Lab Homework 2: Self-Processing Sticky Forms

This lab is about practicing your problem solving skills and further developing your *programmer's toolkit*. In support of this, you'll code up a sticky form that also does some server-side input validation.

1. Learning Objectives

- Learn to work with unfamiliar code.
- Practice debugging issues. Determining where's the bug and *investigating* a solution.
- Learn how to code a self-processing sticky form.
- Provide server-side form validation.
- Further develop your programming problem solving skills.

2. Deadline

Lab Homework	Deadline	Slip Days	Credit	Solution
All Parts	Sun 2/9 at 11:59pm	Max: 2 days	10 points (completion)	Provided

1. What you need.
   - **Your textbook.** You'll probably want to use it as a reference to complete this homework.
   - **PHP Reference Documentation.**
2. Clone your lab repository.

   Clone the following URL:
   `git@github.coecis.cornell.edu:info2300-2020sp/YOUR_GITHUB_USERNAME-lab02.git`
   Replace **YOUR_GITHUB_USERNAME** in the URL with **your Cornell GitHub username**.
3. **Sign the attendance sheet before you leave**.

   It is your responsibility to sign the attendance sheet before you leave. If it wasn't handed to you, **ask to sign it**. No signature on the attendance sheet will result in an immediate 0 for this lab's attendance grade. Forgetting to sign the attendance sheet will not be considered a valid excuse to have your lab attendance corrected.
4. Work together.

   Work with your peers to complete this lab. Talk aloud, discuss, troubleshoot problems. **You are encouraged to work together so long as you do your own work and you don't give away answers.**
5. Submit.

   When you're finished, follow the instructions in **submit.md** to submit your assignment.

Part I: Peer Time

Objective: Talk with your peers around you. Talk about anything. You might ask for anything that you missed this week in lecture. Or maybe you did not quite get something in lecture, ask your peers if they can help you understand it. **We all benefit when we help and support each other.**

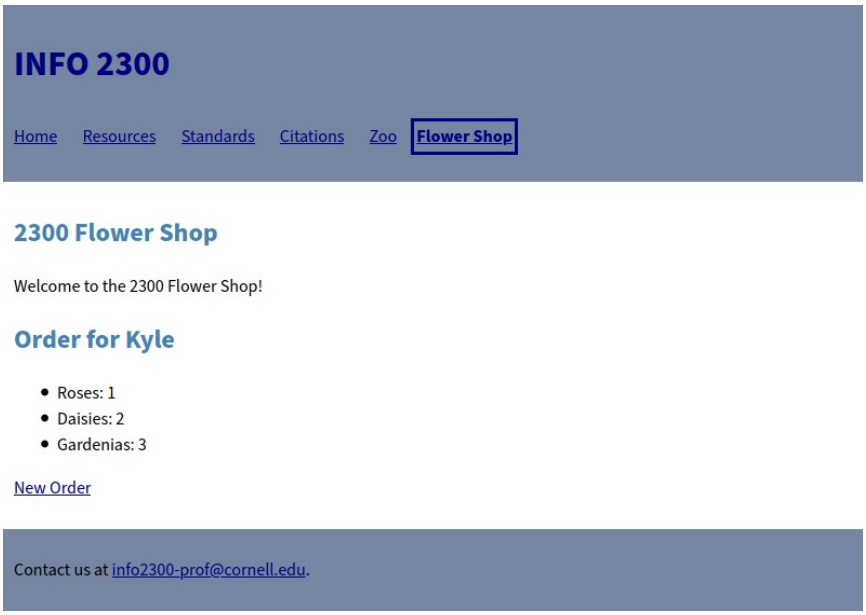
Next, consider setting up a time to complete this homework together. **This is a difficult homework.** Sitting down to do the homework together is strongly encouraged. Just remember, if you work together, you'll still need to do your own work; do not copy another student's answer.

Lastly, yes, this is a hard homework. But remember learning how to do this is hard. Take breaks if you need to. Visit office hours or seek assistance from your peers. Use your mental models or your *programmer's toolbox*. **You can do this!** Believe in yourself. If your head is saying "I can't do this," then you likely won't do it. But if it's saying **"I got this," then you likely will get it!**

1. Objective: 2300 Flower Shop's Sticky Form

Objective: Your client, 2300 Flower Shop, has asked you to help them develop a system for ordering flower stems online. Their previous programmer created the form but didn't know how to finish it. It's your task to get this form working so their customers can order flower stems online!

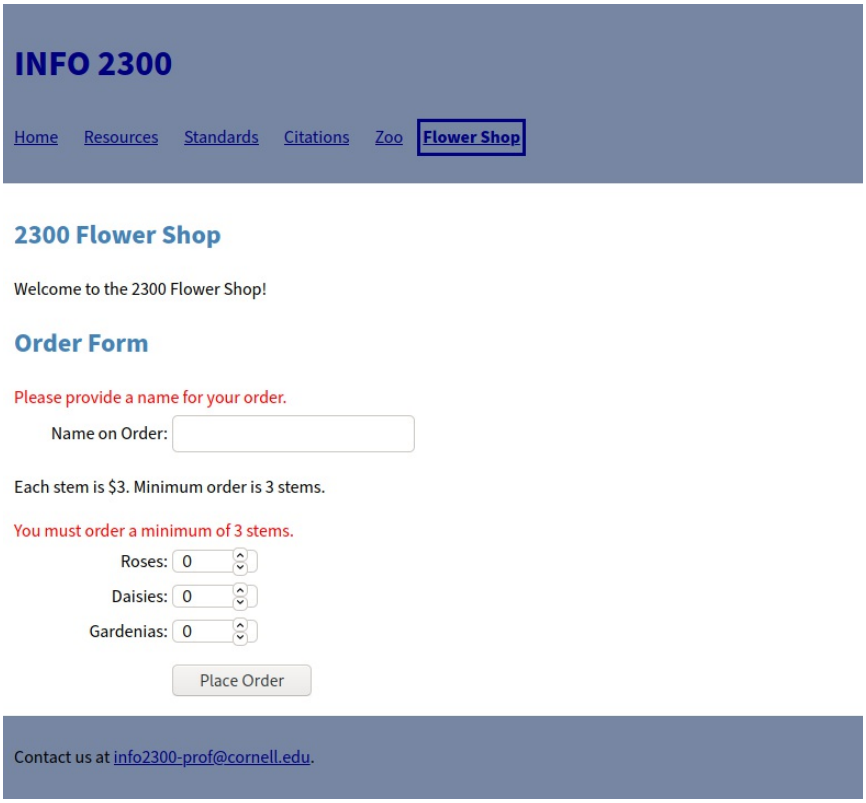
The order form should provide a good user experience for the 2300 Flower Shop's customers. Currently the form doesn't do anything when you submit it. It should go to a **confirmation page** that looks like this:



The form usability is also somewhat sketchy. **The form isn't completely sticky.** The form remembers the number of stems you ordered, but not the order's name. The form should remember all of the user inputs.

Further, **the form data isn't validated**. The order name is required and customers are required to order a minimum of 3 flower stems (not per flower variety, but rather the total number of stems regardless of the variety).

When the form **validation fails**, the **user should be presented with feedback** to correct their order. Currently the original programmer coded feedback for the order name, but it always stays hidden for some reason. There is no corrective feedback for not ordering the minimum number of stems. The feedback should look like this:



You must fix the order form using only PHP, HTML, and CSS using server-side form validation. JavaScript and HTML client-side validation is prohibited (e.g. HTML's `required` attribute).

2. Learning

This lab will help you develop your programming problem solving skills. **These are hard skills to develop.** The only way is to persevere and push through it. When you finish this lab homework you'll have developed more problem solving skills that will make finishing Project 1 easier.

Remember this, everyone learns at different rates. I promise you that if you're struggling with this assignment, so is at least 50% of the class (probably more). **Please work with your peers. Discuss ideas. Help each other troubleshoot. Just don't share code. Sharing code is cheating.**

Also remember that some students have taken other programming classes. They may get this lab done in the fraction of the time it takes you. That's okay. They've already taken other classes to develop their programming problem skills. If you're newer to programming you're still learning them. **It will take you longer and that's totally normal.**

I've been in your shoes before. I remember how frustrating this feels. Learning is hard. Programming is hard. But the reward is so worth it. Keep with it. **You can do this!**

Part II: Sticky Order Name

For some reason when a user clicks the submit button nothing seems to happen. The form just seems to reload.

It's rare in industry that you'll get to write code from scratch. Instead you almost always have to work on other people's code. Working others' code requires that you break out your investigative cap and start debugging their code. Your objective is to first determine what's wrong. Once you have a hypothesis about what's wrong, you can start planning and coding up a fix.

Objective: Try to figure out why when the form's submit button is clicked, it doesn't remember the order name. Once you figure out what's wrong, fix the code!

Hint: Pay special attention to variables at the top of flowershop.php. You'll need at least one of these variables.

Hint: Try using temporary debug messages in your code: `echo("form submitted");`, `var_dump($_GET['order_name'])`.

Hint: Don't forget to sanitize your output using: `htmlspecialchars()`.

Part III: Order Name Validation

Objective: The order name is *required*. It should never be empty (`""` or `''`). Check the order name to see if it's the empty string. If the order name is the empty string you need to mark the form as invalid.

Before you start, make sure you uncomment the `$is_order_valid` variable declaration like this:

```
// TODO: Assume the order is valid (uncomment line below).
// $is_order_valid = TRUE;
```

Next, write your order name validation code here:

```
// Name is required.
$order_name = trim($_POST['order_name']);
// TODO: Check if name is not empty, if it is set the order invalid and show the name feedback.
```

Hint: Make a plan before you change any code.

Hint: Use temporary `var_dump` statements to check the value of `$is_order_valid`.

Hint: `empty()`

Part IV: Order Name Corrective Feedback

Objective: The feedback message for a missing order name already exists in the code. However, it's always hidden. Try to figure out how to show this message when order name isn't valid and to hide it when it is valid.

Hint: `$show_name_feedback`

Part V: Minimum Number of Stems Validation

Objective: Every order has a minimum of 3 stems. It doesn't matter which variety of flower. Validate the number of stems to make sure the total is greater than or equal to 3.

Place your stem validation code here:

```
// Check that a minimum of 3 stems was ordered.
// TODO: check that the total stems ordered is less than 3, if it is, set order invalid, and show
stems feedback.
```

Hint: Check your work by temporarily var_dump'ing `$is_order_valid` and `$show_stems_feedback`.

Part VI: Minimum Order Corrective Feedback

Objective: Provide corrective feedback for orders less than 3 stems.

Hint: You'll need to take a look at the figure above for how the feedback should appear. Model the stems feedback after the name feedback.

Part VII: Order Confirmation

Objective: If the form validates, show the order confirmation. The confirmation should look like the figure above.

Place your confirmation code here:

```
<!-- TODO: order confirmation -->
```

Hint: Don't over think this. You just need to echo some values.

Part VIII: Test It

Objective: Test everything. The sticky form components. The feedback messages. The confirmation page.

Your customer should be able to trust your work. It's easy to make mistakes in programming. That's why it's important to test all of your code.

Contributors

The following individuals made contributions to this assignment.

- Kyle Harms
- Sharon Jeong