

Homework 7 & 8: Creating a Search Application

HW7 Due Friday 3/15 @ 4pm on Courseworks – the usual late policy applies to HW7.

HW8 Due Monday 3/25 @4pm on Courseworks – **HW8 cannot be accepted late.** (Technically it is due Friday 3/22, and we are simply not penalizing late submissions)

What to submit for Homework 7 (5pt):

- 2 uploads titled:
 - hw7.zip: A Flask project containing:
 - server.py
 - templates/ (and the HTML templates you need)
 - static/ (and any static files you need)
 - hw7_writeup.pdf

What to submit for Homework 8 (5pts):

- 3 uploads titled:
 - hw8.zip: A Flask project containing:
 - server.py
 - templates/ (and the HTML templates you need)
 - static/ (and any static files you need)
 - hw8_writeup.pdf
 - hw8.mov (let us know if want to use a different video file extension)

Submit both assignments to Courseworks.

In these two assignments you will create a webpage that lets users enter and search for data on a topic of your choice. Homework 7 will get you started, and Homework 8 will be the complete application.

You must use Flask, HTML, CSS, JQuery, Bootstrap, and JavaScript.

Your application must have 3 routes with the following route names:

- add_item
- search
- view_item/<item_id>

It must have a Bootstrap navbar loaded as a header for all three html templates in Flask.

You must pick a dataset that someone might want to search. Your “database” needs to have 30-50 items in it as a proof of concept, but you must pick a topic that could have ~1000 items if you were to build it for real.

For example, “movies” would be too broad of a topic. There are 100,000+ movies. But “1990’s comedies” is about the right size.

For each item, you will be creating a data entry:

- An image, video, or gif. You must have an external link to this. You cannot download them and server the media locally (because when we run your site, we don't want to have to download all your media)
- A short title
- A text paragraph of explanation (At least 4 sentences)
- Some sort of numerical data (a year, a price, a rating, etc)
- 4-8 other fields of your choosing

Homework 7

1. What data will your website allow users to search?

Example: "1990's comedy films."

2. What is an example of a person who would want to search this and why?

Example: *A college student is watching the Big Lebowski and want to know who directed that film, and what other films they directed to figure out what to watch next.*

3. Show up an example of one item in the database

Example:

```
{  
Id: 1,  
Title: "The Big Lebowski",  
Poster: "https://m.media-amazon.com/images/lebowski.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. Written by ahmetkozan",  
Director: ["Joel Coen", "Ethan Coen"],  
Budget: "$15,000,000",  
Stars: ["Jeff Bridges", "John Goodman", "Julianne Moore"],  
Score: 8.1,  
Genres: ["Comedy", "crime"]  
}
```

4. Build a Flask website with three routes:
 - a. Add_item
 - b. Search
 - c. Item/<item_id>
5. Each route should render an HTML template. The content of each page can be a very simple place holder (like the hello.html template from the people project.)
6. There must be a header template called layout.html
 - a. That header must contain a Bootstrap Navbar with
 - i. The name of your website
 - ii. A link called "Add Item" to add and item
 - iii. A link called "Search" to search for an item
 - iv. (Note there is no navbar link to view the item. You can only get to that from the other two pages)
7. Each route must extend layout.html so that the navbar appears at the top of every page.
8. In the write up, show us screenshots of each of the three routes.

Homework 8

1. The `add_item` page must allow the user to enter a new piece of data (with all the fields) and save it to the server.
 - a. If it is successful, it must inform the user, and give them a link to see the page for their new item.
 - b. If it is unsuccessful, it must also inform the user.
2. The search page should have an input box and a “Search” button that searches for the text across all the fields of the data elements except the id field and any fields that contain image, video, or gif URLs.
 - a. The search should match substrings (not just exact string). For example, if the user types “big”, it should match “Big” and “The Big Lebowski”, “Notorious BIG”, and “Abigail.” (Assuming those things were in some field of the database)
 - b. The results should show a vertical list of all the matching items.
 - c. Design each result to produce a short summary of the 3 or 4 more important bits of information the user can see to assess if this is the correct entry to match their search.
 - i. What fields did you select? And why? Related it back to the user, the domain and their goal.
 - ii. What is the order in which you intend people will see them in?
 - iii. For each field, what two visual design principles did you use to emphasize its importance?
3. When the user clicks on an item from search, it should display in its own page.
 - a. That page should display all the information, using good information design.
 - b. This page must not make any ajax calls. All data must be loaded from the server when it is first displayed.
4. In the write up, show us screenshots of each of the three routes.
5. Submit a short video of you using the app. (This is very easy to do in applications like QuickTime). We don’t need sound, we don’t need high production value, we just need to see the application in action. In the video show the following actions in this order:
 - a. The user should first perform a search, and see multiple results that match the search. Pick a term that shows you did substring matching, and not just exact string matching.
 - b. The user should click on one of the items and go to the item page that displays its full information
 - c. From the item page, the user should click on the “add item” option in the navbar.
 - d. The user should successfully enter a new data item, see the URL to view the item, click the link and then view the item.
 - e. Submit it as `hw8.mov`