

COM3780 Homework Assignment 4:

NerdLuv EXTRA FEATURES

The following are **two extra features** you can complete to add new functionality to your program. These are **required** for the YU version of this assignment.

Extra Feature #1. Robust page with form validation:

If the user submits invalid data that doesn't match the above criteria to [signup-submit.php](#), or if the user submits an empty name to [matches-submit.php](#), do not show any matches or sign up the user. Instead display an error message of your choice, such as the one shown at left.

Add PHP code in [signup-submit.php](#) and [matches-submit.php](#) that tests all query parameters submitted for validity. Use PHP **regular expressions** to do this. Specifically, you must check the following aspects of each query parameter that is submitted using proper strict regular expressions that allow only these patterns:

- The **name** must not be blank (both pages).
- The **age** submitted must be an integer and must be between 0 and 99 inclusive.
- The **gender** submitted must be male or female. (No other values such as "robot" are allowed.)
- 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.
- The **seeking min/max ages** submitted must be integers, must be between 0 and 99 inclusive, and the minimum seeking age must be less than or equal to the maximum.

Also make your pages robust against the following simple basic input errors:

- Re-submitting to [signup-submit.php](#) a person who is already in the file.
- Trying to view matches on [matches-submit.php](#) for a person who isn't in the file.

Note: It may help you to know that PHP has a function `exit` that you can call to immediately stop the currently running script. This function can be useful if you encounter an error and don't want to display any more of the HTML that is to come later in your code. You can display an appropriate error, end the page, and `exit` immediately. See <http://php.net/exit> for more information about this function.

nerdLuvtm
where meek geeks meet

Error! Invalid data.

We're sorry. You submitted invalid user information. Please go back and try again.

This page is for single nerds to meet and date each other! Type in your personal information and wait for the nerdy luv to begin! Thank you for using our site.

Results and page (C) Copyright NerdLuv Inc.

 [Back to front page](#)



Extra feature #2. User photos:

Modify the `matches-submit.php` page to display photos for each user.

There is a set of images in the following folder on the course web site for every user that is initially in the input file:

<http://ycfac197.mc.yu.edu:8080/COM3780/homework/4/images/>

Matches for Bjarne Stroustrup



Leia Organa
gender: F
age: 53
type: ISFJ
OS: Windows



Mary Lou Jepsen
gender: F
age: 45
type: ISTJ
OS: Windows



Roberta Williams
gender: F
age: 57
type: ENFP
OS: Windows

When the user views his/her matches, instead of displaying the default image of `user.jpg`, instead display the proper image for each individual user, as shown at left. (See the course web site for screenshots.) The file names exactly match the users' names, except that the file names are all lowercase and with spaces replaced by dashes. For example, there are image files named `marty_stepp.jpg` and `angry_video_game_nerd.jpg`.

But the system won't have photos for new users that sign up. So you should modify `signup.php` to allow new applicants to submit photos. To do uploading of photos, add a file input box to `signup.php` that allows the user to browse for a file to upload. Use an `input` tag with `type` of `file`. (See screenshot below.) When `signup-submit.php` receives a POST, it will save this photo onto the server into a subdirectory named `images/` using the naming system described (such as `dick_cheney.jpg`) and show it when subsequent users search for matches. You may assume that every user submits a valid JPEG image file and that the `images/` folder already exists.

New User Signup:

Name:	<input type="text"/>
Gender:	<input type="radio"/> Male <input checked="" type="radio"/> Female
Age:	<input type="text"/>
Personality type:	<input type="text"/> (Don't know your type?)
Favorite OS:	<input type="text" value="Windows"/>
Seeking age:	<input type="text"/> to <input type="text"/>
Photo:	<input type="text"/> <input type="button" value="Browse_"/>
<input type="button" value="Sign Up"/>	

You will also need to make one small modification to your code for `matches-submit.php`. When displaying images for each user, some of the images will come from the web URL above (the initially provided users), and some will come from your own local server (any newly signed-up users). To be able to tell which is which, you should insert a test in your code for each match. Test whether the image file exists on the server in the `images/` directory; if so, display it from there. If not, assume that it exists on the course web site in the previously named directory and display it from there.

Note: When saving a file into a directory like `images/`, you may get permission issues. Make sure that your `images/` folder on the server has both "Write" and "Execute" permissions enabled for all users. You may also want to add the line below to your code after saving any user's new image, so that when you save a new image file to `ycfac197`, it is given full permission so that you can move/delete it. Without this line, only the `apache` user that PHP runs as will be able to delete any newly added user image files. This is not a good situation.

```
chmod("images/WHATEVER_IMAGE_FILE.jpg", 0777); # full permission to all users
```