Homework 6: Search Application Functionality

Warm-Up: Due 2/28 @ 11:59pm on Courseworks
Main: Due 3/4 @ 11:59pm on Courseworks
For both there is a grace period until 8am the next morning.

For the main assignment, you will create a webpage that lets users search and view. In this assignment, we focus on functionality, not usability or graphic design. The basic functionality your site must allow is to:

1. View data
2. Search the data.

You must use Flask, HTML, CSS, JQuery, and JavaScript. You may use Bootstrap if you want, but it's not required. The changes to the data must be saved to the Flask server.

You must pick a dataset that someone might want to search. Your dataset needs to have at least 10 items in it. We suggest you create them by hand, like the list of local business were in the log_sales application. Typically, 10 items are not enough to need search functionality for, but we are going to implement it anyway.

Each data item needs to have multiple fields. At the minimum, this must include:

- An id
- A short title or name for the item (movie title, restaurant name, etc)
- A link to media (image, video, or gif).
    - For large media files, you must provide an external link (like a link to a youtube video). Do NOT download the file and submit it with the assignment because if you did that, it will take your grader forever to download all the assignments.
- A text paragraph of explanation (At least 4 sentences)
- Some sort of numerical data (a year, a price, a rating, etc.)
- A list of some kind of data (such as a list of reviews for the movie, list of popular dishes at the restaurant, a list of similar restaurants nearby, etc)

Warm-up:

What to submit:

- hw6_writeup.pdf

Problems:

1. Send a message to everyone in your TA group in Slack. Tell them the answers to warmup problems 2, and 3. Include a screenshot of your message for the the write up.
2. What data will your website allow users to search?
    a. Example: *"Crime Comedy Films."* (You may not use this answer)
3. What is an example of a person who would want to search this and why?
    a. Example: *A Columbia student who saw The Big Lebowski and is now fascinated by the genre of crime comedies and wants to similar films to watch.*
    b. Example: *"A Columbia student LOVES bubble tea and wants to visit every bubble tea in Manhattan. They need to look up the best ones to visit first."*
    c. Example. *"A Columbia student is obsessed with all you can eat restaurants and wants to find cheap ones that span many different cultures and cuisines."*
    d. Bad example: "A person wants to see what makeup is available"
    e. Bad example: "A college student likes ice cream and wants to see flavors"
4. Show up an example of one item in the database, and say how the data fields help the user achieve their goal.

Example: *To find similar Crime/Comedy movies they would like, we have a "similar movies" field with a list of other film ids. Users can also click on the actors and director(s) to see other films they are involved in within the genre. These are also likely be similar.*

```
data = {
  "1": {
    "id": "1",
    "title": "The Big Lebowski",
    "image": "https://cdn.theatlantic.com/thumbor/g-
I6jdrxQ6F6fuIhsY_GeP6xaBQ=/0x23:1842x1059/1600x900/media/img/mt/2014/09/Big_Lebow
ski_2/original.jpg",
    "year": "1998",
    "summary": "When 'The Dude' Lebowski is mistaken for a millionaire Lebowski; two thugs
urinate on his rug to coerce him into paying a debt he knows nothing about. While attempting
to gain recompense for the ruined rug from his wealthy counterpart, he accepts a one-time job
with high pay-off. He enlists the help of his bowling buddy, Walter, a gun-toting Jewish-convert
with anger issues. Deception leads to more trouble, and it soon seems that everyone from porn
empire tycoons to nihilists want something from The Dude.",
    "director": ["Joel Coen", "Ethan Coen"],
    "budget": "$15,000,000",
    "stars": ["Jeff Bridges", "John Goodman", "Julianne Moore"],
    "score": "8.1",
    "genres": ["Comedy", "crime", "buddy", "indie", "screwball","stoner"],
    "similar movie ids": ["2", "3","45"]
  },
}
```

Main

What to submit:

1. Hw6_UNI.zip: A Flask project containing:
    - server.py
    - templates/ (and the HTML templates you need)
    - static/ (and any static files you need)
2. A link to a YouTube video showing off the functionality of your site, and a narration of you explaining how the code works.
    a. Write your name in the code somewhere AND say your name at the beginning of the video.

Functionality requirements:

1. **A navbar**
    a. It should contain the title of the website and search bar (and a search button). This should appear on every page of the site.
    b. When you click on the website name, it should take you to the homepage.
    c. When you perform a search, it should take you to a search results page.
        i. Most navbars implement the search as a form element. Form elements have a submit event. Here's some jquery about how to respond to a submit event:
            1. https://api.jquery.com/submit/
        ii. You could also edit the navbar html and change the form element to a regular div.
    d. It doesn't have to be attractive or well-designed in terms of information hierarchy, it just has to function and be readable.
    e. Here is the documentation and examples for bootstrap navbars for Bootstrap 5.0 (the version we are using).
        i. https://getbootstrap.com/docs/5.0/components/navbar/
2. **A homepage**
    a. It should show 3 popular items that the user can click on to view more information. You may select whatever 3 items you like for this, but the interface must be created dynamically in JavaScript and not hard coded in HTML. Thus, if you wanted to change which were the popular items of the week, you could just make a simple change in the JavaScript rather than re-coding your HTML.
    b. It doesn't have to be attractive or well-designed in terms of information hierarchy, it just has to function and be readable.
3. **Search data.**
    a. In the navbar, when the user presses the search button or presses enter in the text input, have a route on the server that takes that information, searches through all the data, and returns only that matching items to a view that shows a list of the search results.

        i. You may **not** search through the results on the client side in JS. You must do it on the server side in Python.
    b. Return any items that contain the search term anywhere in the title of the item.
        i. For example, a search for “The Big ”, should match “The Big Lebowski”, and “The Big Short.”
        ii. A search for “The Big Le” should only return “The Big Lebowski”
    c. When the results are returned it should produce a list of titles of the items that match the query text.
    d. The page should show what the search text was. For example ‘Showing Results for “The Big ”.’
    e. You may return the search results in any order you like.
    f. If no results are returned print “No results found”.
    g. When the user does a new search, it should clear the previous results and display the new matching results.
    h. If the user issues a search that is all whitespace, do not search anything. Just clear the whitespace out of the search bar and make sure the focus is still in the search bar.
    i. It doesn’t have to be attractive or well-designed in terms of information hierarchy, it just has to function and be readable.
4. **View data.**
    a. When the user clicks on a search result or a popular item on the home page, it should take the user to a new page that shows all the details of the item.
    b. That page has the route ‘view/<id>’, where <id> is the id of the item the user wants to view.
    c. On the view page, it should use a template to display the all fields of the item except the id field.
        i. Items in an array must be iterated over and displayed.
    d. It doesn’t have to be attractive or well-designed in terms of information hierarchy, it just has to function and be readable.

Turn in your code and a ~2 minute video of you using the site. In the video, show off the functionality of your site, and narrate how the code works. Write your name in the code somewhere AND say your name at the beginning of the video

UPDATE: The video should be about 2 minutes. Anything longer than 3 minutes we won’t watch.

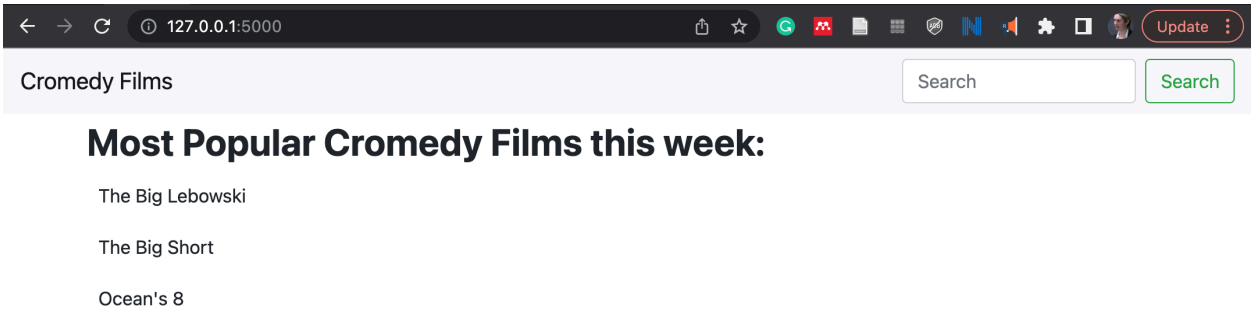
In the video, we should see:

1. The home page with three items on it.
2. Click on an item and let us see the new page showing all the information for an item.
3. From that page, use the navbar to do a search that will return multiple results
4. On the result page, click the second item so we can see view the item. Make sure we can see all the data displayed

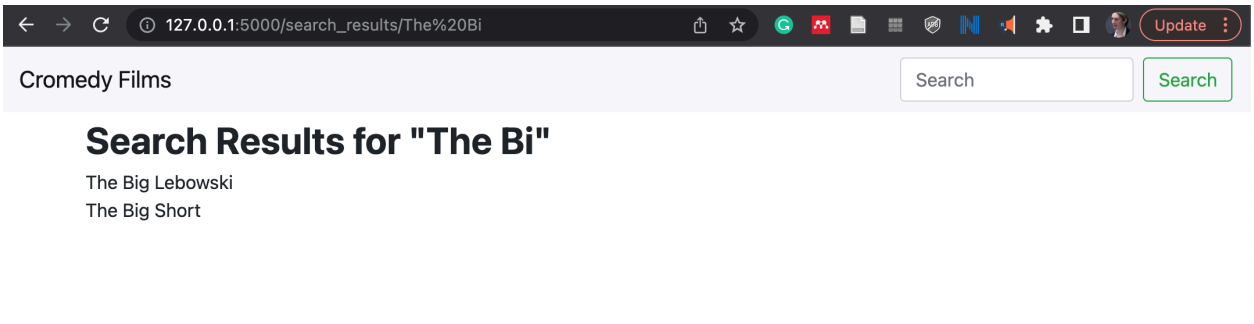
5. From that page, do a search that has zero results, make sure we can see “No results found.”
6. From that page, issue a search that is just whitespace and show that the system does not perform a search.
7. Finally, go back to the home page by clicking on the title of the website in the navbar.

Here are example screenshots from my example. You do not have to copy them, they are just here for reference.

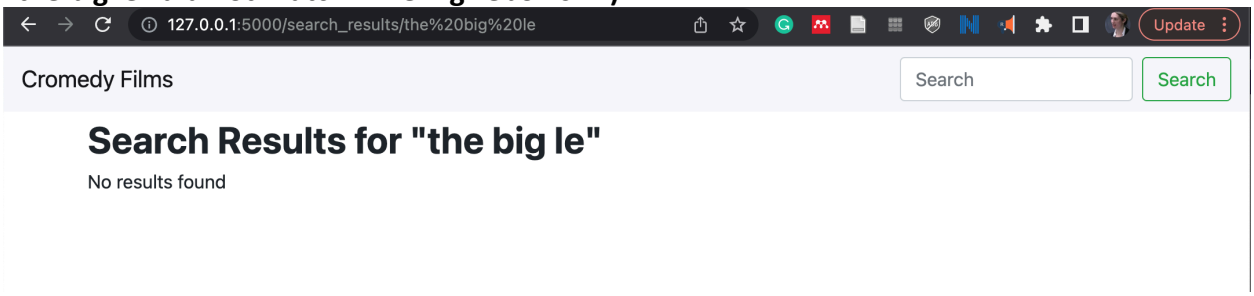
Homepage with Navbar and 3 popular items



A search that returns 2 search results



Search results when there are no matches (note, my search function was case sensitive, so “the big le” did not match “The Big Lebowski”)



Viewing an item