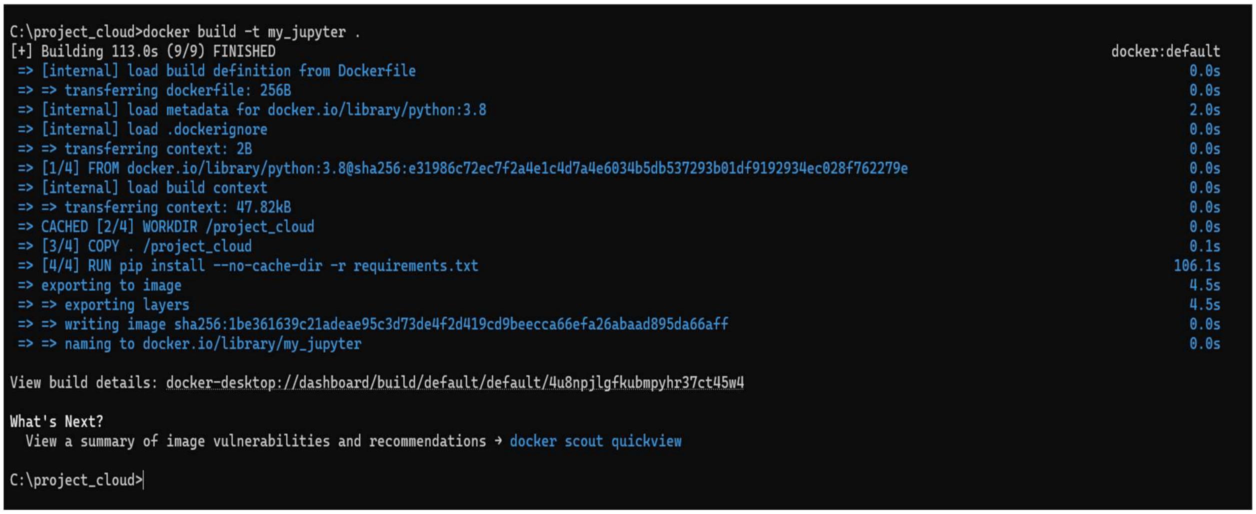
Cloud computing

Name: Karim Basuny Abdelrazik Ibrahim

ID: 2206205

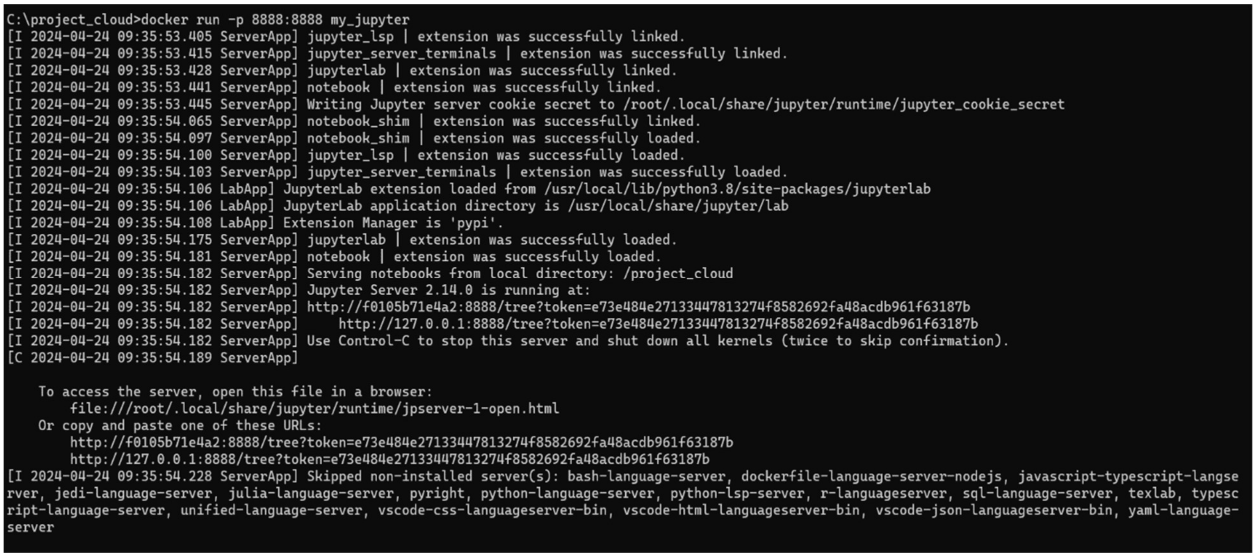
Cybersecurity Level2

CMD command to build docker image:



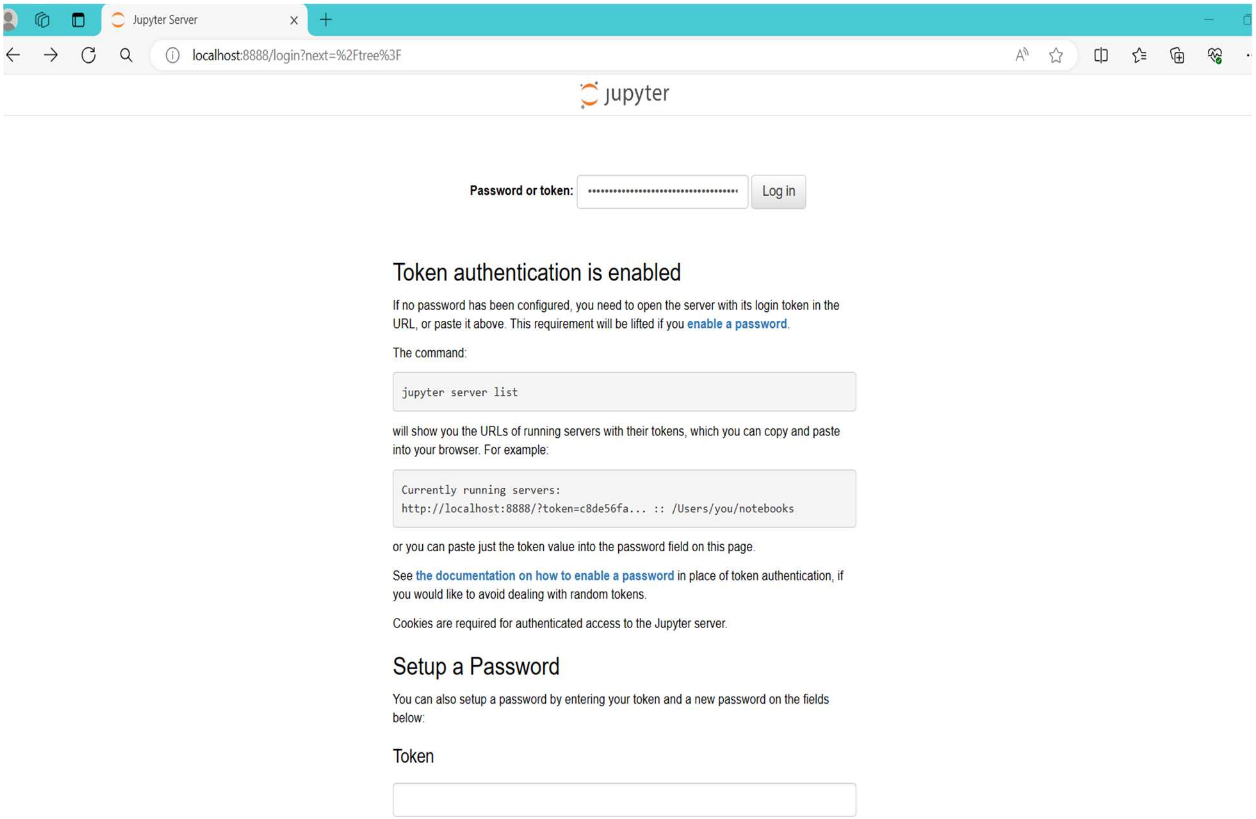
```
C:\project_cloud>docker build -t my_jupyter .
[+] Building 113.0s (9/9) FINISHED                                                                                                   docker:default
 => [internal] load build definition from Dockerfile                                                                                           0.0s
 => => transferring dockerfile: 256B                                                                                                           0.0s
 => [internal] load metadata for docker.io/library/python:3.8                                                                                  2.0s
 => [internal] load .dockerignore                                                                                                              0.0s
 => => transferring context: 2B                                                                                                                0.0s
 => [1/4] FROM docker.io/library/python:3.8@sha256:e31986c72ec7f2a4e1c4d7a4e6034b5db537293b01df9192934ec028f762279e                            0.0s
 => [internal] load build context                                                                                                              0.0s
 => => transferring context: 47.82kB                                                                                                           0.0s
 => CACHED [2/4] WORKDIR /project_cloud                                                                                                        0.0s
 => [3/4] COPY . /project_cloud                                                                                                                0.1s
 => [4/4] RUN pip install --no-cache-dir -r requirements.txt                                                                                 106.1s
 => exporting to image                                                                                                                         4.5s
 => => exporting layers                                                                                                                        4.5s
 => => writing image sha256:1be361639c21adeae95c3d73de4f2d419cd9beecca66efa26abaad895da66aff                                                   0.0s
 => => naming to docker.io/library/my_jupyter                                                                                                  0.0s

View build details: docker-desktop://dashboard/build/default/default/4u8npjlgfkubmpyhr37ct45w4

What's Next?
  View a summary of image vulnerabilities and recommendations → docker scout quickview

C:\project_cloud>
```

Command to run docker container:



```
C:\project_cloud>docker run -p 8888:8888 my_jupyter
[I 2024-04-24 09:35:53.405 ServerApp] jupyter_lsp | extension was successfully linked.
[I 2024-04-24 09:35:53.415 ServerApp] jupyter_server_terminals | extension was successfully linked.
[I 2024-04-24 09:35:53.428 ServerApp] jupyterlab | extension was successfully linked.
[I 2024-04-24 09:35:53.441 ServerApp] notebook | extension was successfully linked.
[I 2024-04-24 09:35:53.445 ServerApp] Writing Jupyter server cookie secret to /root/.local/share/jupyter/runtime/jupyter_cookie_secret
[I 2024-04-24 09:35:54.065 ServerApp] notebook_shim | extension was successfully linked.
[I 2024-04-24 09:35:54.097 ServerApp] notebook_shim | extension was successfully loaded.
[I 2024-04-24 09:35:54.100 ServerApp] jupyter_lsp | extension was successfully loaded.
[I 2024-04-24 09:35:54.103 ServerApp] jupyter_server_terminals | extension was successfully loaded.
[I 2024-04-24 09:35:54.106 LabApp] JupyterLab extension loaded from /usr/local/lib/python3.8/site-packages/jupyterlab
[I 2024-04-24 09:35:54.106 LabApp] JupyterLab application directory is /usr/local/share/jupyter/lab
[I 2024-04-24 09:35:54.108 LabApp] Extension Manager is 'pypi'.
[I 2024-04-24 09:35:54.175 ServerApp] jupyterlab | extension was successfully loaded.
[I 2024-04-24 09:35:54.181 ServerApp] notebook | extension was successfully loaded.
[I 2024-04-24 09:35:54.182 ServerApp] Serving notebooks from local directory: /project_cloud
[I 2024-04-24 09:35:54.182 ServerApp] Jupyter Server 2.14.0 is running at:
[I 2024-04-24 09:35:54.182 ServerApp] http://f0105b71e4a2:8888/tree?token=e73e484e27133447813274f8582692fa48acdb961f63187b
[I 2024-04-24 09:35:54.182 ServerApp]     http://127.0.0.1:8888/tree?token=e73e484e27133447813274f8582692fa48acdb961f63187b
[I 2024-04-24 09:35:54.182 ServerApp] Use Control-C to stop this server and shut down all kernels (twice to skip confirmation).
[C 2024-04-24 09:35:54.189 ServerApp]

    To access the server, open this file in a browser:
        file:///root/.local/share/jupyter/runtime/jpserver-1-open.html
    Or copy and paste one of these URLs:
        http://f0105b71e4a2:8888/tree?token=e73e484e27133447813274f8582692fa48acdb961f63187b
        http://127.0.0.1:8888/tree?token=e73e484e27133447813274f8582692fa48acdb961f63187b
[I 2024-04-24 09:35:54.228 ServerApp] Skipped non-installed server(s): bash-language-server, dockerfile-language-server-nodejs, javascript-typescript-langse
rver, jedi-language-server, julia-language-server, pyright, python-language-server, python-lsp-server, r-languageserver, sql-language-server, texlab, typesc
ript-language-server, unified-language-server, vscode-css-languageserver-bin, vscode-html-languageserver-bin, vscode-json-languageserver-bin, yaml-language-
server
```

In running container command when you run it you will find a token copy it then enter in your browser localhost:8888 and paste it



Password or token: ••

Token authentication is enabled

If no password has been configured, you need to open the server with its login token in the URL, or paste it above. This requirement will be lifted if you **enable a password**.

The command:

```
jupyter server list
```

will show you the URLs of running servers with their tokens, which you can copy and paste into your browser. For example:

```
Currently running servers:
http://localhost:8888/?token=c8de56fa... :: /Users/you/notebooks
```

or you can paste just the token value into the password field on this page.

See **the documentation on how to enable a password** in place of token authentication, if you would like to avoid dealing with random tokens.

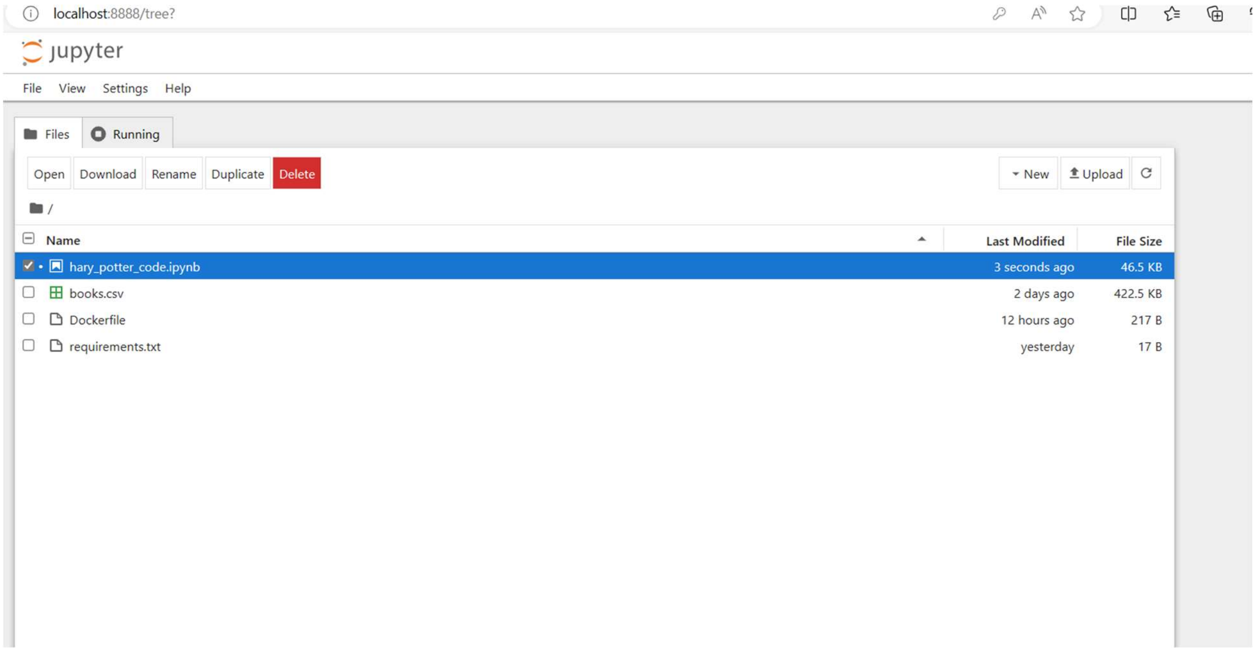
Cookies are required for authenticated access to the Jupyter server.

Setup a Password

You can also setup a password by entering your token and a new password on the fields below:

Token

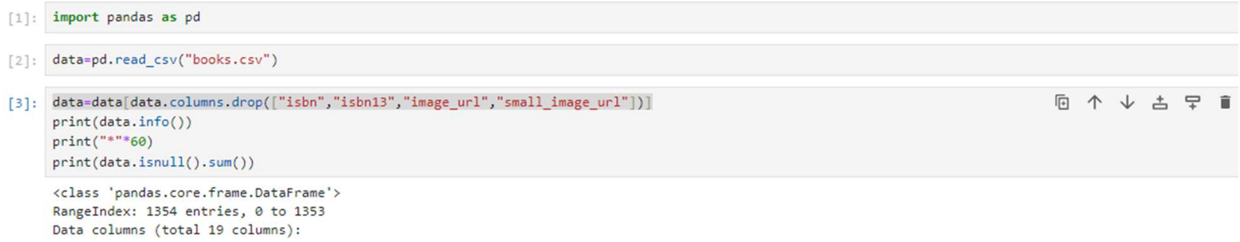
After that it will open your file and you will se the notebook and docker file and requirements file



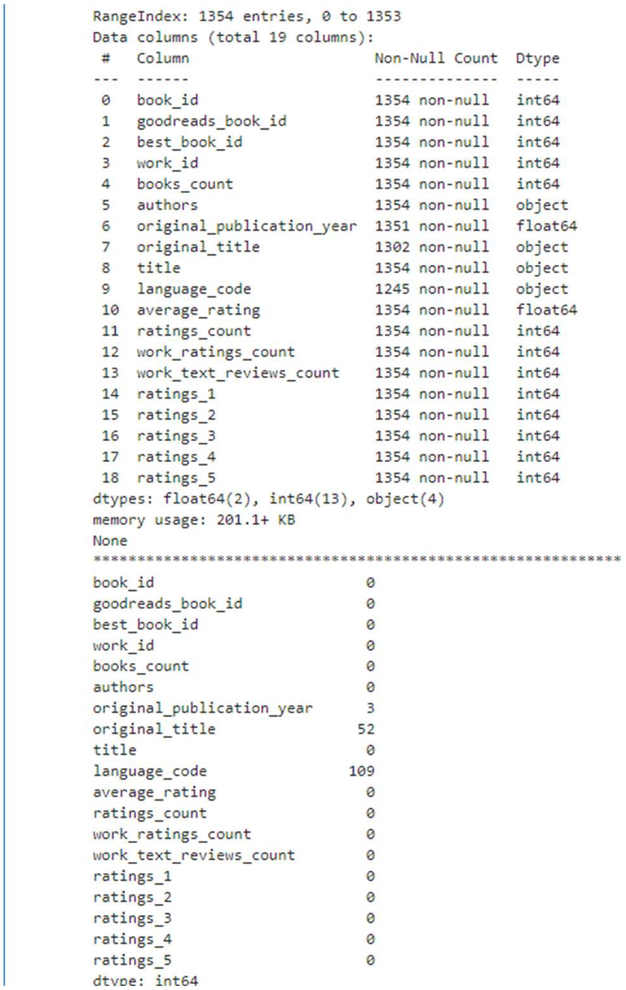
Name	Last Modified	File Size
☑ hary_potter_code.ipynb	3 seconds ago	46.5 KB
□ books.csv	2 days ago	422.5 KB
□ Dockerfile	12 hours ago	217 B
□ requirements.txt	yesterday	17 B

In block 3 it drops the cols with name provided and it creates a new data frame with the rest of data

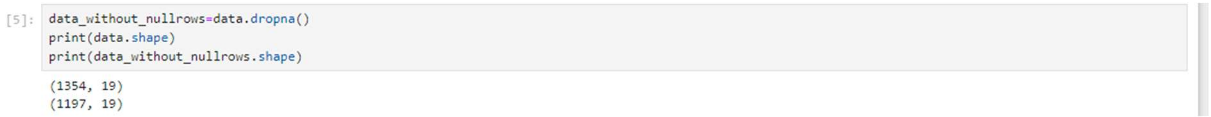
And calculated the sum on null values in each column



```
[1]: import pandas as pd

[2]: data=pd.read_csv("books.csv")

[3]: data=data[data.columns.drop(["isbn","isbn13","image_url","small_image_url"])]
     print(data.info())
     print("*"*60)
     print(data.isnull().sum())

     <class 'pandas.core.frame.DataFrame'>
     RangeIndex: 1354 entries, 0 to 1353
     Data columns (total 19 columns):
```

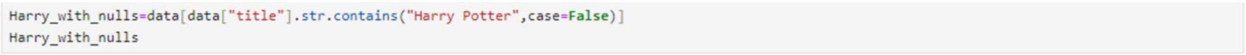


```
RangeIndex: 1354 entries, 0 to 1353
Data columns (total 19 columns):
 #   Column                     Non-Null Count  Dtype
---  ------                     --------------  -----
 0   book_id                    1354 non-null   int64
 1   goodreads_book_id          1354 non-null   int64
 2   best_book_id               1354 non-null   int64
 3   work_id                    1354 non-null   int64
 4   books_count                1354 non-null   int64
 5   authors                    1354 non-null   object
 6   original_publication_year  1351 non-null   float64
 7   original_title             1302 non-null   object
 8   title                      1354 non-null   object
 9   language_code              1245 non-null   object
 10  average_rating             1354 non-null   float64
 11  ratings_count              1354 non-null   int64
 12  work_ratings_count         1354 non-null   int64
 13  work_text_reviews_count    1354 non-null   int64
 14  ratings_1                  1354 non-null   int64
 15  ratings_2                  1354 non-null   int64
 16  ratings_3                  1354 non-null   int64
 17  ratings_4                  1354 non-null   int64
 18  ratings_5                  1354 non-null   int64
dtypes: float64(2), int64(13), object(4)
memory usage: 201.1+ KB
None
************************************************************
book_id                        0
goodreads_book_id              0
best_book_id                   0
work_id                        0
books_count                    0
authors                        0
original_publication_year      3
original_title                52
title                          0
language_code                109
average_rating                 0
ratings_count                  0
work_ratings_count             0
work_text_reviews_count        0
ratings_1                      0
ratings_2                      0
ratings_3                      0
ratings_4                      0
ratings_5                      0
dtype: int64
```



```
[5]: data_without_nullrows=data.dropna()
     print(data.shape)
     print(data_without_nullrows.shape)

     (1354, 19)
     (1197, 19)
```

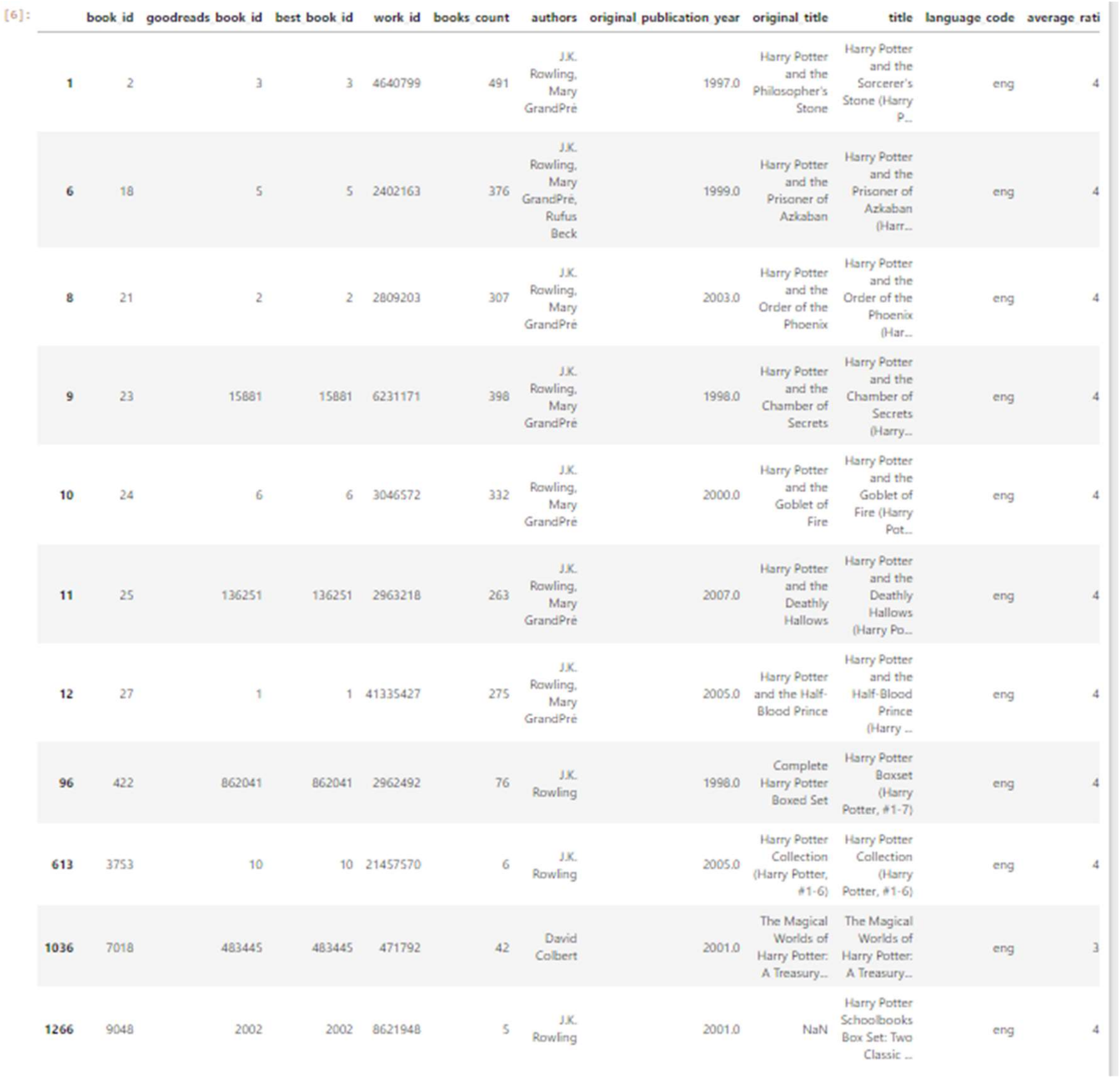
drops any rows from the DataFrame data that contain missing values . It returns a new DataFrame called data_without_nullrows which contains only the rows with complete data and then prints the shape of the original DataFrame then prints the shape of the new DataFrame



```
Harry_with_nulls=data[data["title"].str.contains("Harry Potter",case=False)]
Harry_with_nulls
```

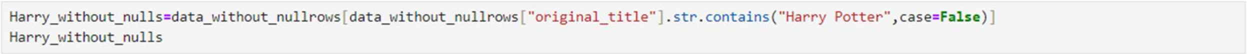
filtering the DataFrame data to select rows where the "title" column contains the substring Harry Potter also, the DataFrame data may contain NaN values in its "title" column.and its case-insensitive.

And here the output:



[6]:

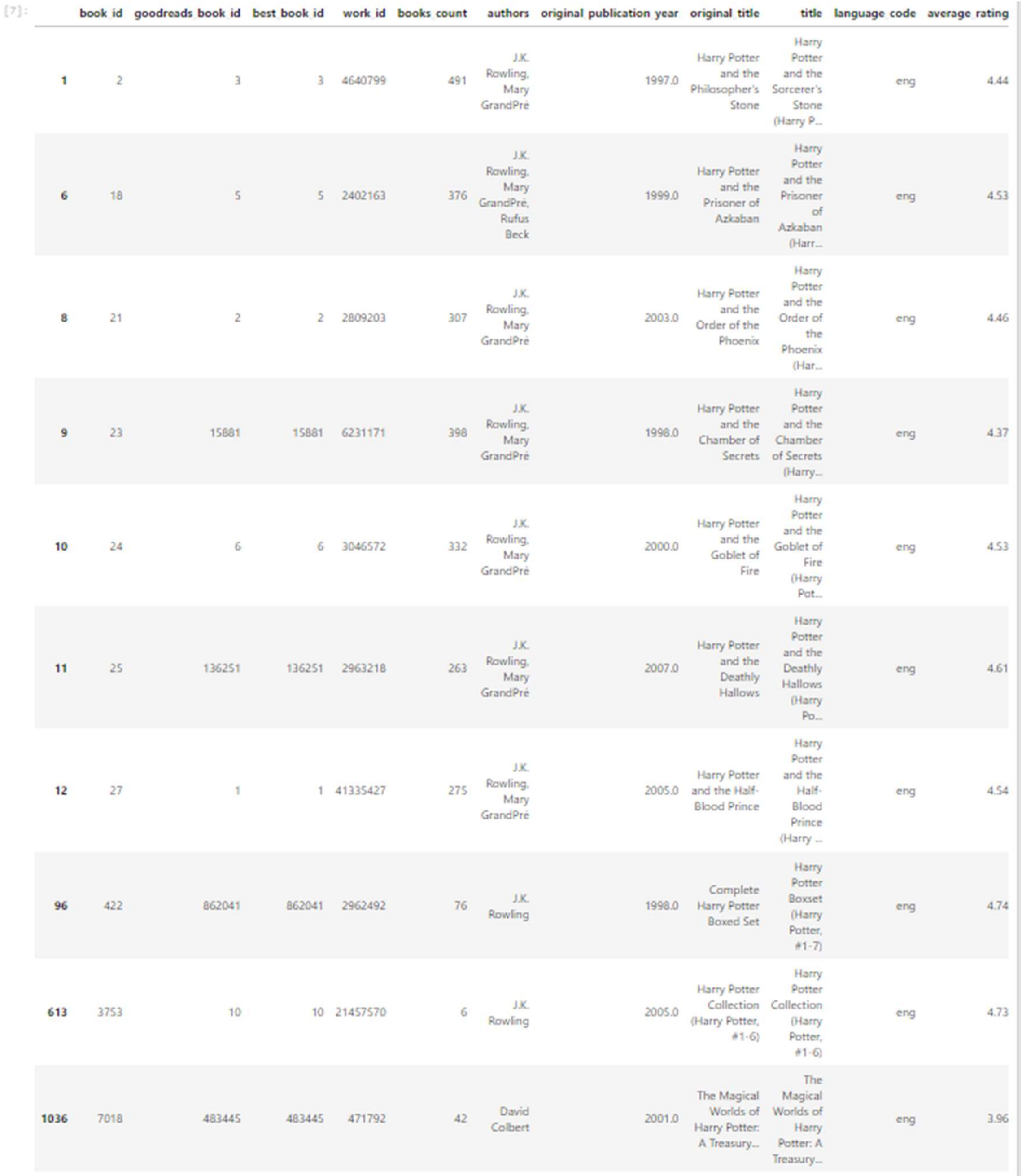
	book_id	goodreads_book_id	best_book_id	work_id	books_count	authors	original_publication_year	original_title	title	language_code	average_rati
1	2	3	3	4640799	491	J.K. Rowling, Mary GrandPré	1997.0	Harry Potter and the Philosopher's Stone	Harry Potter and the Sorcerer's Stone (Harry P...	eng	4
6	18	5	5	2402163	376	J.K. Rowling, Mary GrandPré, Rufus Beck	1999.0	Harry Potter and the Prisoner of Azkaban	Harry Potter and the Prisoner of Azkaban (Harr...	eng	4
8	21	2	2	2809203	307	J.K. Rowling, Mary GrandPré	2003.0	Harry Potter and the Order of the Phoenix	Harry Potter and the Order of the Phoenix (Har...	eng	4
9	23	15881	15881	6231171	398	J.K. Rowling, Mary GrandPré	1998.0	Harry Potter and the Chamber of Secrets	Harry Potter and the Chamber of Secrets (Harry...	eng	4
10	24	6	6	3046572	332	J.K. Rowling, Mary GrandPré	2000.0	Harry Potter and the Goblet of Fire	Harry Potter and the Goblet of Fire (Harry Pot...	eng	4
11	25	136251	136251	2963218	263	J.K. Rowling, Mary GrandPré	2007.0	Harry Potter and the Deathly Hallows	Harry Potter and the Deathly Hallows (Harry Po...	eng	4
12	27	1	1	41335427	275	J.K. Rowling, Mary GrandPré	2005.0	Harry Potter and the Half-Blood Prince	Harry Potter and the Half-Blood Prince (Harry ...	eng	4
96	422	862041	862041	2962492	76	J.K. Rowling	1998.0	Complete Harry Potter Boxed Set	Harry Potter Boxset (Harry Potter, #1-7)	eng	4
613	3753	10	10	21457570	6	J.K. Rowling	2005.0	Harry Potter Collection (Harry Potter, #1-6)	Harry Potter Collection (Harry Potter, #1-6)	eng	4
1036	7018	483445	483445	471792	42	David Colbert	2001.0	The Magical Worlds of Harry Potter: A Treasury...	The Magical Worlds of Harry Potter: A Treasury...	eng	3
1266	9048	2002	2002	8621948	5	J.K. Rowling	2001.0	NaN	Harry Potter Schoolbooks Box Set: Two Classic ...	eng	4



```
Harry_without_nulls=data_without_nullrows[data_without_nullrows["original_title"].str.contains("Harry Potter",case=False)]
Harry_without_nulls
```

the DataFrame obtained after removing rows with missing values So, Harry_without_nulls will contain all rows from data_without_nullrows where the "original_title" column contains the substring Harry Potter in a case-insensitive manner. Since data_without_nullrows doesn't have any NaN values, it only considers rows with complete data.

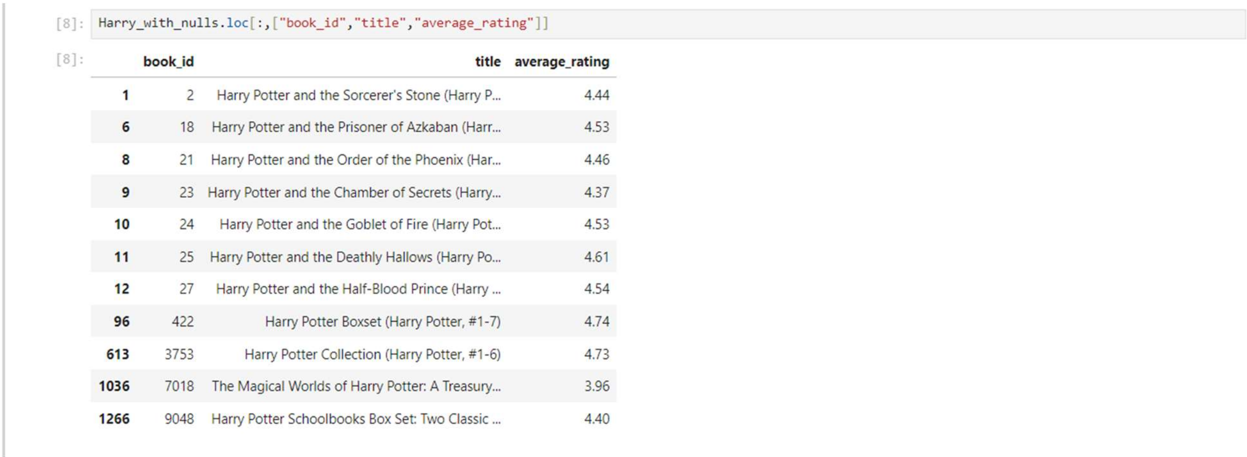
And here the output:



[7]:

	book_id	goodreads_book_id	best_book_id	work_id	books_count	authors	original_publication_year	original_title	title	language_code	average_rating
1	2	3	3	4640799	491	J.K. Rowling, Mary GrandPré	1997.0	Harry Potter and the Philosopher's Stone	Harry Potter and the Sorcerer's Stone (Harry P...	eng	4.44
6	18	5	5	2402163	376	J.K. Rowling, Mary GrandPré, Rufus Beck	1999.0	Harry Potter and the Prisoner of Azkaban	Harry Potter and the Prisoner of Azkaban (Harr...	eng	4.53
8	21	2	2	2809203	307	J.K. Rowling, Mary GrandPré	2003.0	Harry Potter and the Order of the Phoenix	Harry Potter and the Order of the Phoenix (Har...	eng	4.46
9	23	15881	15881	6231171	398	J.K. Rowling, Mary GrandPré	1998.0	Harry Potter and the Chamber of Secrets	Harry Potter and the Chamber of Secrets (Harry...	eng	4.37
10	24	6	6	3046572	332	J.K. Rowling, Mary GrandPré	2000.0	Harry Potter and the Goblet of Fire	Harry Potter and the Goblet of Fire (Harry Pot...	eng	4.53
11	25	136251	136251	2963218	263	J.K. Rowling, Mary GrandPré	2007.0	Harry Potter and the Deathly Hallows	Harry Potter and the Deathly Hallows (Harry Po...	eng	4.61
12	27	1	1	41335427	275	J.K. Rowling, Mary GrandPré	2005.0	Harry Potter and the Half-Blood Prince	Harry Potter and the Half-Blood Prince (Harry ...	eng	4.54
96	422	862041	862041	2962492	76	J.K. Rowling	1998.0	Complete Harry Potter Boxed Set	Harry Potter Boxset (Harry Potter, #1-7)	eng	4.74
613	3753	10	10	21457570	6	J.K. Rowling	2005.0	Harry Potter Collection (Harry Potter, #1-6)	Harry Potter Collection (Harry Potter, #1-6)	eng	4.73
1036	7018	483445	483445	471792	42	David Colbert	2001.0	The Magical Worlds of Harry Potter: A Treasury...	The Magical Worlds of Harry Potter: A Treasury...	eng	3.96

this code will return a DataFrame containing the specified columns ("book_id", "title", and "average_rating") for all rows in Harry_with_nulls using loc[]: This is a method used to access a group of rows and columns by labels.

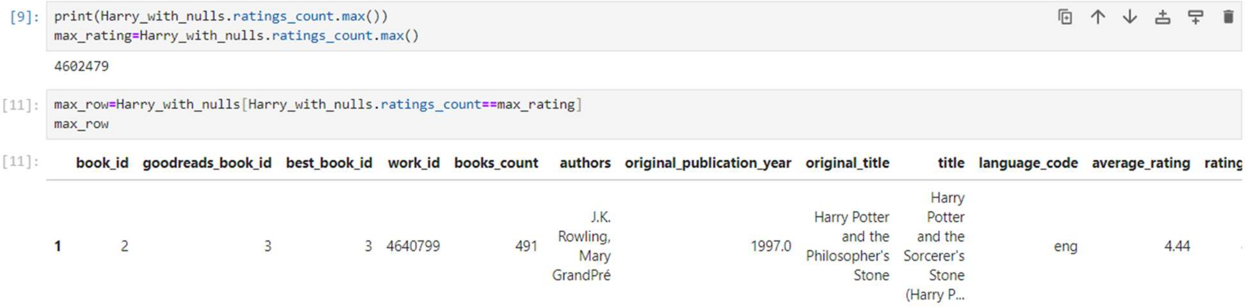


```
[8]: Harry_with_nulls.loc[:,["book_id","title","average_rating"]]
```

[8]:

	book_id	title	average_rating
1	2	Harry Potter and the Sorcerer's Stone (Harry P...	4.44
6	18	Harry Potter and the Prisoner of Azkaban (Harr...	4.53
8	21	Harry Potter and the Order of the Phoenix (Har...	4.46
9	23	Harry Potter and the Chamber of Secrets (Harry...	4.37
10	24	Harry Potter and the Goblet of Fire (Harry Pot...	4.53
11	25	Harry Potter and the Deathly Hallows (Harry Po...	4.61
12	27	Harry Potter and the Half-Blood Prince (Harry ...	4.54
96	422	Harry Potter Boxset (Harry Potter, #1-7)	4.74
613	3753	Harry Potter Collection (Harry Potter, #1-6)	4.73
1036	7018	The Magical Worlds of Harry Potter: A Treasury...	3.96
1266	9048	Harry Potter Schoolbooks Box Set: Two Classic ...	4.40

1. prints the maximum value found in the "ratings_count" column of the DataFrame Harry_with_nulls.
2. assigns the maximum value found in the "ratings_count" column of the DataFrame Harry_with_nulls to the variable max_rating.
3. filters the DataFrame Harry_with_nulls to select rows where the "ratings_count" is equal to max_rating and assigns these rows to the variable max_row.
4. Using max_row to print row with max rating



```
[9]: print(Harry_with_nulls.ratings_count.max())
     max_rating=Harry_with_nulls.ratings_count.max()

     4602479

[11]: max_row=Harry_with_nulls[Harry_with_nulls.ratings_count==max_rating]
      max_row
```

[11]:

	book_id	goodreads_book_id	best_book_id	work_id	books_count	authors	original_publication_year	original_title	title	language_code	average_rating	rating
1	2	3	3	4640799	491	J.K. Rowling, Mary GrandPré	1997.0	Harry Potter and the Philosopher's Stone	Harry Potter and the Sorcerer's Stone (Harry P...	eng	4.44	

[12]:

```
best_seller = data[data['ratings_count'] == max_rating]

print(f"The best selling Harry Potter book is: {best_seller['title'].values[0]}")
```

```
The best selling Harry Potter book is: Harry Potter and the Sorcerer's Stone (Harry Potter, #1)
```

[13]:

```
average_rating = data['average_rating'].mean()
median_rating = data['average_rating'].median()
print("Average Rating:", average_rating)
print("Median Rating:", median_rating)
```

```
Average Rating: 3.999357459379616
Median Rating: 4.0
```

[14]:

```
df = pd.read_csv('books.csv')

hp_books = df[df['authors'].str.contains('J.K. Rowling')]

total_ratings = hp_books['ratings_count'].sum()
average_rating = total_ratings / len(hp_books)

print(f'Average rating of Harry Potter books is {average_rating}')
```

```
Average rating of Harry Potter books is 1310798.8333333333
```