

University of Washington, CSE 190 M, Spring 2007 Homework Assignment 7: Online Dating

due Tuesday, May 22, 2007, 11:59pm electronically

This assignment asks you to use HTML forms and PHP. You will write the following pages:

The screenshot displays the NerdLuv.com website. On the left, there is a form titled "About Me:" with the following fields: "Name:" (Marty Stepp), "Gender:" (radio buttons for Male and Female, with Female selected), "Seeking:" (checkboxes for Male and Female, with Female checked), "Keirsey Personality Type (Don't know your type?):" (ISTJ), and "Favorite Operating System:" (Windows). Below the form is a "Sign Up" button. On the right, there is a section titled "Marty Stepp's Matches" with two entries: "Black Canary" (with a small image of a blonde woman) and "Grace Hopper" (with a small image of a woman in a military uniform). Each match entry lists "seeking:", "type:", "OS:", and "strength of match:". At the bottom of the page, there are logos for W3C XHTML 1.0 and W3C CSS.

Online dating has become mainstream with popular sites such as [eHarmony](#), [match.com](#), [OkCupid](#), and [Plenty of Fish](#). Your task for this assignment is to write the HTML and PHP code for a fictional online dating web site called NerdLuv.com. The site will consist of two pages: a front page called `dating.html` that contains a form for a user to apply, and a result page called `dating_server.php` that displays single people who match well with the applicant.

You do not need to submit any `.js` file, since by default this assignment needs no client-side scripting. Turn in the following files:

- `dating.html`, the HTML code for the front signup page
- `dating_style.css`, the CSS styles for both pages
- `dating_server.php`, the PHP code for the results page

Part of your grade will come from posting your program to *exactly* the following URL:

`https://students.washington.edu/your_UWnetID/cse190m/homework/7/dating.html`

Signup Page Appearance:

The images referred to in this document exist in the following folder:

http://www.cs.washington.edu/education/courses/cse190m/CurrentQtr/homework/7-online_dating/images/

The `dating.html` page begins with the NerdLuv logo image, stored in `nerdluv.png` on the course web site. The text "where meek geeks meet" appears under the image. 2em below this text is an HTML form with the following elements:

- Name: A label and a 15-letter-wide text input box allowing the user to type his/her name. Initially this box is empty.
- Gender: A label and two radio buttons allowing the user to select his/her gender of Male or Female. Initially the Female button is checked.
- Seeking: A label and two checkboxes allowing the user to select the genders he/she is interested in meeting. Initially only the Male box is checked. It is legal for an applicant to check the box for his/her own gender or to check both boxes.
- Keirse Personality Type: A label and a 4-letter-wide text box allowing the user to type his/her Keirse personality type, such as ISTJ or ENFP. The label also contains a link to a site with a survey for the applicant to find out his/her type if it isn't known. The link goes to <http://www.humanmetrics.com/cgi-win/JTypes2.asp> . Initially the personality type text box is empty.
- Favorite Operating System: A label and a drop-down selection box allowing the user to select his/her computer's operating system. The choices are the following, sorted into two groups:
 - Popular OSes
 - Windows
 - Mac OS X
 - Linux
 - Not-so-popular OSes
 - BSD
 - Solaris

1em below this is a Sign Up button that, when pressed, submits the form to your `dating_server.php` script for processing. This button has a background color of `FFAAAA`, and its text is bold and 1.5 times as large as normal text on the page. (See the 14-forms slides about styling form elements with CSS attribute selectors.) Use a POST request to submit your form.

The overall set of form fields is 30em wide and has a background color of `E0E0FF` and a 1px black border. The field set has an overall label of `About Me:` with a white background and a 1px black border. The default font for all text on the page is 12pt in Century Gothic, Arial, or any sans-serif font available on the system.

Both pages have a title of `NerdLuv.com`. Both pages have a "favorites icon" of `heart.gif`, which is located on the course web site. At the bottom right of both pages is the standard pair of W3C validator buttons.

Results Page Appearance:

The `dating_server.php` receives input from `dating.html` and should output an HTML page. The page begins with the same `nerdluv.png` banner image and slogan.

2em below the banner slogan should be a level 1 heading saying "*APPLICANT NAME*'s Matches" where *APPLICANT NAME* is the name of the person who just submitted the form. Below this is a list of singles that "match" the applicant. A "match" is a single with the following qualities:

- A gender that matches the "seeking" gender(s) of the applicant.
- A "seeking" value that matches the gender of the applicant.
- A "strength of match" score of at least 2. A single scores points for the following:
 - One point for having a dimension of personality type that matches the same dimension for the applicant. For example, if a single has type ISTP and the applicant has type ESTJ, the single's "strength of match" will go up by 2 points because the S and T match.
 - Two points for matching the favorite operating system. For example, if the single prefers Mac OS X and the applicant also prefers Mac OS X, that single's "strength of match" goes up by 2.

Each matching single should have his/her name displayed on a line with a background color of `E0E0FF`. Below this is an image of the single, shown with a width of 150px. 1em to the right of this image is text about the single, including what he/she is seeking, the personality type, OS, and strength of match (explained below).

If there are no matches, the behavior is up to you. You can show the same page is shown but without any matches visible, or you can show a "No matches found" message.

Screenshots in this document are from Windows XP in Firefox 2.0, which may differ from your system.

Data:

Your `dating_server.php` page reads its data about singles from a file named `singles.txt`, which is placed in the same folder as the script itself. This file contains data records in the following format:

```
Marty Stepp,M,F,ISTJ,Windows
Ada Lovelace,F,M,ISTJ,Linux
Donald Knuth,M,MF,INTJ,BSD
```

Each time the script is run, it should read this file's contents to find the singles that match the applicant based on the above criteria. The applicant's submitted data doesn't need to be added to this file; the file is only read and does not need to be modified (unless you do an extra feature described later). See the 15-php slides about reading files and matching/splitting strings using regular expressions.

The image for each single is stored in a file in the `images` subfolder relative to the PHP script. The image for a single is a file with the `.jpg` extension whose name is equal to the person's name in lowercase with all non-alphabetic characters converted into underscores. For example, the image for Deanna Troi is stored in `deanna_troi.jpg`.

Hints and Suggestions:

Write your HTML/CSS code first and your PHP last. Test your HTML by submitting it to `http://faculty.washington.edu/stepp/params.php`.

Extra Features / Creativity Aspect:

To encourage you to be creative on this assignment, we will award you **+1 free late day** (up to a maximum of +2) for each of the following possible additions to your page. You are not required to do any of these.

- **server-side form validation:** You can add PHP code that tests all query parameters received for validity. Use regular expressions to do this. Specifically:
 - The name must not be blank.
 - The gender submitted must be male or female. (No other values such as "robot" are allowed.)
 - The "seeking" gender submitted must be male, female, or both. (It is not valid to leave both blank or to submit other values.)
 - The Keirsey personality type must be a 4-letter string whose letters come from the Keirsey personality dimensions: I or E for the 1st letter, N or S 2nd, F or T 3rd, and J or P 4th.
 - The favorite operating system must come from the choices provided.

If some of the form data is invalid, instead of showing matches, the response page should contain an error message displaying what was wrong. For example, if neither of the "seeking" checkboxes are checked, a suitable error message would be, "Error: You forgot to submit a value for seeking." If the user typed an invalid of "hello" as the Keirsey personality type, a suitable error message would be, "Error: invalid value for type."

- **client-side form validation:** Add Javascript code that tests all query parameters for validity using regular expressions before the form is sent to the server, using the same criteria named above. If any form data is invalid, cancel the form submission, show a message alerting the user about what was wrong, and highlight the invalid input field with a background color so that the user sees the error.
- **persistent applicant data with photos:** Have your program save the applicant's data into the `singles.txt` file as it is submitted. You don't have to worry about duplicates; you can simply append the data to the end and allow the same name to appear twice in the file.
For the applicant to fit in with the other singles, they'll need to send a photo. To handle uploading of photos, add a file input box to the `dating.html` page that allows the user to browse for a file to upload as a photo. When the `dating_server.php` script receives this photo, put it in the `images` folder with the others and show it when subsequent users search for matching singles.
- **Other:** If you have an idea for a creative addition to this program, please ask us and we may approve it for late day credit. (Extra features that haven't been approved by us will not receive credit.)

Note that regardless of how many of these additions you choose to implement, your main program behavior and appearance should still work as specified. You also may not submit the assignment more than 3 days past its listed due date.

Submission and Grading:

Submit your assignment online from the turnin area of the course web site. You don't need to ZIP it; turn in the `.html`, `.css`, and `.php` files. For reference, our solution has 92 unique lines of HTML and a 56-line PHP block.

Your PHP code should follow reasonable stylistic guidelines similar to those you would follow on a CSE 14x programming assignment. In particular, you should minimize redundancy, follow proper procedural decomposition, correctly use indentation and spacing, and place a comment header at the top of your file and atop every function explaining that function's behavior. You should only use material that has been discussed through the lecture on which this program was assigned, unless given explicit permission from the instructor.

Please do not place a solution to this assignment online on a publicly accessible (un-passworded) web site.

