISBN	Author	Image	Rating	Rating Count	Review Count	Similar Books
<a href="#">Delete</a>	18043011	Clean Architecture	0134494164	Robert C. Martin		4.24	33811	2742	<a href="#">Click to see similar books</a>
<a href="#">Delete</a>	84985	Agile Software Development, Principles, Patterns, and Practices	0135974445	Robert C. Martin		4.25	1192	49	<a href="#">Click to see similar books</a>
<a href="#">Delete</a>	84983	Agile Principles, Patterns, and Practices	0131857258	Robert C. Martin		4.3	713	34	<a href="#">Click to see similar books</a>

And a list of all authors in the database will be shown:

React App    localhost:3000/api

## Authors

Search Authors

Author ID	Name	Author URL	Image URL
Rating	Rating Count	Review Count	
Add Author			

	Author ID	Name	Image	Rating	Rating Count	Review Count	Author's Books	Similar Authors
<a href="#">Delete</a>	3174788	Frederick P. Brooks Jr.		4.03	12273	839	<a href="#">Click to see author's books</a>	<a href="#">Click to see similar authors</a>
<a href="#">Delete</a>	25215	Martin Fowler		4.12	15843	722	<a href="#">Click to see author's books</a>	<a href="#">Click to see similar authors</a>
<a href="#">Delete</a>	48622	Erich Gamma		4.2	16722	612	<a href="#">Click to see author's books</a>	<a href="#">Click to see similar authors</a>
<a href="#">Delete</a>	153350	Brian W. Kernighan		4.3	16936	862	<a href="#">Click to see author's books</a>	<a href="#">Click to see similar authors</a>
<a href="#">Delete</a>	7969625	Martin Kleppmann		4.7	4088	442	<a href="#">Click to see author's books</a>	<a href="#">Click to see similar authors</a>
<a href="#">Delete</a>	14019088	John Ousterhout		4.14	1357	199	<a href="#">Click to see author's books</a>	<a href="#">Click to see similar authors</a>

When you click on “click to see similar books”, a list of similar books and their link will be shown. You can click once again to close.

The screenshot shows a web browser window with the title "React App" and the URL "localhost:3000/api". At the top, there is a search bar labeled "Add Book" and a button labeled "Update". Below the search bar is a table with columns: Book ID, Title, ISBN, Author, Image, Rating, Rating Count, Review Count, and Similar Books. A row for the book "Clean Architecture" by Robert C. Martin is selected. The "Similar Books" column for this row contains a blue box with the following content:

Click to see similar books

- Refactoring: Improving the Design of Existing Code
- The Pragmatic Programmer: From Journeyman to Master
- Test-Driven Development: By Example
- Domain-Driven Design: Tackling Complexity in the Heart of Software
- Design Patterns: Elements of Reusable Object-Oriented Software
- Building Microservices: Designing Fine-Grained Systems
- Patterns of Enterprise Application Architecture
- Designing

Similarly, you can click on “click to see author’s books” and “click to see similar authors” to see a list of author’s books and a list of similar authors.

Search		Add Author						
	Author ID	Name	Image	Rating	Rating Count	Review Count	Author's Books	Similar Authors
<a href="#">Delete</a>	63853	James L. Adams		3.74	648	48	Click to see author's books <ul style="list-style-type: none"> <li>• Bir Mühendisin Dünyası</li> <li>• Good Products, Bad Products: Essential Elements to Achieving Superior Quality</li> <li>• The Care and Feeding of Ideas: A Guide to Encouraging Creativity</li> <li>• Conceptual Blockbusting: A Pleasurable Guide to Better Problem Solving</li> <li>• The Building of an Engineer: Making, Teaching, and Thinking</li> <li>• A Full Service Bank: How BCCI Stole Billions Around The World</li> <li>• Mentaru Burokkubasutā =Mental Blockbuster: Jiyūna Hasso O Samatageru Gtsu No Kabe O Buchiyabure</li> <li>• Whippoorwills Only Sing at Night</li> <li>• Sblocca il cervello</li> </ul>	Click to see similar authors <ul style="list-style-type: none"> <li>• Anne McCaffrey</li> <li>• Steve McConnell</li> <li>• Mary Roach</li> <li>• Henry Petroski</li> <li>• Dean Karnazes</li> <li>• David Ruellé</li> <li>• Klaus Schwab</li> <li>• Sue Henger</li> <li>• Chris Woodard</li> <li>• George Polya</li> <li>• Ashlee Vance</li> <li>• Rumi</li> <li>• Yusuf Atılgan</li> <li>• Cahit Külebi</li> <li>• İber Ortaylı</li> <li>• Edward D. Hess</li> <li>• John E. Doerr</li> <li>• Holly Madison</li> <li>• Scott Jurek</li> <li>• Gérard Dorel</li> <li>• Bernadette Jiwa</li> <li>• Hector García Puigcerver</li> <li>• Caitlin Doughty</li> <li>• OWP/P Architects</li> <li>• Meik Wiking</li> <li>• Jennifer Le Zotte</li> </ul>
<a href="#">Delete</a>	14909	Dino Esposito		3.9	662	37	Click to see author's books	Click to see similar authors

## Test Get:

The user can choose the field they want to search for and enter the query. If the field and query is valid, the searched book will be shown in the table, and a success message box will be shown on the top right.

React App

localhost:3000/api

- Table
- Authors Visualization
- Books Visualization

Books

Search Books

Query

Clean Architecture

Search

Book ID Title Book URL

ISBN Image URL

Author Author URL

Rating Rating Count Review Count

Add Book

	Book ID	Title	ISBN	Author	Image	Rating	Rating Count	Review Count	Similar Books
<button>Delete</button>	18043011	Clean Architecture	0134494164	Robert C. Martin		4.24	33811	2742	Click to see similar books
<button>Edit</button>									

10 ▾ 1

Authors

Search Authors

Author ID	Name	Author URL	Image URL

Successfully searched book!

If the field or query is invalid, a warning message box will be shown on the top right.

React App

localhost:3000/api

- Table
- Authors Visualization
- Books Visualization

Books

Search Books

Query

xxx

Search

Book ID Title Book URL

ISBN Image URL

Author Author URL

Rating Rating Count Review Count

Add Book

	Book ID	Title	ISBN	Author	Image	Rating	Rating Count	Review Count	Similar Books
<button>Delete</button>	18043011	Clean Architecture	0134494164	Robert C. Martin		4.24	33811	2742	Click to see similar books
<button>Edit</button>									

10 ▾ 1

Authors

Search Authors

Author ID	Name	Author URL	Image URL

Bad Request: Invalid query

If the search result is none, no book will be shown in the table.

The screenshot shows a web application titled "React App" running at "localhost:3000/api". The main menu includes "Table", "Authors Visualization", and "Books Visualization".

**Books Section:**

- Search Books:** A dropdown menu set to "Rating" with a query input field containing "< 1" and a "Search" button.
- Table Headers:** Book ID, Title, Book URL.
- Table Data:** An empty table row with fields for ISBN, Image URL, Author, Author URL, Rating, Rating Count, and Review Count.
- Buttons:** "Add Book" and a "Similar Books" link.
- Table Footer:** A dropdown menu set to "10".

**Authors Section:**

- Search Authors:** A dropdown menu set to "ID" with a query input field and a "Search" button.
- Table Headers:** Author ID, Name, Author URL, Image URL.
- Table Data:** An empty table row with fields for Rating, Rating Count, and Review Count.
- Buttons:** "Add Author" and a "Similar Books" link.

Similarly, for authors' searching, on success, a message box will pop up and the author will be shown in the table.

		Object-Oriented Software									
<a href="#">Delete</a>	44936	Refactoring: Improving the Design of Existing Code	0201485672	Martin Fowler		4.25	7159	294	<a href="#">Click to see similar books</a>		<div>Successfully searched author!</div>
<a href="#">Delete</a>	39832183	The Guernsey Literary and Potato Peel Pie Society	1984801813	Mary Ann Shaffer		4.17	636414	55322	<a href="#">Click to see similar books</a>		

## Authors

Search Authors	Author ID	Name	Author URL	Image URL
Rating	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
Query	Rating	Rating Count	Review Count	
<input type="text" value=" &gt; 4.5"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	
<input type="button" value="Search"/>	<input type="button" value="Add Author"/>			

	Author ID	Name	Image	Rating	Rating Count	Review Count	Author's Books	Similar Authors
Delete	7969625	Martin Kleppmann		4.7	4088	442	Click to see author's books	Click to see similar authors
Edit								

10 ▾

1

A warning message box will pop out for invalid query.

The screenshot shows a browser window titled "React App" at the URL "localhost:3000/api". At the top right, there are several icons: a magnifying glass, a star, a green circle, a puzzle piece, a red circle with a white "Y", and an "Update" button. Below the address bar is a search input field with the placeholder "Add Book". Underneath the search field is a table with the following columns: Book ID, Title, ISBN, Author, Image, Rating, Rating Count, Review Count, and Similar Books. The table has one row of data. At the bottom left, there is a dropdown menu set to "10". A yellow callout box on the right side of the screen displays the error message "Bad Request: Invalid query".

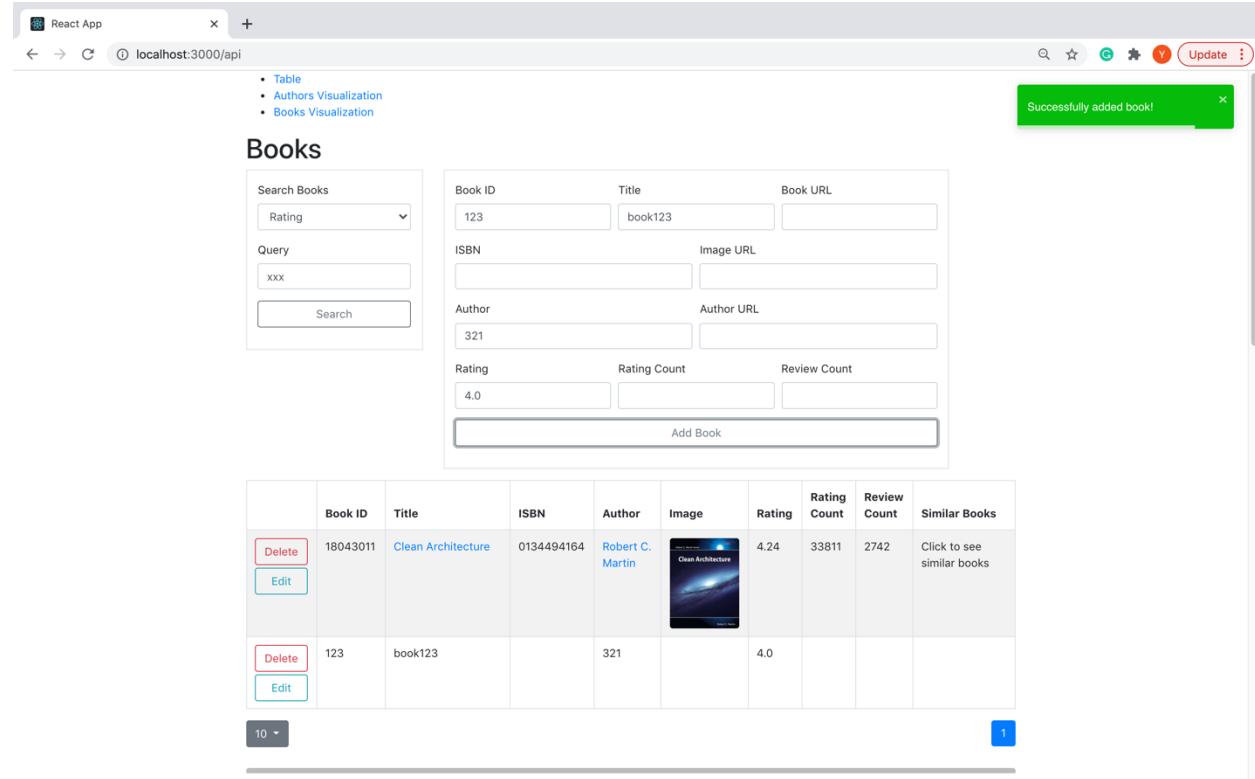
1

## Authors

Search Authors	Author ID	Name	Author URL	Image URL
<input type="button" value="Rating Count"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
Query	Rating	Rating Count	Review Count	
<input type="text" value="-1"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	
<input type="button" value="Search"/>	<input type="button" value="Add Author"/>			

# Test POST:

Users can also add books. If the book is in correct format, a success message box will pop out, and the newly added book will be shown at the end of the table.



The screenshot shows a web application interface for managing books. At the top, there's a navigation bar with links for 'Table', 'Authors Visualization', and 'Books Visualization'. A green success message box in the top right corner says 'Successfully added book!'. Below the navigation, there's a search form with dropdowns for 'Rating' (set to 'Rating') and 'Query' (set to 'xxx'), and a 'Search' button. To the right of the search form is a detailed book input form with fields for Book ID (123), Title (book123), Book URL (empty), ISBN (empty), Image URL (empty), Author (321), Author URL (empty), Rating (4.0), Rating Count (empty), Review Count (empty), and an 'Add Book' button. Below these forms is a table displaying book data. The table has columns: Book ID, Title, ISBN, Author, Image, Rating, Rating Count, Review Count, and Similar Books. It contains two rows. The first row corresponds to the book just added, showing '18043011' as the Book ID, 'Clean Architecture' as the Title, '0134494164' as the ISBN, 'Robert C. Martin' as the Author, an image thumbnail of the book cover, a rating of '4.24', 33811 reviews, 2742 ratings, and a link 'Click to see similar books'. The second row shows '123' as the Book ID, 'book123' as the Title, an empty ISBN field, '321' as the Author, an empty image field, a rating of '4.0', an empty rating count field, an empty review count field, and an empty similar books section. At the bottom of the table are pagination controls showing '10' and '1'.

	Book ID	Title	ISBN	Author	Image	Rating	Rating Count	Review Count	Similar Books
<a href="#">Delete</a> <a href="#">Edit</a>	18043011	Clean Architecture	0134494164	Robert C. Martin		4.24	33811	2742	<a href="#">Click to see similar books</a>
<a href="#">Delete</a> <a href="#">Edit</a>	123	book123		321		4.0			

If format of the book is invalid, a warning message box will be shown:

React App    localhost:3000/api

- Table
- Authors Visualization
- Books Visualization

**Books**

Search Books

ID	Book ID	Title	Book URL
-1			

ISBN	Image URL

Author	Author URL

Rating	Rating Count	Review Count
-1		

Add Book

	Book ID	Title	ISBN	Author	Image	Rating	Rating Count	Review Count	Similar Books
<a href="#">Delete</a>	18043011	Clean Architecture	0134494164	Robert C. Martin		4.24	33811	2742	Click to see similar books
<a href="#">Delete</a>	84985	Agile Software Development, Principles, Patterns, and Practices	0135974445	Robert C. Martin		4.25	1192	49	Click to see similar books
<a href="#">Delete</a>	84983	Agile Principles, Patterns, and	0131857258	Robert C. Martin		4.3	713	34	Click to see similar books

Similarly, for adding invalid-format author, new book will be shown at the end of the table.

React App    Data | Atlas: MongoDB Atlas    localhost:3000/api

	Professional Programmers						
--	--------------------------	--	--	--	--	--	--

10 ▾ 1 2 >

**Authors**

Search Authors

ID	Author ID	Name	Author URL	Image URL
123	123	author		

Rating	Rating Count	Review Count
	1	

Add Author

	Author ID	Name	Image	Rating	Rating Count	Review Count	Author's Books	Similar Authors
<a href="#">Delete</a>	63853	James L. Adams		3.74	648	48	Click to see author's books	Click to see similar authors
<a href="#">Delete</a>	14909	Dino Esposito		3.94	662	37	Click to see author's books	Click to see similar authors
<a href="#">Delete</a>	123	author		1				

10 ▾ 1 2 < >

A warning message box will be shown for invalid author format.

The screenshot shows a web application interface. At the top, there is a header bar with a 'React App' logo, a search icon, a refresh icon, and a URL 'localhost:3000/api'. To the right of the URL are icons for search, star, GitHub, and more, followed by a red 'Update' button and a three-dot menu. Below the header is a table with book details. The first row contains: a 'Delete' button, ID '10284614', title 'The Clean Coder: A Code of Conduct for Professional Programmers', author 'Robert C. Martin', a thumbnail image of the book cover, rating '4.28', reviews '6684', and a link 'Click to see similar books'. To the right of the table is a yellow warning box with the text 'Invalid format' and a close 'x' button. Below the table is a page navigation bar with '10 < > 1 2' and a '10' dropdown. The main content area is titled 'Authors' and contains a search form with fields for 'ID' (dropdown), 'Query' (text input), and a 'Search' button. It also has sections for 'Author ID', 'Name', 'Author URL', 'Image URL', 'Rating', 'Rating Count', and 'Review Count', along with a 'Rating' input field and two empty input fields for 'Rating Count' and 'Review Count'. Below this is a large 'Add Author' button. The bottom part of the screenshot shows a table of authors with columns: Author ID, Name, Image, Rating, Rating Count, Review Count, Author's Books, and Similar Authors. Three rows are listed:

	Author ID	Name	Image	Rating	Rating Count	Review Count	Author's Books	Similar Authors
<a href="#">Delete</a>	3174788	Frederick P. Brooks Jr.		4.03	12273	839	Click to see author's books	Click to see similar authors
<a href="#">Delete</a>	25215	Martin Fowler		4.12	15843	722	Click to see author's books	Click to see similar authors
<a href="#">Delete</a>	48622	Erich Gamma		4.2	16722	612	Click to see author's books	Click to see similar authors

## Test PUT:

If the user clicked on the edit button, a pop-up window will be shown with the original value of the book. If the user clicked cancel, no changes will be made.

React App

localhost:3000/api

Title: Clean Architecture 1

Book URL: https://www.goodreads.com/book/show/18043011-clean-architecture

ISBN: 0134494164

Image URL: https://i.gr-assets.com/images/S/compressed.photo.goodreads.com/books/1471680093/18043011.\_SX31

Author: Robert C. Martin

Author URL: https://www.goodreads.com/author/show/45372.Robert\_C\_Martin

Rating: 3

Rating Count: 33811

Review Count: 2742

[Edit](#) [Cancel](#)

On success, the book edited will be refreshed.

React App

localhost:3000/api

- Table
- Authors Visualization
- Books Visualization

## Books

	Book ID	Title	ISBN	Author	Image	Rating	Rating Count	Review Count	Similar Books
<a href="#">Delete</a>	18043011	Clean Architecture 1	0134494164	Robert C. Martin		3	33811	2742	<a href="#">Click to see similar books</a>
<a href="#">Delete</a>	84985	Agile Software Development, Principles, Patterns, and Practices	0135974445	Robert C. Martin		4.25	1192	49	<a href="#">Click to see similar books</a>
<a href="#">Delete</a>	84983	Agile Principles, Patterns, and Practices	0131857258	Robert C. Martin		4.3	713	34	<a href="#">Click to see similar books</a>

React App

localhost:3000/api

Authors Visualization

Title: Clean Architecture 1 Book URL: https://www.goodreads.com/book/show/18043011-clean-architecture

ISBN: 0134494164 Image URL: https://i.gr-assets.com/images/S/compressed.photo.goodreads.com/books/1471680093/18043011.\_SX31

Author: Robert C. Martin Author URL: https://www.goodreads.com/author/show/45372.Robert\_C\_Martin

Rating: -1 Rating Count: 33811 Review Count: 2742

[Edit](#) [Cancel](#)

If the user input invalid value, like negative rating, a warning message box will pop up.

React App

localhost:3000/api

Books Visualization

Books

Search Books

ID:

Query:

[Search](#)

	Book ID	Title	ISBN	Author	Image	Rating	Rating Count	Review Count	Similar Books
<a href="#">Delete</a>	18043011	Clean Architecture 1	0134494164	Robert C. Martin		3	33811	2742	<a href="#">Click to see similar books</a>
<a href="#">Delete</a>	84985	Agile Software Development, Principles, Patterns, and Practices	0135974445	Robert C. Martin		4.25	1192	49	<a href="#">Click to see similar books</a>
<a href="#">Delete</a>	84983	Agile Principles, Patterns, and Practices in C#	0131857258	Robert C. Martin		4.3	713	34	<a href="#">Click to see similar books</a>

[Add Book](#)

Invalid input!

Similarly, the user can also edit author. On success, a success message box will pop up and the author edited will be refreshed.

A screenshot of a modal dialog box titled "Name Author URL Image URL". It contains fields for Name (Frederick P. Brooks Jr.), Author URL (https://www.goodreads.com/author/show/3174788.Frederick\_P\_Brook), and Image URL. Below these are fields for Rating (4), Rating Count (1227), and Review Count (839). At the bottom are "Edit" and "Cancel" buttons.

A screenshot of a table view showing a single book entry. The columns include ID (10284614), Title (The Clean Coder: A Code of Conduct for Professional Programmers), Author (Robert C. Martin), Image (Thumbnail of the book cover), Rating (4.28), Rating Count (6684), and Review Count (495). There is a link "Click to see similar books". Navigation buttons at the bottom show page 1 of 2.

## Authors

A screenshot of a search form for authors. It includes fields for "Search Authors" (ID dropdown, Query input, Search button), "Author ID", "Name", "Author URL", and "Image URL". Below these are fields for "Rating", "Rating Count", and "Review Count". A "Add Author" button is also present.

A screenshot of a table view showing multiple author entries. The columns include Author ID, Name, Image, Rating, Rating Count, Review Count, Author's Books, and Similar Authors. Each row shows an author with their details, a thumbnail image, and links to see their books and similar authors. The authors listed are Frederick P. Brooks Jr., Martin Fowler, Erich Gamma, and Brian W. Kernighan.

	Author ID	Name	Image	Rating	Rating Count	Review Count	Author's Books	Similar Authors
<a href="#">Delete</a>	3174788	Frederick P. Brooks Jr.		4	1227	839	Click to see author's books	Click to see similar authors
<a href="#">Delete</a>	25215	Martin Fowler		4.12	15843	722	Click to see author's books	Click to see similar authors
<a href="#">Delete</a>	48622	Erich Gamma		4	1672	612	Click to see author's books	Click to see similar authors
<a href="#">Delete</a>	153350	Brian W. Kernighan		4.3	16936	862	Click to see author's books	Click to see similar authors

A warning message box will pop up if invalid input:

The screenshot shows a browser window with two tabs: "React App" and "Data | Atlas: MongoDB Atlas". The URL is "localhost:3000/api". The main content is an edit form for an author. It has fields for Name, Author URL, Image URL, Rating, Rating Count, and Review Count. The "Rating" field contains "-100". A red warning box labeled "Invalid input!" is positioned above the "Review Count" field.

Name	Author URL	Image URL
Frederick P. Brooks Jr.	<a href="https://www.goodreads.com/author/show/3174788.Frederick_P_Brook">https://www.goodreads.com/author/show/3174788.Frederick_P_Brook</a>	

Rating	Rating Count	Review Count
4	-100	839

**Edit** **Cancel**

The screenshot shows a browser window with two tabs: "React App" and "Data | Atlas: MongoDB Atlas". The URL is "localhost:3000/api". The main content includes a search interface with fields for Query, Rating, Rating Count, and Review Count, and a button to Add Author. A red warning box labeled "Invalid input!" is positioned above the "Review Count" field. Below is a table of authors:

ID	Author ID	Name	Image	Rating	Rating Count	Review Count	Author's Books	Similar Authors
	3174788	Frederick P. Brooks Jr.		4	1227	839	Click to see author's books	Click to see similar authors
	25215	Martin Fowler		4.12	15843	722	Click to see author's books	Click to see similar authors
	48622	Erich Gamma		4	1672	612	Click to see author's books	Click to see similar authors
	153350	Brian W. Kernighan		4.3	16936	862	Click to see author's books	Click to see similar authors
	7969625	Martin Kleppmann		4.7	4088	442	Click to see author's books	Click to see similar authors
	14019088	John Ousterhout		4.14	1357	199	Click to see author's books	Click to see similar authors
	6089	Michael T. Nygard		4.25	2692	216	Click to see author's books	Click to see similar authors

# Test Delete:

Users can delete a book by clicking the delete button. Then a confirmation box will pop up.

The screenshot shows a web browser window with the URL `localhost:3000/api`. On the left, there's a sidebar with links: 'Table', 'Authors Visualization', and 'Books Visualization'. The main area has a title 'Books' and a search form with fields for 'ID' (dropdown), 'Query' (text input), and a 'Search' button. To the right, a modal dialog box is open with the message 'localhost:3000 says' and 'Want to delete?'. It contains two buttons: 'Cancel' and 'OK'. Below the modal is a table with columns: Book ID, Title, Book URL, ISBN, Image URL, Author, Author URL, Rating, Rating Count, Review Count, and Similar Books. The table contains three rows of book data. Each row has a 'Delete' button and an 'Edit' button in a red-bordered box. The first book is 'Clean Architecture' by Robert C. Martin, the second is 'Agile Software Development, Principles, Patterns, and Practices' by Robert C. Martin, and the third is 'Agile Principles, Patterns, and Practices' by Robert C. Martin. The 'Similar Books' column for each row contains the text 'Click to see similar books'.

	Book ID	Title	ISBN	Author	Image	Rating	Rating Count	Review Count	Similar Books
<a href="#">Delete</a> <a href="#">Edit</a>	18043011	Clean Architecture 1	0134494164	Robert C. Martin		3	33811	2742	Click to see similar books
<a href="#">Delete</a> <a href="#">Edit</a>	84985	Agile Software Development, Principles, Patterns, and Practices	0135974445	Robert C. Martin		4.25	1192	49	Click to see similar books
<a href="#">Delete</a> <a href="#">Edit</a>	84983	Agile Principles, Patterns, and	0131857258	Robert C. Martin		4.3	713	34	Click to see similar books

After clicking ok in the confirmation box, a success box will pop up and the table will be refreshed.

React App Data | Atlas: MongoDB Atlas localhost:3000/api

- Table
- Authors Visualization
- Books Visualization

**Books**

Search Books

Book ID	Title	Book URL

ISBN	Image URL

Author	Author URL

Rating	Rating Count	Review Count

Add Book

	Book ID	Title	ISBN	Author	Image	Rating	Rating Count	Review Count	Similar Books
<a href="#">Delete</a>	84985	Agile Software Development, Principles, Patterns, and Practices	0135974445	Robert C. Martin		4.25	1192	49	Click to see similar books
<a href="#">Delete</a>	84983	Agile Principles, Patterns, and Practices in C#	0131857258	Robert C. Martin		4.3	713	34	Click to see similar books
<a href="#">Delete</a>	45280021	Clean Agile: Back to Basics	0135781868	Robert C. Martin		4.32	544	79	Click to see similar books

Similarly, for deleting author:

React App Data | Atlas: MongoDB Atlas localhost:3000/api

localhost:3000 says  
Want to delete?

[Cancel](#) [OK](#)

	Author ID	Name	Image	Rating	Rating Count	Review Count	Author's Books	Similar Authors
<a href="#">Delete</a>	3174788	Frederick P. Brooks Jr.		4	1227	839	Click to see author's books	Click to see similar authors
<a href="#">Delete</a>	25215	Martin Fowler		4.12	15843	722	Click to see author's books	Click to see similar authors
<a href="#">Delete</a>	48622	Erich Gamma		4	1672	612	Click to see author's books	Click to see similar authors
<a href="#">Delete</a>	153350	Brian W. Kernighan		4.3	16936	862	Click to see author's books	Click to see similar authors
<a href="#">Delete</a>	7969625	Martin Kleppmann		4.7	4088	442	Click to see author's books	Click to see similar authors
<a href="#">Delete</a>	14019088	John		4.14	1357	199	Click to see author's books	Click to see similar authors

React App Data | Atlas: MongoDB Atlas localhost:3000/api

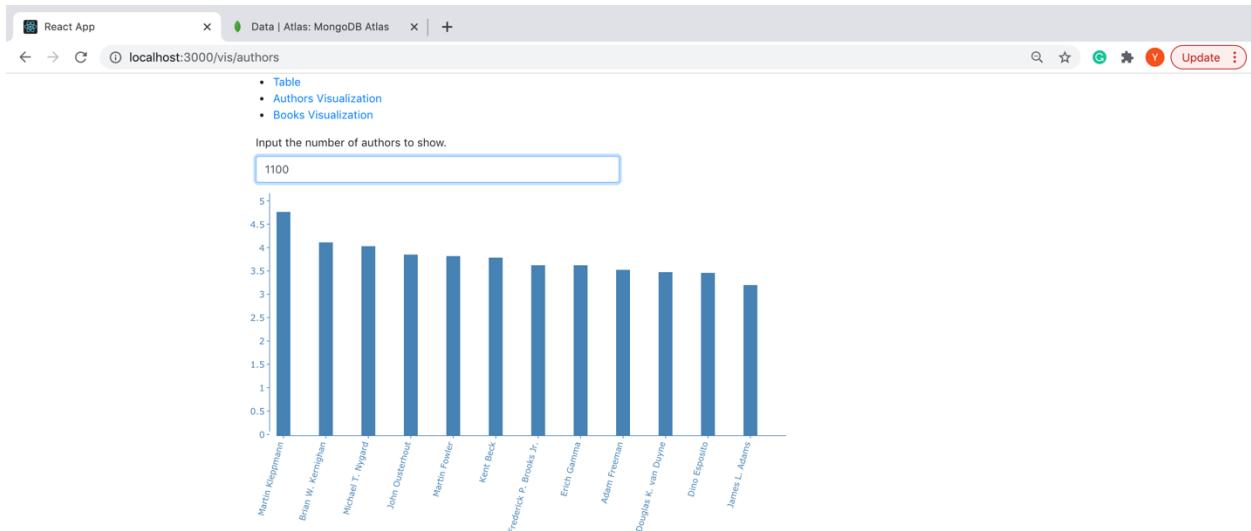
**Authors**

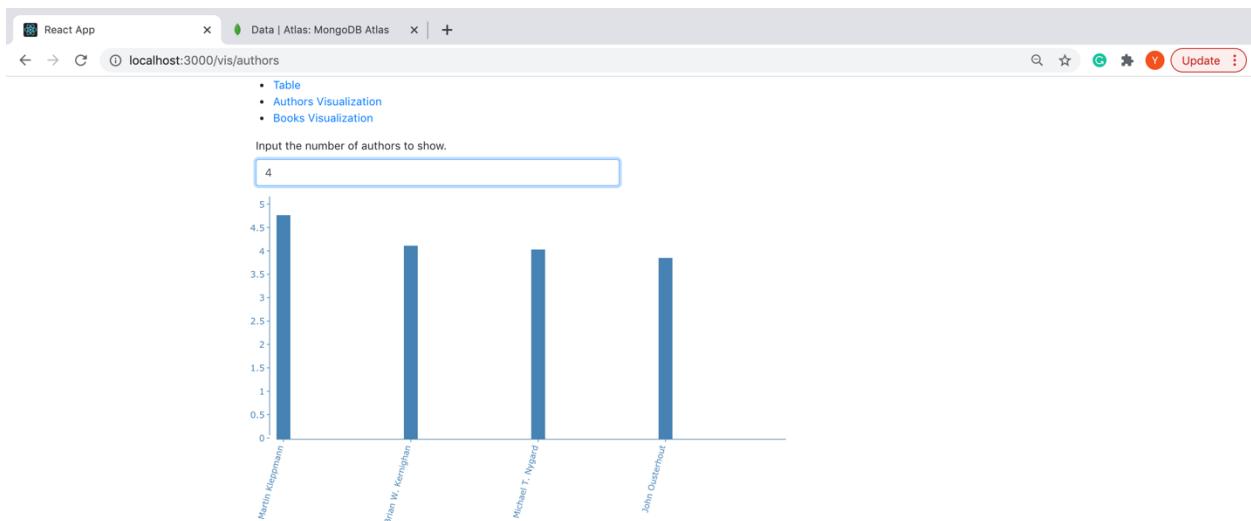
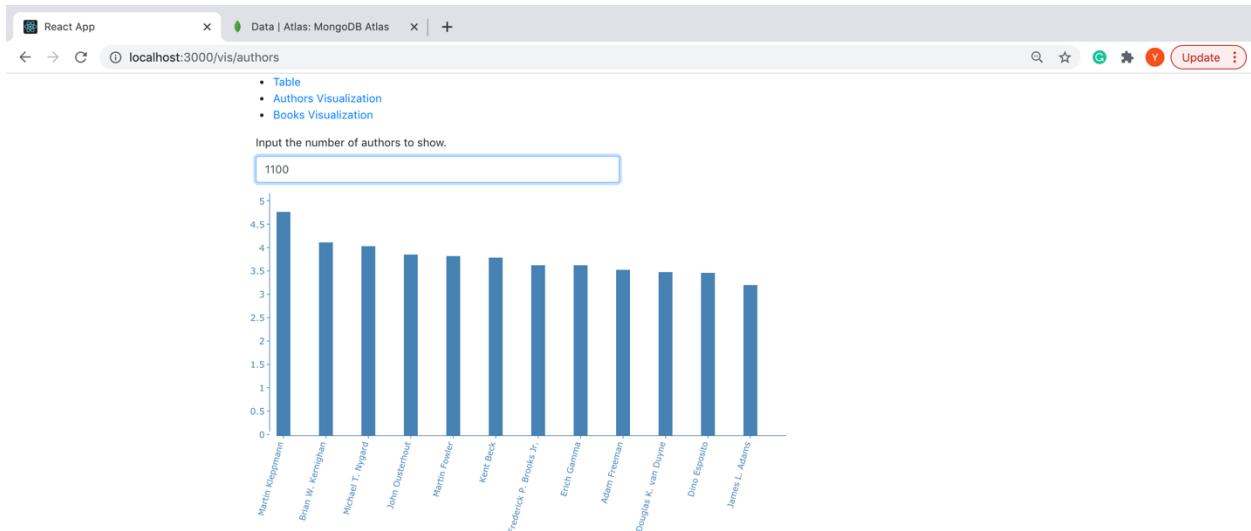
Successfully deleted book!

	Author ID	Name	Image	Rating	Rating Count	Review Count	Author's Books	Similar Authors
<a href="#">Delete</a>	3174788	Frederick P. Brooks Jr.		4	1227	839	<a href="#">Click to see author's books</a>	<a href="#">Click to see similar authors</a>
<a href="#">Edit</a>	25215	Martin Fowler		4.12	15843	722	<a href="#">Click to see author's books</a>	<a href="#">Click to see similar authors</a>
<a href="#">Delete</a>	48622	Erich Gamma		4	1672	612	<a href="#">Click to see author's books</a>	<a href="#">Click to see similar authors</a>
<a href="#">Edit</a>	153350	Brian W. Kernighan		4.3	16936	862	<a href="#">Click to see author's books</a>	<a href="#">Click to see similar authors</a>
<a href="#">Delete</a>	7969625	Martin Kleppmann		4.7	4088	442	<a href="#">Click to see author's books</a>	<a href="#">Click to see similar authors</a>
<a href="#">Edit</a>	14019088	John		4.14	1357	199	<a href="#">Click to see author's books</a>	<a href="#">Click to see similar authors</a>
<a href="#">More...</a>								

## Test Authors Visualization:

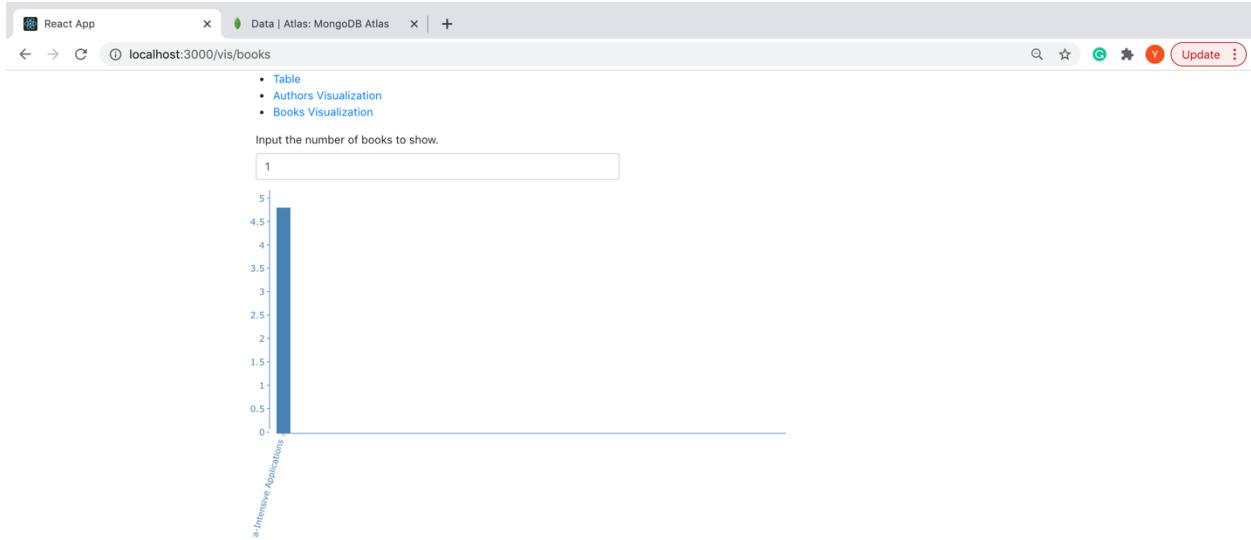
Users can input the number k to demonstrate the top k authors' rating:





# Test Books Visualization:

Users can input the number k to demonstrate the top k books' rating:



The web page is also responsive: it can be used on a phone or pad.

React App Data | Atlas: MongoDB Atlas localhost:3000/api iPhone X 375 x 812 90% Online

Books

Search Books

ID

Query

Search

Book ID

Title

Book URL

ISBN

Image URL

Author

lete  
dit

4845

Code Complete

0735

lete  
dit

10284614

The Clean Coder: A Code of Conduct for Professional Programmers

0137

lete  
dit

58128

Head First Design Patterns

0596

lete  
dit

84985

Agile Software Development, Principles, Patterns, and Practices

0135

10 >

1 2 >

Elements Console Sources Network

This HTML file is a template.  
If you open it directly in the browser, you will see an empty page.

You can add webfonts, meta tags, or analytics to this file.  
The build step will place the bundled scripts into the <body> tag.

To begin the development, run 'npm start' or 'yarn start'.  
To create a production bundle, use 'npm run build' or 'yarn build'.

<script src="/static/js/bundle.js"></script>  
<script src="/static/js/vendors/main.chunk.js"></script>  
<script src="/static/js/main.chunk.js"></script>

body div#root

Styles Computed Layout Event Listeners DOM Breakpoints Properties Accessibility

element.style {

Console What's New

New CSS angle visualization tools  
Better visualize and edit CSS angle in the Styles pane.

Emulate unsupported image types  
2 new emulations to disable AVIF and WebP image formats in the Rendering tab.

Simulate storage quota size

React App Data | Atlas: MongoDB Atlas localhost:3000/api iPhone X 375 x 812 90% Online

lete  
dit

4845

Code Complete

0735

lete  
dit

10284614

The Clean Coder: A Code of Conduct for Professional Programmers

0137

lete  
dit

58128

Head First Design Patterns

0596

lete  
dit

84985

Agile Software Development, Principles, Patterns, and Practices

0135

10 >

1 2 >

Elements Console Sources Network

This HTML file is a template.  
If you open it directly in the browser, you will see an empty page.

You can add webfonts, meta tags, or analytics to this file.  
The build step will place the bundled scripts into the <body> tag.

To begin the development, run 'npm start' or 'yarn start'.  
To create a production bundle, use 'npm run build' or 'yarn build'.

<script src="/static/js/bundle.js"></script>  
<script src="/static/js/vendors/main.chunk.js"></script>  
<script src="/static/js/main.chunk.js"></script>

body div#root

Styles Computed Layout Event Listeners DOM Breakpoints Properties Accessibility

element.style {

Console What's New

New CSS angle visualization tools  
Better visualize and edit CSS angle in the Styles pane.

Emulate unsupported image types  
2 new emulations to disable AVIF and WebP image formats in the Rendering tab.

Simulate storage quota size