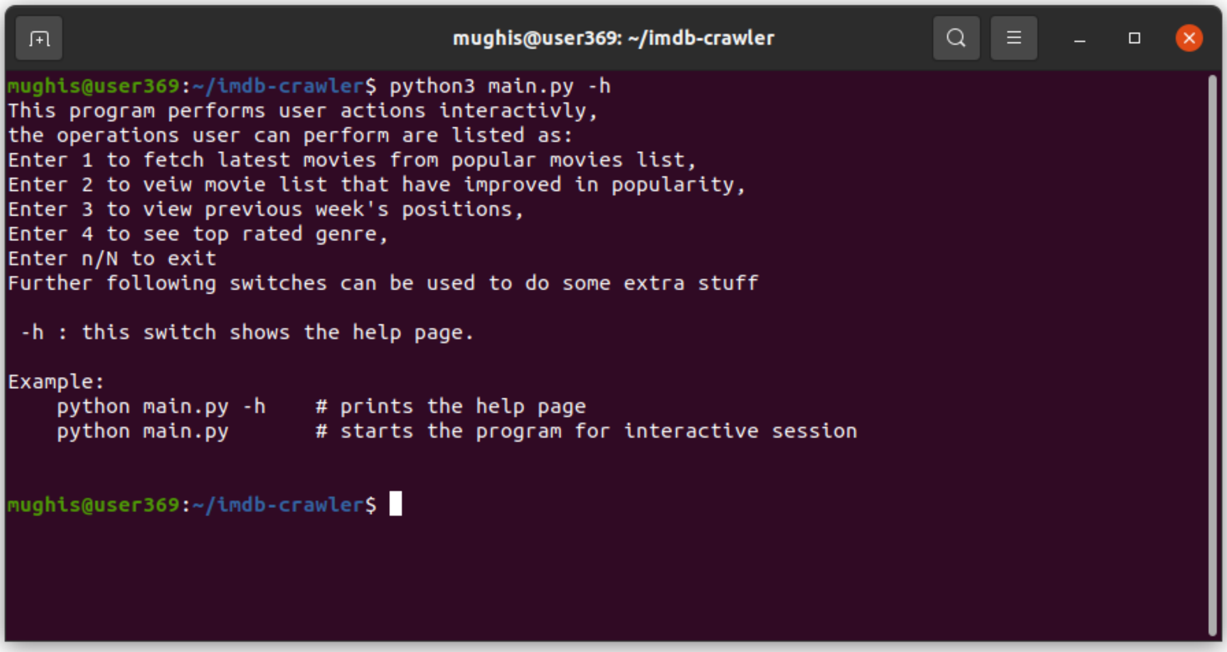
a) Output

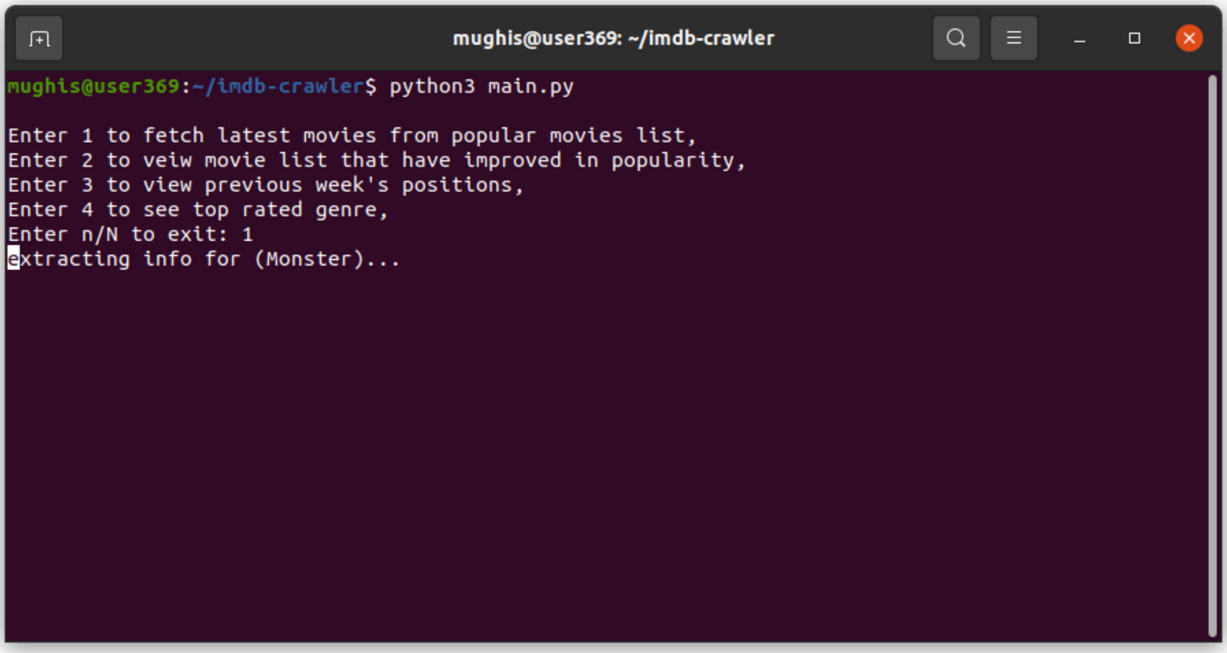
imdb_crawler is started like a normal python script. However, it takes one additional argument for help.



```
mughis@user369:~/imdb-crawler$ python3 main.py -h
This program performs user actions interactivly,
the operations user can perform are listed as:
Enter 1 to fetch latest movies from popular movies list,
Enter 2 to veiw movie list that have improved in popularity,
Enter 3 to view previous week's positions,
Enter 4 to see top rated genre,
Enter n/N to exit
Further following switches can be used to do some extra stuff

 -h : this switch shows the help page.

Example:
    python main.py -h    # prints the help page
    python main.py       # starts the program for interactive session


mughis@user369:~/imdb-crawler$ 
```

Screenshot shows help page of program

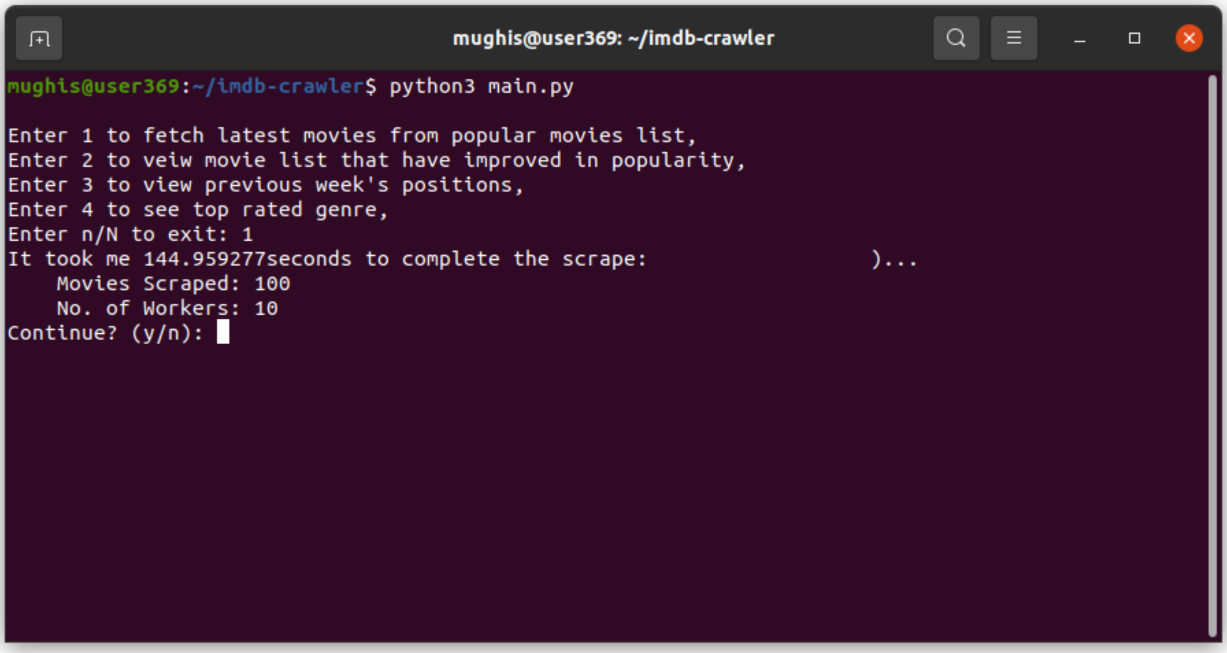
1 – scrapping from most popular page(IMDB)



```
mughis@user369:~/imdb-crawler$ python3 main.py

Enter 1 to fetch latest movies from popular movies list,
Enter 2 to veiw movie list that have improved in popularity,
Enter 3 to view previous week's positions,
Enter 4 to see top rated genre,
Enter n/N to exit: 1
extracting info for (Monster)...
```

Screenshot shows program running in scrape mode

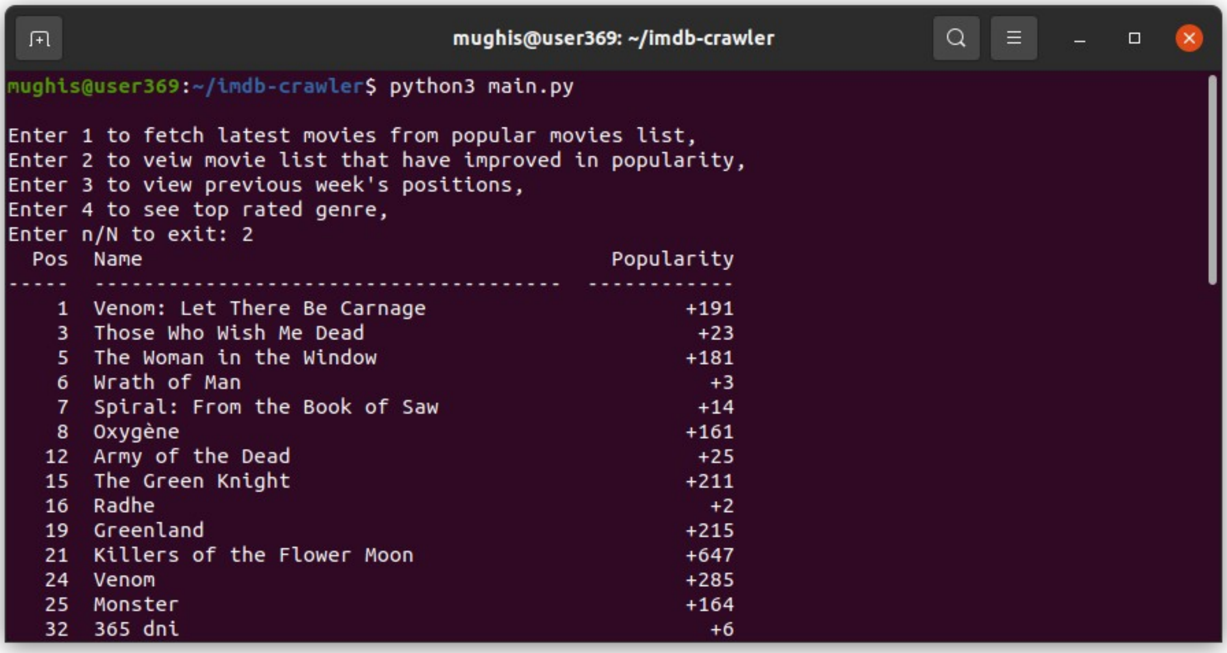
After completing the scrape it prints out the stats and prompt user if the user wants to continue using the program.



```
mughis@user369:~/imdb-crawler$ python3 main.py

Enter 1 to fetch latest movies from popular movies list,
Enter 2 to veiw movie list that have improved in popularity,
Enter 3 to view previous week's positions,
Enter 4 to see top rated genre,
Enter n/N to exit: 1
It took me 144.959277seconds to complete the scrape:                 )...
    Movies Scraped: 100
    No. of Workers: 10
Continue? (y/n):
```

Screenshot shows a completed scrape

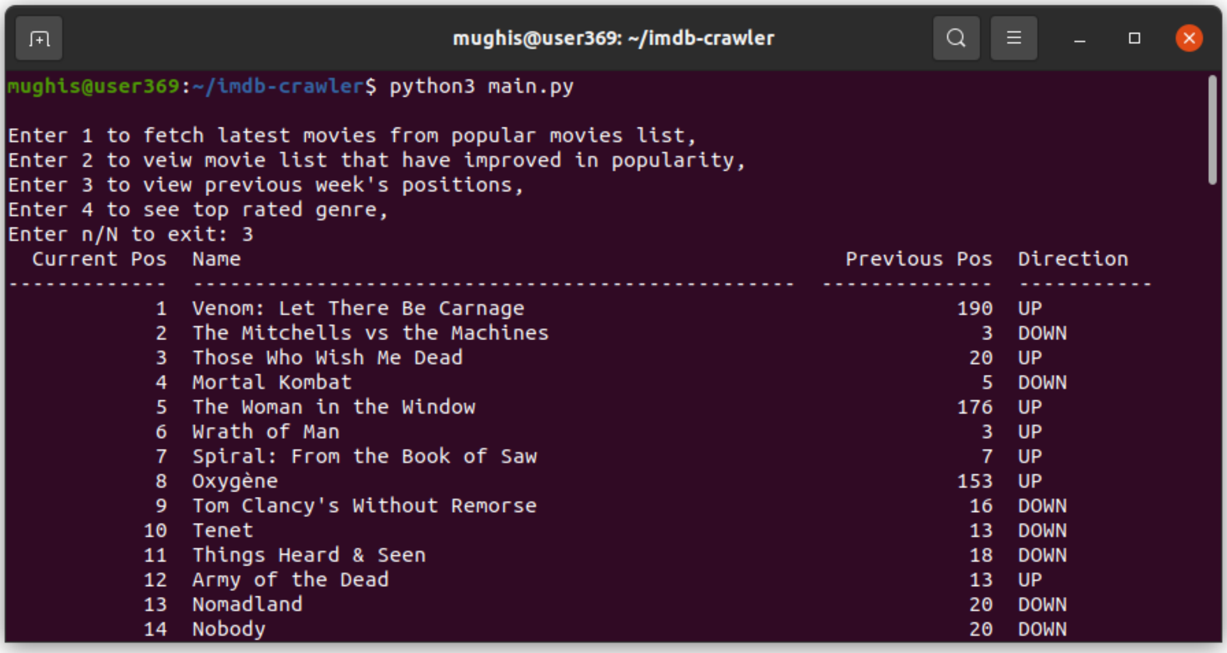
2 – View list of movies that have +ev popularity



```
mughis@user369:~/imdb-crawler$ python3 main.py

Enter 1 to fetch latest movies from popular movies list,
Enter 2 to veiw movie list that have improved in popularity,
Enter 3 to view previous week's positions,
Enter 4 to see top rated genre,
Enter n/N to exit: 2
  Pos  Name                                     Popularity
-----  -------------------------------------  ------------
    1  Venom: Let There Be Carnage                    +191
    3  Those Who Wish Me Dead                          +23
    5  The Woman in the Window                        +181
    6  Wrath of Man                                     +3
    7  Spiral: From the Book of Saw                    +14
    8  Oxygène                                        +161
   12  Army of the Dead                                +25
   15  The Green Knight                               +211
   16  Radhe                                            +2
   19  Greenland                                      +215
   21  Killers of the Flower Moon                     +647
   24  Venom                                          +285
   25  Monster                                        +164
   32  365 dni                                          +6
```

Screenshot shows movies that are increased in popularity

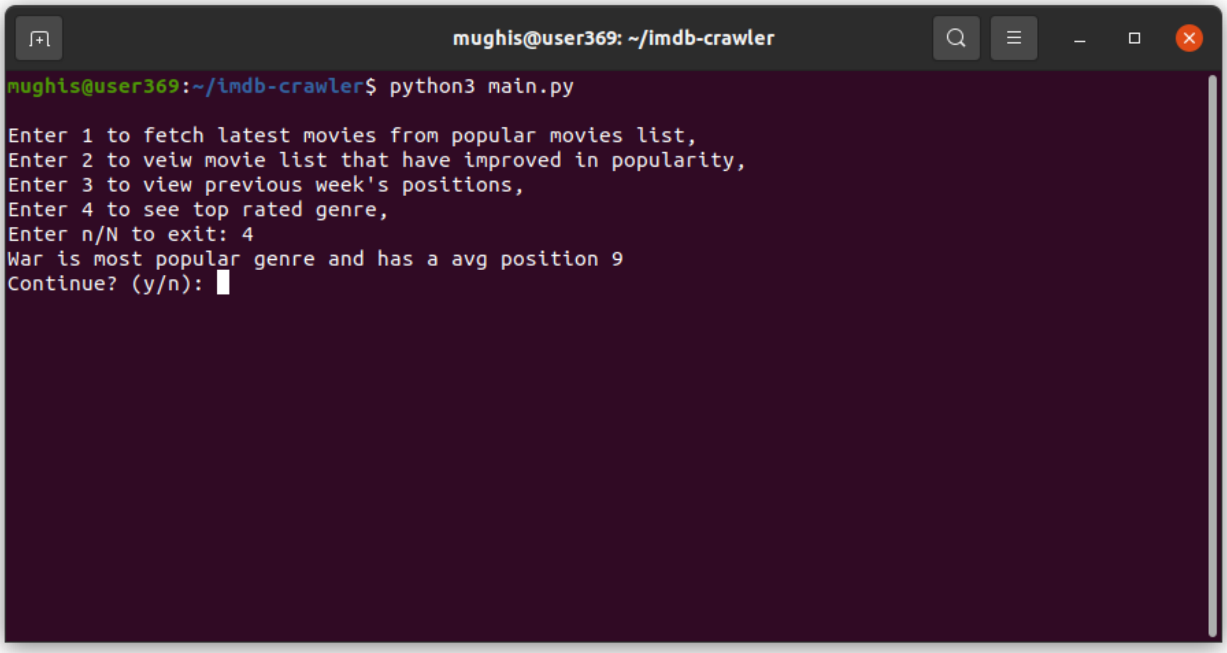
3 – List the position of previous week for all movies
This list is obtained by subtracting/adding the popularity from the position.



```
mughis@user369:~/imdb-crawler$ python3 main.py

Enter 1 to fetch latest movies from popular movies list,
Enter 2 to veiw movie list that have improved in popularity,
Enter 3 to view previous week's positions,
Enter 4 to see top rated genre,
Enter n/N to exit: 3
  Current Pos  Name                                                Previous Pos  Direction
-------------  --------------------------------------------------  --------------  -----------
            1  Venom: Let There Be Carnage                                    190  UP
            2  The Mitchells vs the Machines                                    3  DOWN
            3  Those Who Wish Me Dead                                          20  UP
            4  Mortal Kombat                                                    5  DOWN
            5  The Woman in the Window                                        176  UP
            6  Wrath of Man                                                     3  UP
            7  Spiral: From the Book of Saw                                     7  UP
            8  Oxygène                                                        153  UP
            9  Tom Clancy's Without Remorse                                    16  DOWN
           10  Tenet                                                           13  DOWN
           11  Things Heard & Seen                                             18  DOWN
           12  Army of the Dead                                                13  UP
           13  Nomadland                                                       20  DOWN
           14  Nobody                                                          20  DOWN
```

Screenshot shows list of movies with previous week’s position

4 – Top rated genre from the list



```
mughis@user369:~/imdb-crawler$ python3 main.py

Enter 1 to fetch latest movies from popular movies list,
Enter 2 to veiw movie list that have improved in popularity,
Enter 3 to view previous week's positions,
Enter 4 to see top rated genre,
Enter n/N to exit: 4
War is most popular genre and has a avg position 9
Continue? (y/n):
```

The score is calculated by creating a dictionary of genres and their corresponding position values in a list. This list of position is further used to evaluate the average position for each genre the lower the value the higher genre rating.

b) Technology Stack Used

- Python 3.7
- Requests
- BaeutifulSoup4 (BS4)
- JSON

- As I love to code in python, so there was no confusion about the programming language.
- I chose the requests module for its simplicity.
- For extracting information from HTML, BS4 is used as the parsing was not much intensive. Besides BS4 have much less overhead than Scrappy and Selenium and it has no third-party dependencies.
- JSON is selected over CSV due to its flexible and comprehensive structure.

c) Time it Took

It took me almost 4-5 hours to complete the task but not in a single sitting.

d) Improvements

If I had more time I would've

- Removed the code redundancy in some places.
- Followed object oriented pattern
- Made the output look more cleaner
- Made a simple user interface
- Improve top rated genre calculation (taking average is not that reliable)