

Homework 7: Usable CRUD Application

Due **Friday** 3/11 @ 11:59pm on Courseworks. There is a grace period until 8am the next morning.

This assignment is the equivalent of a midterm in other classes – it displays cumulative knowledge of the things learned in all the lectures in the first half of the class and it is worth 10% of your final grade. Consider this assignment a take-home midterm and treat it with the same urgency as an in-class written midterm. However, just like a regular homework, you are still allowed to Google things, come to office hours, work with others, and seek other forms of help.

If you have an emergency and request an extension on this assignment – the normal rules apply. Wednesday 3/16 at 4pm is the latest it can be considered, as the staff need ample time to grade assignments. Additionally, the staff will not hold office hours or answer questions after 3/11.

Warm up:

Design the visual information hierarchy for one data item in the view/ <id> page. Like HW1, we recommend using a PowerPoint-like application to make your design. Do not implement your design yet.

There should be at least two conceptual groups, probably three groups. On your design, indicate and annotate:

1. Where the groups are
2. What the concept behind the group is (and what goal they help the user achieve).
3. What the most important information in that group is and why.
4. What tools did you use to indicate importance and to pass the squint test.

You may edit the information you present as long as:

1. You transfer these edits to your actual data in your application across all items.
2. The data still meets the requirements (length, diversity, types) as stated in HW6.

What to submit:

1 pdf that includes three pages:

- Write a short sentence describing your participation on Wednesday (unless instructed otherwise by your section TA).
- **User info.**
 - What data will your website allow users to search? (you may copy this from HW6 or edit it)
 - Example: *"Crime Comedy Films."* (You may not use this answer)
 - What is an example of a person who would want to search this and why? (you may copy this from HW6 or edit it)
 - Example: *A college student who saw The Big Lebowski and is now fascinated by the genre of crime comedies and wants to find more to watch*
 - What information on this page should help the user issue a new search? And why?
 - Example: *The genres, directors and actors are searchable because they can help the user find similar movie they may like.*
- **Design.** An image of your design without annotation
- **Annotated design.** An image of your design with annotations indicating the visual information design we requested.

Main Assignment

What to submit:

- Hw7_<UNI>.zip: A Flask project containing:
 - server.py
 - templates/ (and the HTML templates you need)
 - static/ (and any static files you need)
- a link to a youtube video.
- An answer to problem 1.

Part 1: Functionality:

As in HW6, your application must have: a navbar on all pages, a home page, and search result page, and a view page with all the functionality we required there.

Additionally,

1. Tell us where and when you will meet your TA on Monday 3/21 Look for a slack message or email from your TA to coordinate this!
2. **Home.**
 - a. There should be a conceptual group whose goal is to indicate the purpose and the audience for the site. Consider using a title, an image, and some descriptive text.
 - b. It should show at least 3 entries database entries that should be designed in a way that entices the viewer to click on something and start exploring. (Just the title is not enough). But it should not show all the information.
 - c. The layout of the entries should be formatted using bootstrap rows and columns. You can decide how you want the layout to look.
 - d. When you click the image, it should take you to the page for viewing the item.
3. **Search.** When the user presses “go” on the search link (or presses enter), it should search for the items and return a list of all matching results.
 - a. **Flexibility.** The query must do substring matching that is not case sensitive on the title and at least **two** other text field.
 - b. **Feedback.** In addition to returning the results and the search text, the page must say how many results there are. If there are zero results, it should tell the user there are no matches found.
 - c. **Feedback.** When you present the results to the user, the text that matches the substring must be easy to scan for, according to gestalt principles and/or visual information design “tools”.
4. **Add data.** Create a new route for adding data called /add. On the template for creating a new database item, you will still have input boxes for all the fields the user must input. In addition:
 - a. **State change:** This route should only be accessible from the navbar.
 - b. **State:** On the page should be labeled fields that the user can type in to create their new entry.
 - c. **Error Detection.** When creating a new database entry, there must be error handling on all the fields. If the field must be a number, then ensure it is a

number. At the very least, you can check that the field is not blank (remember to trim the text to test if it's blank). Design the error feedback so that it directs the user's attention to the right place to correct the error.

- d. **Options and transitions.** After the user presses "submit" and the data successfully submits, allow the user to either view the item or enter a new item.
 - i. At the top of the page it should say, "New item successfully created." With a button or link that says "see it here" (or words to that effect). This links to a page for viewing the item.
 - ii. Additionally, the input boxes should clear and the focus should be placed on the first text box so the user is ready to submit another item.
 - iii. This data should be saved on the server. This data exchange must be implemented with Ajax.
5. **Edit data.** Create and /edit/<id> route.
- a. **State Change.** In the /view/<id> route, create a button that is not very salient that allows users to go to the editing route and edit the data.
 - b. **New State.** On the /edit/<id> page, have text boxes or other inputs that allows the user to edit the field. Each of the field should be pre-populated with the current value. For each field that can be edited, create a small edit icon next to it.
 - c. **Options.** There are two buttons on the page: "submit" and "discard changes."
 - d. **Transitions.**
 - i. After the users presses "submit" the data should be saved to the server, and user should be sent to the view/<id> page where they can see their changes. Do not use ajax to implement this, but do use POST data in the view/<id> route.
 - ii. **Error prevention.** After the user presses "discard changes" the page should present a dialog box widget that asks whether they are sure. If they are sure, don't save the data, but take the user back to the view/<id> page so they can see their edits have not been saved. If they aren't sure, let them keep editing.

Part 2 – Design:

6. **Accessibility.** All media must have alt text. In the video you will open the Inspect page tool of the browser and click on a few media items to and show that there is alt text.
7. **Visual information design** There are now 5 distinct states the user can be in (home, search, create, edit) In each state, there should be good information hierarchy using all the principles taught in class,
 - a. Good conceptual groups
 - b. Appropriate use of whitespace, location, size, contrast, images, and color to indicate what the most important information in each group is.
 - c. 7 tools for visually indicating importance,
 - d. information scent (good labels), and insights about color, text and gestalt. It doesn't have to be fancy, but it should pass the squint test (where the most important things is visible) and it should have good conceptual grouping – things related to each other should be close together.
8. **Color.** Your site must have
 - a. A base color that is prominent
 - b. An accent color that is less prominent, but that pops
 - c. A light grey
 - d. A dark grey.

In your video, show a user doing the following **in this order**:

1. Coming to site and clicking on a card to randomly explore a data item.
2. From that item, clicking on a link that performs another search, or take them to another item.
3. Do some searching and show the results. Make sure you show:
 - a. The search doing substring matching with the title.
 - b. The search doing substring matching on two other fields.
 - c. The search returning no matches
 - d. The search returning more than one match.
4. For one of these searches, have the user click on one of the search results to view the item
5. The user views the item. Make sure we can see all the data on the screen. Pause for a length of time that will allow someone to scan the page.
6. While viewing the item, click on something that will perform another search.
7. View another item.
8. They see something they want to edit, enter the edit state
9. After editing, press “discard”. When it asks “are you sure”, say no, and continue editing.
10. After more editing, press “submit”, and the user is taken to the view page where their edit should be visible.
11. Go to the add data item in the navbar.
12. Add a new piece of data with an error and try to submit it. Show the feedback.
13. They correct the error, then submit it. Show the feedback and new state.
14. They click the button that says something like “see it here”
15. After viewing the item, they click on the home button on the navbar. This takes them to the home screen
16. Use the inspector tools of the browser and hover over a few image elements to show there is alt text.

Your video should be short and sweet. If it’s longer than 5 minutes, you’re doing something wrong. 2-3 minutes should be plenty.

You don’t need to do voice over, but you can if you want. It usually takes me 3 tries to get a good recording where I actually remember to do all the things. Give yourself some time to do this. Usually an hour is enough.

In general, you have some creative license for what you name buttons and other bits of text on the UI. As long as it is clear from the video what the function is, it’s fine.

We prefer you use a screen recorder (Quicktime is great for this), but you can also take a video with your phone.