University of Washington, CSE 190 M, Spring 2009
Section 4: Complaint Letter Generator (Tuesday, April 21)
Basic PHP

section idea and code by Stefanie Hatcher; contributions by Brian Le

This section is about HTML forms and processing form data in PHP.

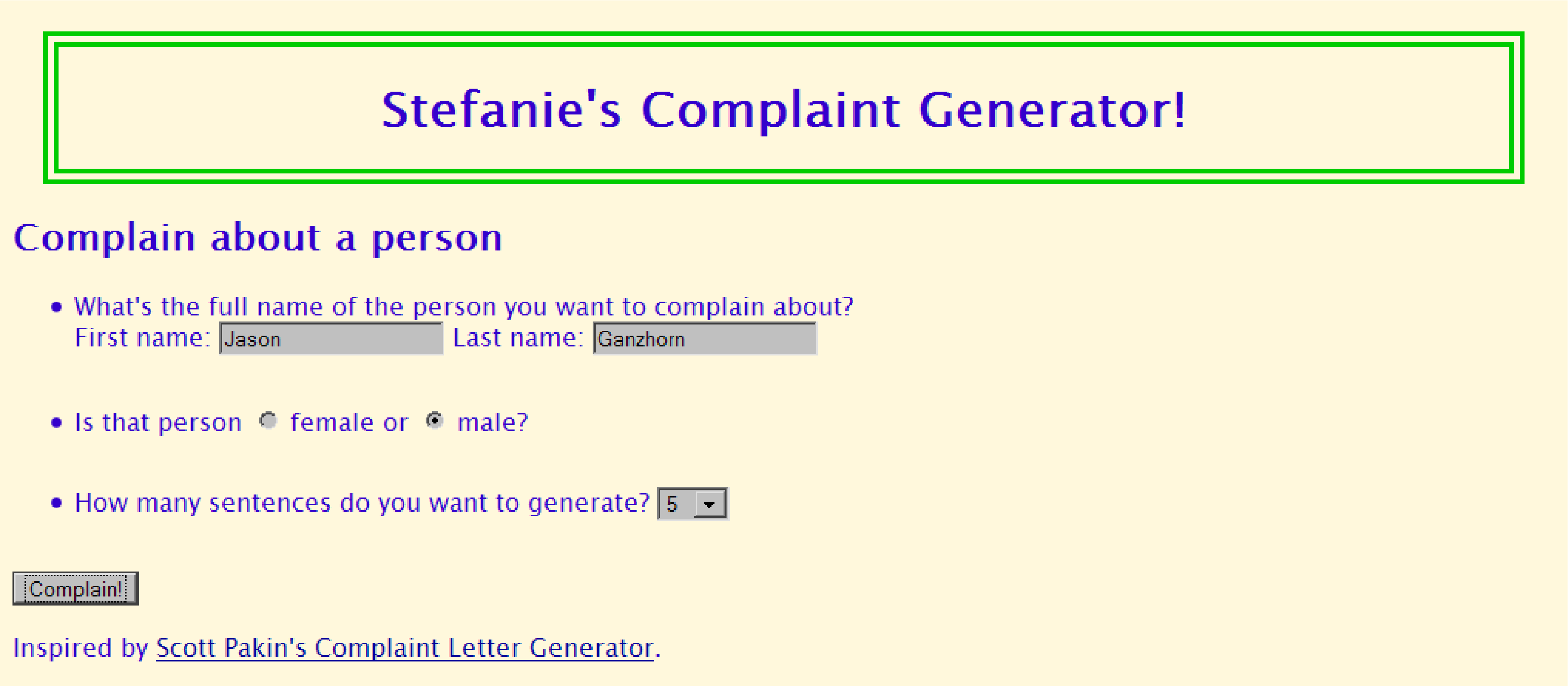
Buggy HTML form:

The following HTML form has several mistakes that causes it not to submit its data correctly, as well as some poor design choices that make it unpleasant to the user. Look at the form, find the bug(s), and correct the problems.

1. buggy form page

Complaint Letter Generator (inspired by Scott Pakin's Complaint Generator):

Write an HTML form that allows the user to generate complaint letters. The user will specify the first/last name of the person to complain about, the person's gender, and how many sentences of complaints to generate. The following should be the appearance of your form:



Stefanie's Complaint Generator!

Complain about a person

- What's the full name of the person you want to complain about?
  First name: Jason Last name: Ganzhorn
- Is that person ○ female or ● male?
- How many sentences do you want to generate? 5

Inspired by Scott Pakin's Complaint Letter Generator.

Start from the following skeleton:

- complaint.html

Test your form by having it initially submit its data to the following URL:

```
http://webster.cs.washington.edu/params.php
```

Once your form submits properly, write a PHP page `letter.php` on the server to process this form data. Start from the following skeleton:

- letter.php

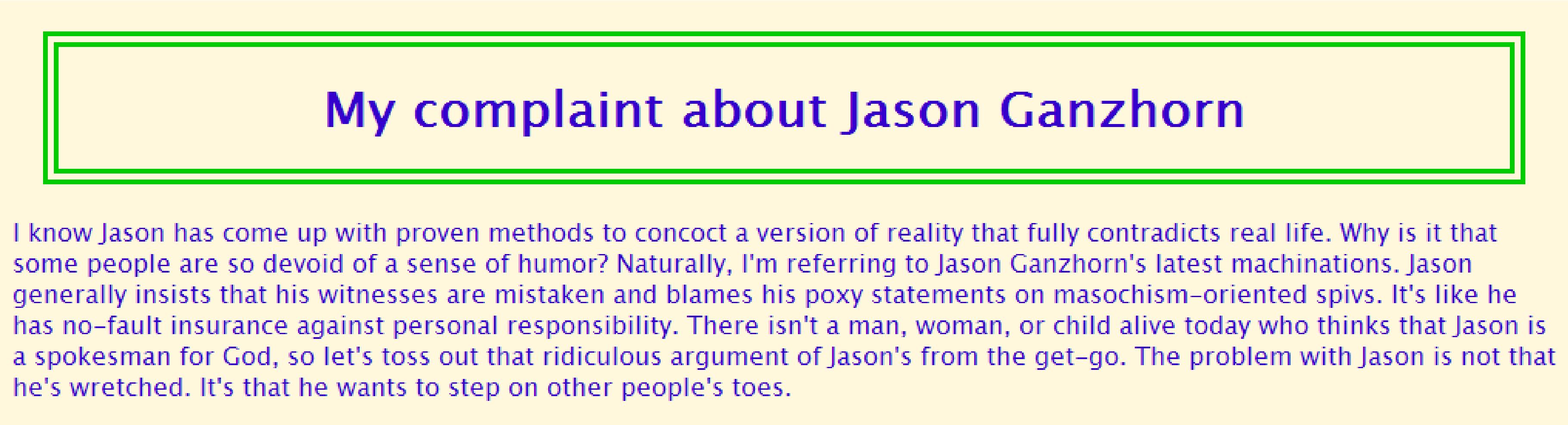
On the server there is a file `sentence.txt` containing a bunch of complaint sentences. Your PHP code should read this file, randomly pick sentences from it, and turn these into a complaint letter. The file has one sentence per line, such as:

```
_FNAME_ only wants to dominate or intimidate others.
It's knowledge that _HESHE_ recently claimed that truth is merely a social construct.
No matter what terms are used, _NAME_ is unconstrained by conscience.
```

You will need to personalize the letter by inserting the person's name and other information into it. The following are the patterns you will need to replace:

- `_FNAME_` : person's first name
- `_LNAME_` : person's last name
- `_NAME_` : person's full name
- `_HESHE_` : he (male) or she (female)
- `_HIMHER_` : him (male) or her (female)
- `_HISHER_` : his (male) or her (female)

You can use the `str_replace` function to help you replace the above patterns in the text. If you write the code correctly, you can do this with a single call. When you're finished with your page, it should look like the following:



My complaint about Jason Ganzhorn

I know Jason has come up with proven methods to concoct a version of reality that fully contradicts real life. Why is it that some people are so devoid of a sense of humor? Naturally, I'm referring to Jason Ganzhorn's latest machinations. Jason generally insists that his witnesses are mistaken and blames his poxy statements on masochism-oriented spivs. It's like he has no-fault insurance against personal responsibility. There isn't a man, woman, or child alive today who thinks that Jason is a spokesman for God, so let's toss out that ridiculous argument of Jason's from the get-go. The problem with Jason is not that he's wretched. It's that he wants to step on other people's toes.

Your finished code might look like the following sample solution, written by TA Stefanie Hatcher:

- complaint.html | letter.php
- Runnable solution

Basic Validation:

If you finish all of the above work and have time left, try to add some basic server-side validation to your form. If the user submits an empty first or last name, you should print an error message rather than creating a complaint letter.