

CS 328 - Homework 11

Deadline

11:59 pm on Friday, May 1, 2026

Purpose

To get more practice with postback PHP documents using OCI along with 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.

And when you re-run these, you should also re-run:

- **Homework 3's** `new-titles.sql`, to re-add your additional titles
- **Homework 5's** `328hw5.sql`, to re-create its stored subroutines
- **Homework 6's** `328hw6.sql`, to re-create its stored function.

Problem 1 - practice with PHP sessions and OCI using the bookstore database - 45 points

Purpose: to practice more with a postback PHP document using OCI to make use of the bookstore database along with cookie-based sessions.

Task: Choose one of the two options/finite state diagrams below, and design a postback PHP document **328hw11-1.php** meeting the following requirements and the requirements for your selected option.

For either option, make a copy of the CS 328 class helper function **hum_conn_sess** in your directory for this problem:

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

Also, make a **COPY** of your file **bks.css** from **Homework 10** in this problem's 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:

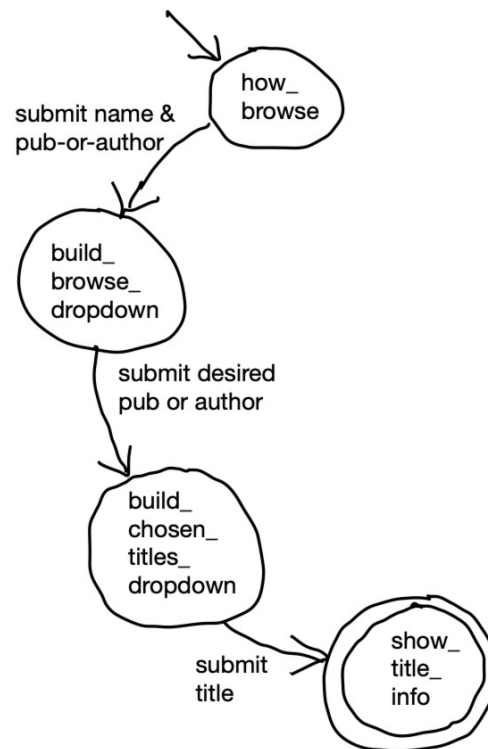
- Actually, it is fine if you decide that you do not need to make any changes to this, as long as it meets Homework 7 - Problem 2's requirements.
- But, whether it is changed or not, make sure the resulting **bks.css** still validates as valid CSS.

For either option, your postback PHP document **328hw11-1.php** should meet the following shared 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 its **head** element: include a regular PHP tag or tags:
 - to enable PHP error reporting
 - to include the definition of function **hum_conn_sess**, from your 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 328hw11-1.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**).
- Within its **footer** element near the end of the **body** element, add a **p** element whose content includes your name.
- Your **328hw11-1.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 1 - Option *1* - multiple states but no "branch-back"

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



If you choose this option, your postback PHP document **328hw11-1.php** should meet the following additional requirements:

Problem 1 - 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 1 - 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 1 - 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 1 - 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 an **a**/anchor element (hypertext link) with appropriate text that links back to your **328hw11-1.php**.
- Note that this should be the logical end of the application -- this state should destroy the session following CS 328 course style standards.

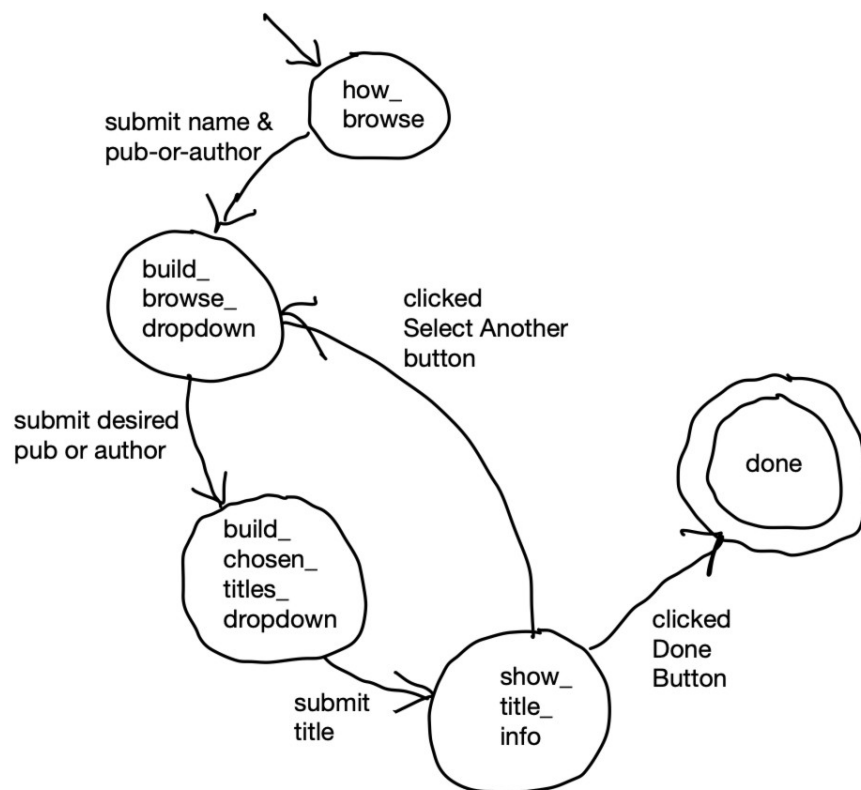
Submit your resulting files:

- **328hw11-1.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 1 - Option **2 - multiple states WITH a "branch-back"**

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



The states are similar to Problem 1 - 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**.

If you choose this option, your postback PHP document **328hw11-1.php** should meet the following additional requirements:

Problem 1 - 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 1 - 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 1 - 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 1 - 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 1 - 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 an **a**/anchor element (hypertext link) with appropriate text that links back to your **328hw11-1.php**.
- Note that this should be the logical end of the application -- this state should destroy the session

following CS 328 course style standards.

Submit your resulting files:

- **328hw11-1.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 - one more application atop your selected "second" database - 55 points

Purpose: to practice more with a postback PHP document using OCI to make use of your "second" database along with cookie-based sessions.

Task: Design a postback PHP document **328hw11-2.php** meeting the following requirements.

If your files for this problem are in a separate directory than those for Homework 11 - Problem 1, make a copy of the CS 328 class helper function **hum_conn_sess** in your directory for this problem:

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

Also, make a **COPY** of your file **custom.css** from **Homework 10** in this problem's 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 3's requirements.
- But, whether it is changed or not, make sure the resulting **custom.css** still validates as valid CSS.

Your postback PHP document **328hw11-2.php** should meet 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 its **head** element: include a regular PHP tag or tags:
 - to enable PHP error reporting
 - to include the definition of function **hum_conn_sess**, from your 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 328hw11-2.php.**
- Within the **body** element, each state's output should include an **h1** element with appropriate content.

- Within its **footer** element near the end of the **body** element, add a **p** element whose content includes your name.
- Your **328hw11-2.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 **328hw11-2.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 **328hw11-2-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, following CS 328 course style standards.
- 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 variables.
- 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 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 at least one 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 an **a**/anchor element (hypertext link) with appropriate text that links back to your **328hw11-2.php**.

Submit your resulting files:

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