

# CS 328 - Homework 11

## Deadline

11:59 pm on Friday, May 2, 2025

## Purpose

To think a bit more about accessibility, and to practice with PHP using cookie-based sessions.

## How to submit

Each time you wish to submit your work-so-far, submit your files using `~st10/328submit` on nrs-projects, with a homework number of **11**.

## Important notes

- **NOTE:** you are **welcome** to use `require_once/include_once/require/include` in your PHP documents!
  - BUT when you use them in homework problems' documents, be sure to also **SUBMIT** copies of all files that you are requiring/including!
- **DO NOT USE ANY PHP FRAMEWORKS FOR THESE PROBLEMS** -- one of this course's purposes is to provide you with practice with "plain" PHP.
- Remember: You are required to use `normalize.css` for all of your web pages for CS 328, and to add `link` elements for additional CSS external stylesheets *after* this (but still **within** the **head** element).  
EXCEPT for `normalize.css`, **DO NOT USE ANY CSS FRAMEWORKS or PREDEFINED LIBRARIES for this course** (unless you get prior, explicit permission). One of this course's purposes is to provide you with some practice with the basics of "plain" CSS, so you can better make use of frameworks and predefined libraries later.
- You are expected to follow all course coding standards posted on the public course web site; course documents (including documents generated by your PHP files) are also expected to validate as "strict"-style HTML, and valid CSS.
- Make sure that you have executed the scripts `create-bks.sql` and `pop-bks.sql`, and that the bookstore tables are successfully created and populated.

## Problem 1 - more on accessibility - 16 points

We have talked in class a bit about accessibility-- to add a bit of depth to your knowledge in accessibility, consider:

- the short article "**Designing for accessibility is not that hard**" and
- the Teach Access Initiative's short tutorial on accessibility

...posted on the now-added link to "**A selection of Accessibility and Web Design references**" on the

public course web site (and also posted along with this homework handout).

Then, in a **plain-text file** whose name includes **hw11-access** and ends with the suffix **.txt**:

- include your name
- give at least **three** rules-of-thumb or guidelines from the article "Designing for accessibility is not that hard" that you want to make sure to remember as you design web sites/web applications **OR** that you wish others would remember as they do so
- give at least **three** rules-of-thumb or guidelines (**different** from those you just selected from the article above) from the posted Teach Access Initiative's short tutorial on accessibility that you want to make sure to remember as you design web sites/web applications **OR** that you wish others would remember as they do so

Submit your file **\*hw11-access\*.txt**.

## Problem 2 - practice with PHP sessions and OCI - 42 points

Recall that CS 328 class helper function **hum\_conn\_sess** is very similar to **hum\_conn\_no\_login**, except that it also destroys the current session if the attempted database connection fails.

Note that you can make a copy of **hum\_conn\_sess.php**, that contains the helper function **hum\_conn\_sess**, in your current nrs-projects directory with this command:

```
cp ~st10/hum_conn_sess.php . # remember the space and period!
```

Make a **COPY** of your file **bks.css** from **Homework 10 - Problem 1** in a **DIFFERENT** directory on nrs-projects (so you will not interfere with your earlier homework's files!).

Your Homework 11 version of **bks.css** should meet the following requirements:

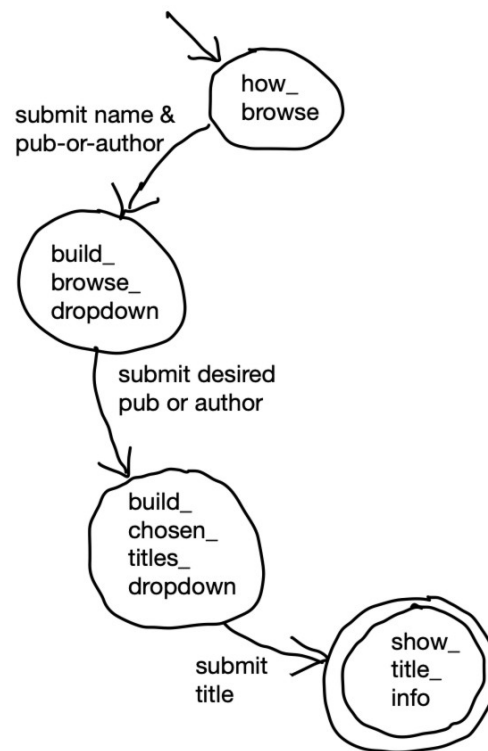
- Actually, it is fine if you decide that you do not need to make any changes to this, as long as it meets Homework 7 - Problem 5's requirements.
- But, whether it is changed or not, make sure the resulting **bks.css** still validates as valid CSS.

To practice with PHP, cookie-based sessions, and OCI, now **CHOOSE ONE** of the following options:

(Problem 2 - Option \*1\* starts on the next page.)

**Problem 2 - Option \*1\***

Consider the following finite state machine for a desired small application involving the bookstore db: database:



Create a postback PHP document **328hw11-2.php** that meets the following requirements:

- Include your name and last modified date in its opening comment, **AND** the URL this can be run from.
  - (You *will* lose some credit if this URL does not work when I or the grader paste it into a browser!)
- **Within** the **head** element, edit the **title** element, giving it appropriate content.
- **Within** the **head** element: **if** you would like, include PHP tag(s) with statement(s):
  - to enable PHP error reporting
  - to include the definition of function **hum\_conn\_sess**, from a local copy of file **hum\_conn\_sess.php**
- **Within** the **head** element, add a second **link** element so that this will be further styled using Homework 11's version of **bks.css**.
  - **Do not include any inline or internal CSS rules in your 328hw11-2.php.**
- Within the **body** element, each state's output should include an **h1** element with appropriate content that somehow includes "your" bookstore's name (from previous homeworks' **about-bks.html**).

- *Somewhere* in the **body** element, include an element that **visibly** includes **your name**.
  - (Just in case you'd like to try out using **328footer-plus-end.html** for this problem, I am *not* requiring that your name be in the **footer** element for this problem.)
- Your **328hw11-2.php**'s logic should be designed so that its response includes what is desired for exactly *one* of the states in the given finite state diagram -- its response should not include what is desired for more than one of its states.

## Problem 2 - Option \*1\* - state **how\_browse**

The response for the state **how\_browse** should include a **form** element that includes at least:

- an **action** attribute with the value:
 

```
"<?= htmlentities($_SERVER["PHP_SELF"], ENT_QUOTES) ?>"
```
- a logically-related **label** and textfield for entering a user's name
- a **fieldset** containing logically-grouped radio buttons that let the user specify whether they wish to browse by **Publisher** or by **Author**
- a submit button (an **input** element with **type="submit"**)

## Problem 2 - Option \*1\* - state **build\_browse\_dropdown**

The state **build\_browse\_dropdown** should:

- Attempt to SAFELY grab the **user's name** and **browse-by preference** from the submitted form from state **how\_browse**
- SAVE these as the values of suitable session keys in **\$\_SESSION**
- **INCLUDE** the user's entered name in some visible fashion in this state's response.
- Query either **publisher-names-and-ids** or **author-last-names**, depending on the submitted browse-by choice, *dynamically* building a **form** including at least:
  - an **action** attribute with the value:
 

```
"<?= htmlentities($_SERVER["PHP_SELF"], ENT_QUOTES) ?>"
```
  - a logically-related **label** and a **select**/drop-down element that allows the user to choose a desired publisher or author (based on the submitted browse-by choice)
  - a submit button (an **input** element with **type="submit"**)

## Problem 2 - Option \*1\* - state **build\_chosen\_titles\_dropdown**

The state **build\_chosen\_titles\_dropdown** should:

- Attempt to SAFELY grab the **chosen publisher id** or **chosen author last name** from the submitted form from state **build\_browse\_dropdown**.
- Grab the previously-entered user's name from **\$\_SESSION**.
- **INCLUDE** the user's name in some visible fashion in this state's response.
- Query **title-names-and-isbns** for the chosen publisher id or the chosen author last name,

(using a query carefully built using a **bind variable** for the submitted publisher id or author last name),

*dynamically* building a form including at least:

- an **action** attribute with the value:  
`"<?= htmlentities($_SERVER["PHP_SELF"], ENT_QUOTES) ?>"`
- a logically-related **label** and a **select**/drop-down element that allows the user to choose a desired **title** from that chosen publisher or author
- a submit button (an **input** element with **type="submit"**)

## Problem 2 - Option \*1\* - state show\_title\_info

The state **show\_title\_info** should:

- Attempt to SAFELY grab the **chosen title ISBN** from the submitted form from state **build\_chosen\_titles\_dropdown**.
- Grab the previously-entered user's name from **\$\_SESSION**.
- **INCLUDE the user's name in some visible fashion** in this state's response.
- Build a query/queries projecting the following information for the selected title (each carefully built using a **bind variable** for the submitted title ISBN):
  - the title's name
  - the last name of the title's author
  - the title's price
  - the name of the title's publisher
- Display this queried information about the selected title in a tasteful, strict-style manner.
- For more-convenient starting over, include a **hypertext** link with appropriate text that links back to your **328hw11-2.php**.

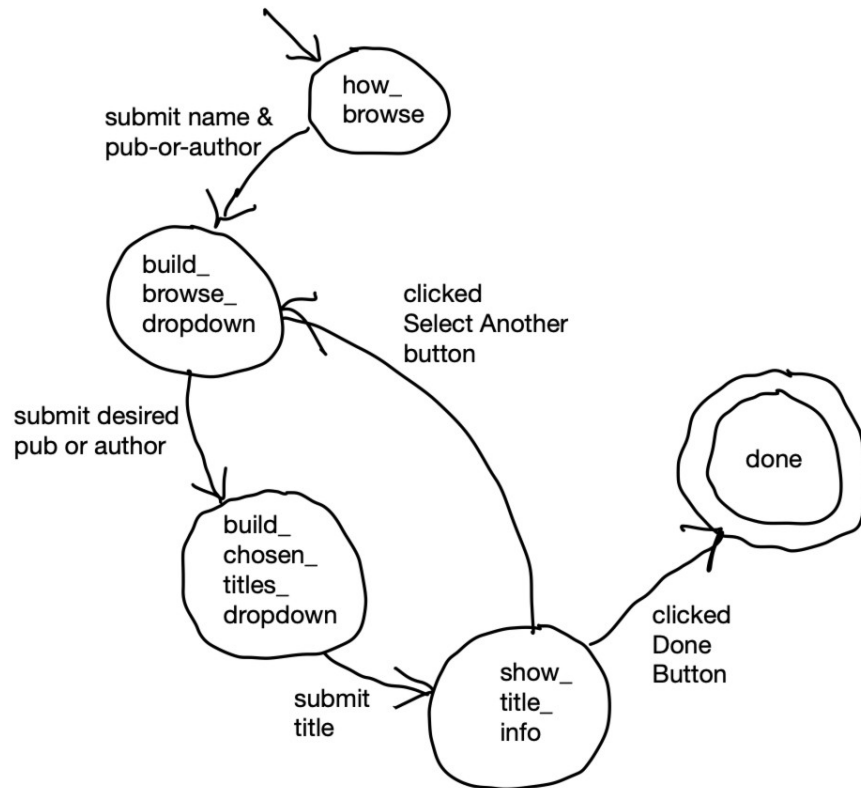
Submit your resulting files:

- **328hw11-2.php** (and all additional files it uses, if any)
- **bks.css**.

(Note: your PHP *is* expected to generate strict-style HTML, but because of the clunkiness in trying to create `.xhtml` files when sessions are involved when so many browsers re-request the document for showing the source, you are **not** required to submit `.xhtml` files for this homework problem.)

**Problem 2 - Option \*\*2\*\***

Implement this variation of Problem 2 - Option 1's state diagram that includes a "branch-back":



The states are similar to Problem 2 - Option 1's version, except state **show\_title\_info** is not the final state -- it should include a form with two submit buttons, allowing the user to specify if they are done, or if they wish to select another option from the form created by state **build\_browse\_dropdown**.

Create a postback PHP document **328hw11-2.php** that meets the following requirements:

- Include your name and last modified date in its opening comment, **AND** the URL this can be run from.
  - (You *will* lose some credit if this URL does not work when I or the grader paste it into a browser!)
- **Within** the **head** element, edit the **title** element, giving it appropriate content.
- **Within** the **head** element: **if** you would like, include PHP tag(s) with statement(s):
  - to enable PHP error reporting
  - to include the definition of function **hum\_conn\_sess**, from a local copy of file **hum\_conn\_sess.php**
- **Within** the **head** element, add a second **link** element so that this will be further styled using Homework 11's version of **bks.css**.

- **Do not include any inline or internal CSS rules in your 328hw11-2.php.**
- Within the **body** element, each state's output should include an **h1** element with appropriate content that somehow includes "your" bookstore's name (from previous homeworks' **about-bks.html**).
- *Somewhere* in the **body** element, include an element that **visibly** includes **your name**.
  - (Just in case you'd like to try out using **328footer-plus-end.html** for this problem, I am *not* requiring that your name be in the **footer** element for this problem.)
- Your **328hw11-2.php**'s logic should be designed so that its response includes what is desired for exactly *one* of the states in the given finite state diagram -- its response should not include what is desired for more than one of its states.

## Problem 2 - Option \*\*2\*\* - state **how\_browse**

The response for the state **how\_browse** should include a **form** element that includes at least:

- an **action** attribute with the value:
 

```
"<?= htmlentities($_SERVER["PHP_SELF"], ENT_QUOTES) ?>"
```
- a logically-related **label** and textfield for entering a user's name
- a **fieldset** containing logically-grouped radio buttons that let the user specify whether they wish to browse by **Publisher** or by **Author**
- a submit button (an **input** element with **type="submit"**)

## Problem 2 - Option \*\*2\*\* - state **build\_browse\_dropdown**

The state **build\_browse\_dropdown** should:

- *IF* coming from state **how\_browse**:
  - Attempt to SAFELY grab the **user's name** and **browse-by preference** from the submitted form from state **how\_browse**
  - SAVE these as the values of suitable session keys in **\$\_SESSION**

*IF* coming from state **show\_title\_info**:

- Grab the **user's name** and **browse-by preference** saved previously in **\$\_SESSION**
- **INCLUDE** the user's entered name in some visible fashion in this state's response.
- Query either **publisher-names-and-ids** or **author-last-names**, depending on the submitted browse-by choice, *dynamically* building a **form** including at least:
  - an **action** attribute with the value:
 

```
"<?= htmlentities($_SERVER["PHP_SELF"], ENT_QUOTES) ?>"
```
  - a logically-related **label** and a **select**/drop-down element that allows the user to choose a desired publisher or author (based on the submitted browse-by choice)
  - a submit button (an **input** element with **type="submit"**)

## Problem 2 - Option \*\*2\*\* - state `build_chosen_titles_dropdown`

The state `build_chosen_titles_dropdown` should:

- Attempt to SAFELY grab the **chosen publisher id** or **chosen author last name** from the submitted form from state `build_browse_dropdown`.
- Grab the previously-entered user's name from `$_SESSION`.
- **INCLUDE the user's name in some visible fashion** in this state's response.
- Query **title-names-and-isbns** from the chosen publisher id or the chosen author last name, (using a query carefully built using a **bind variable** for the submitted publisher id or author last name),

*dynamically* building a form including at least:

- an **action** attribute with the value:  

```
"<?= htmlentities($_SERVER["PHP_SELF"], ENT_QUOTES) ?>"
```
- a logically-related **label** and a **select**/drop-down element that allows the user to choose a desired **title** from that chosen publisher or author
- a submit button (an **input** element with **type="submit"**)

## Problem 2 - Option \*\*2\*\* - state `show_title_info`

The state `show_title_info` should:

- Attempt to SAFELY grab the **chosen title ISBN** from the submitted form from state `build_chosen_titles_dropdown`.
- Grab the previously-entered user's name from `$_SESSION`.
- **INCLUDE the user's name in some visible fashion** in this state's response.
- Build a query/queries projecting the following information for the selected title (each carefully built using a **bind variable** for the submitted title ISBN):
  - the title's name
  - the last name of the title's author
  - the title's price
  - the name of the title's publisher
- Display this queried information about the selected title in a tasteful, strict-style manner.
- Then include a small **form** element with **two** submit buttons (two **input** elements of **type="submit"**) to allow the user to specify if they are done, or if they wish to select another option from the form created by state `build_browse_dropdown`.

## Problem 2 - Option \*\*2\*\* - state `done`

The state `done` should:



- Grab the previously-entered user's name from `$_SESSION`.
- Display a tasteful farewell message including the previously-entered user's name in a tasteful, strict-style manner.
- For more-convenient starting over, include a **hypertext** link with appropriate text that links back to your `328hw11-2.php`.

Submit your resulting files:

- `328hw11-2.php` (and all additional files it uses, if any)
- `bks.css`.

(Note: your PHP *is* expected to generate strict-style HTML, but because of the clunkiness in trying to create `.xhtml` files when sessions are involved when so many browsers re-request the document for showing the source, you are **not** required to submit `.xhtml` files for this homework problem.)

### Problem 3 - one more application atop your selected "second" database - 42 points

Make a **COPY** of your file `custom.css` from **Homework 10** in a **DIFFERENT** directory on nrs-projects (so you will not interfere with your earlier homework's files!).

Your Homework 11 version of `custom.css` should meet the following requirements:

- Actually, it is fine if you decide that you do not need to make any changes to this, as long as it meets Homework 7 - Problem 6's requirements.
- But, whether it is changed or not, make sure the resulting `custom.css` still validates as valid CSS.

To practice with PHP, cookie-based sessions, and OCI one more time, create a postback PHP document `328hw11-3.php` involving your selected "second" database that meets the following requirements:

- Include your name and last modified date in its opening comment, **AND** the URL this can be run from.
  - (You *will* lose some credit if this URL does not work when I or the grader paste it into a browser!)
- **Within** the **head** element, edit the **title** element, giving it appropriate content.
- **Within** the **head** element: **if** you would like, include PHP tag(s) with statement(s):
  - to enable PHP error reporting
  - to include the definition of function `hum_conn_sess`, from a local copy of file `hum_conn_sess.php`
- **Within** the **head** element, add a second **link** element so that this will be further styled using Homework 11's version of `custom.css`.
  - **Do not include any inline or internal CSS rules in your `328hw11-3.php`.**

- Within the **body** element, each state's output should include an **h1** element with appropriate content.
- *Somewhere* in the **body** element, include an element that **visibly** includes **your name**.
  - (Just in case you'd like to try out using **328footer-plus-end.html** for this problem, I am *not* requiring that your name be in the **footer** element for this problem.)
- Your **328hw11-3.php**'s logic should be designed so that its response includes what is desired for exactly *one* of its states -- its response should not include what is desired for more than one of its states.
- It needs to include **at least three states**; after reading through these requirements, sketch out a finite state diagram for your **328hw11-3.php**.
  - use the same style as that used for the finite state machines for Week 13's **try-quad.php**, the Week 13 Lab Exercise's **328lab13.php**, and the two Homework 11 - Problem 2 options in this handout
  - submit a photo/scan/image of your finite state machine in a file named **328hw11-3-fsm.pdf**.
- It must use sessions, and at least one piece of information from an earlier state must be stored in **\$\_SESSION** so it can be used in a later state.
  - For example, a user name entered in state 1 could be saved in state 2 and included in the response of state 3.
  - Be sure to explicitly destroy your session at your application's logical end.
- The response for each of its states must use strict-style HTML elements appropriate for that response.
- At least *one* of its states must respond to a previous state's form using **OCI** and a statement that requires the use of **bind variables**, in generating its response.

For example:

- This could be a call to a **PL/SQL subroutine** (your PL/SQL subroutine called in Homework 10, or a different PL/SQL subroutine).
- This could be a SQL **select** statement that uses one or more bind variable (it is fine if this is the same as one you used in Homework 10 - Problem 2).
- This could be another SQL statement that uses one or more bind variables (such as an **insert**, **update**, or **delete**).
- NOTE that you will lose **substantial** credit if you use concatenation or PHP variables to include any user-submitted information within a SQL or PL/SQL statement string.
- Be sure to appropriately free your statement(s) and close your connection at the end of each state that uses OCI.
- Be sure that any **user-submitted** information is appropriately **validated-and/or-sanitized** before it is used.
- Each form generated by one of its states:
  - should have an **action** attribute whose value is:

```
"<?= htmlentities($_SERVER["PHP_SELF"], ENT_QUOTES) ?>"
```

- should include logically-related **label** elements for its (non-submit-button) form widgets
- should include a submit button (an **input** element with **type="submit"**)
- if form widget(s) are representing database values, dynamically generate the content for those widgets instead of hard-coding them (for example, query and dynamically build the **option** elements for a **select**/drop-down form widget instead of hard-coding them).
- In your final state's response, for more-convenient starting over, include a **hypertext** link with appropriate text that links back to your **328hw11-3.php**.

Submit your resulting files:

- **328hw11-3.php** (and all additional files it uses, if any)
- **328hw11-3-fsm.pdf**
- **custom.css**.

(Note: your PHP *is* expected to generate strict-style HTML, but because of the clunkiness in trying to create `.xhtml` files when sessions are involved when so many browsers re-request the document for showing the source, you are **not** required to submit `.xhtml` files for this homework problem.)