

Project 2

This project incorporates conditionals, PHP validation, HTML elements, and topics covered from **Chapters 1 - 10** in your textbook.

In this assignment, you will create a "sticky" registration form that allows users to register their dog for a free class and have a confirmation email sent to the email address that the user specified within the form. You will build one page that contains the form for the user fill out and the php script. I will not be providing step-by-step instructions but rather some basic requirements. You will use the information within the textbook, the sample screen shots for the verbiage, and the skills and knowledge you have acquired in this course to date to complete this project. Please see the **Project 2 Screenshots PDF** document for reference.

Let's get started!

Section 1.0 - Getting Started

Begin by downloading the project zip file that contains the files that you will need for this project. The zip file contains the following: one PHP starter file, an images folder with one png image file, and a CSS file.

1. Download the **project2_student_files.zip** file and extract it.
2. Rename the folder to **project2**.
3. Open **register.php** within your editor and add a title of **Class Registration** within the <title> tag.
4. **Link** the external CSS file to register.php.
5. Jump down to the <footer> tag area and hyperlink the **Course Homepage** text to link to your
6. course homepage.
 - a. Remember to include the title attribute within the anchor tag that reads **Course Homepage**.
7. Save your work.

Section 1.1 – Form Requirements

I recommend building the form first. The form should be "sticky" so reviewing the information in Chapter 8 starting on page 220 will be very useful. Use the screenshots within the **Project 2 Screenshots PDF** as a reference for what your completed form should look like.

HINT: Review Page 230.

The form should contain the following:

1. The form tag should include the following:
 - a. An action attribute referencing register.php
 - b. A post method
2. A text input field for the **Dog's First Name**. Wrap the text **Dog's First Name** within a **label** tag.
 - a. Include the autofocus Boolean attribute.
 - b. Add a size of 20

- c. Name the field whatever you wish just be consistent otherwise your script won't work. For example, I named the field **first_name**.
 - d. For the value, add the code that makes it sticky, see page 221 as a reference.
3. A text input field for **Your Last Name**. Wrap the text **Your Last Name** within a **label** tag and add the appropriate attributes: size, name, and sticky value code.
4. An email input field for **Your Email Address**. Wrap the text **Your Email Address** within a **label** tag and add the appropriate attributes: size, name, and sticky value code.
 - a. Be sure to use the email type.
 - b. You can adjust the size of the input field if you wish.
5. A text input field for **Dog Breed**. Wrap the text **Dog Breed** within a **label** tag and add the appropriate attributes: size, name, and sticky value code.
6. A text input field for **Phone Number**. Wrap the text **Phone Number** within a **label** tag and add the appropriate attributes: size, name, and sticky value code.
7. Add a submit button that reads **Register!**.
8. Save your work.

Section 1.2 – Building the PHP Script

Now it's your turn to do the PHP scripting behind gathering, validating, and emailing the form data. Please use the information from Chapters 1 - 10 to accomplish the coding for the script. The Chapter 8 information will be very useful. Note that you can use either **echo** or **print** within your script. **PHP** should be used to display the content below. Bold where appropriate.

1. Add the PHP delimiters inside where the comments read **Start** and **End PHP script**.
2. Use PHP comments to describe the purpose of the script.
3. Add the comment **Print some introductory text & image:** and using either **echo** or **print**, add the following:
 - a. A **heading 1 tag** that reads Registration Form
 - b. An **image tag** that adds the dogs.jpg file to the page (don't forget to add the responsive class to the image tag as well as the alt and title attributes).
 - c. A paragraph of introductory text that reads:

Register for our **FREE** "Your Healthy Dog" class! We will have presenters from local pet supply stores, healthy dog treats and food, supplements to support your dog's immune system, and healthy treats for humans too!

4. Another paragraph that reads:

Register below to join in on the fun and we will be contacting you via email with the date and time.

5. A heading 2 tag that reads **Keep Wagging!**.
6. Add a PHP comment that reads **Check if the form has been submitted:**

Use pages 228 – 229 as a reference to complete the following next steps. The **if conditional** should be used within your script.

7. Add code that checks to see if the form was submitted.
8. Add a PHP comment that reads **Check for each value.**
9. Perform validation for all of the form fields and if the fields are not filled out or contain correct values, display an error message on screen.
 - a. Be sure to add the error **class** to the error message print statement so it displays in red.
10. Be sure you that validation for the email address looks to see if it's formatted correctly.
11. For validating the **phone** field, the field should not be empty and the phone number should include numbers only.
 - a. This may be helpful: `!= null && !is_numeric($_POST['phone'])`
12. Add the PHP comment **If there weren't any problems...**
13. Print a **success** message that reads (the dog's name should display where it's blurred out below):

You are now registered [blurred] ! Please check your hooman's email for your Registration Confirmation.

- a. Use the **ucfirst()** function to capitalize the Dog's first name.
 - b. You will need to concatenate text strings.
14. Add a PHP comment that reads **Send the email:**
15. Using **page 229** as an example include the code within your script that send an email, the **\$body** should read (include the Dog's first name):

Thank you [blurred], for registering for the FREE 'Your Healthy Dog' class! We will see and your hooman soon! We will be contacting you with the date & time of our class. Keep Wagging!

- a. Include the **mail()** function that sends the email from your email address (see the bottom of page 229 as a reference).
16. Add a PHP comment that reads **Clear the posted values:**
17. Using the code at the bottom of page 229 and top of page 230, complete the script.
18. Save your work.
19. Upload your work to the web server and test your form, the validation, and to see if an email is sent.

Section 1.3 – Error(s) and Help

Please use the **Help** forum for assistance within Canvas. All errors and/or issues should be posted to the forums so the entire class can participate and offer assistance and solutions for correcting issues. Your classmates are a valuable learning resource.

Section 1.4 - Update & Submit for Grading

Please note that you always need to notify me that your work is ready to be graded through the appropriate assignment area in Canvas each week. Missing notifications will not receive credit.

1. Log in to the Canvas classroom.
 - a. Click on the **Home** tab in the side navigation bar.
 - b. Click on **Module 10** and locate the **Module 10 – Project 2 Submission** link.
 - c. In the comments area let me know you are finished and your assignment is ready for grading, "my work is ready for grading." will do just fine.
 - d. **Supply me with your course homepage URL, e.g.,**
<http://youruserID.macombserver.net/itwp2750/home.htm>
 - e. Please note that you will not be attaching your completed assignment file(s) to the assignment area but rather uploading it to the web server where I will access it. It is not necessary to attach any files. **The file attachment option has been disabled since you will be uploading your work to the student web server.**
 - f. Once you have submitted notification and your work has been graded, you cannot resubmit for a better grade, the original grade remains as is.

Please be sure to test the links to your work. Test access to your assignment from your course homepage to be sure everything is correctly hyperlinked and in working order. If I cannot access your work, I cannot grade it. The result will be a zero for the assignment.

Please review the Weekly Schedule for other required assignment(s) due for this module.

*Grades will be assigned within the limits of the grading policy. Points listed within the Value column of the rubric below are the maximum number of points you can receive if your work is 100% correct. Incorrect work may receive zero through the maximum point value range listed within the Value column. **Late work is not accepted. The appropriate assignment area(s) will be closed at the time of assignment's due date. Email attachments of any assigned submitted work will not be accepted.***

Please review the course Late Policy within Syllabus (First Day Handout).

Project 2 Grading Rubric

Criterion	Value
An appropriate page title is present within the title tag. A page title of "Untitled" will not receive credit.	1
The external CSS file is properly linked to the PHP document.	2
Heading 1 tag containing verbiage is added to the web page using a print/echo statement as stated in the instructions. Syntax must be correct.	3
Included image is added to the web page using a print/echo statement with appropriate class, alt, and title attributes. Image must display properly within the order web page.	3
Introductory paragraph containing verbiage is added to the web page using a print/echo statement as stated in the instructions. Syntax must be correct.	3
Added form container tag with appropriate action and method attributes included. Syntax must be correct.	2
Add inputs for the first name, last name, email address, phone number, dog breed, and submit as stated in the instructions. Syntax must be correct.	3

Code is added to each form input that makes it "sticky" as stated in the instructions. Syntax must be correct.	10
Code is added to check to see if the form has been submitted as stated in the instructions. Syntax must be correct.	2
Validate is present for the dog's first name, last name, email address, dog breed, and phone numbers fields as stated in the instructions. Syntax must be correct.	5
Appropriate error messages display if the validation is not successful as stated in the instructions. Syntax must be correct.	5
A confirmation message is displayed if the form has been filled out correctly and contains the Dog's first name within the message as stated in the instructions. Syntax must be correct.	5
The body of the email contains the message, including the Dog's first name as stated in the instructions. Syntax must be correct.	5
The email is successfully sent and received using the mail() function.	5
PHP delimiters are present. Syntax must be correct.	2
Correct use of PHP commenting is present throughout each PHP document. Syntax must be correct.	2.5
Correct display of content/output to the Web browser as stated in the instructions. This includes a functioning PHP script, HTML, and CSS.	15
The Course Homepage text is hyperlinked to link to your course homepage with the footer tag of each document. Relative paths must be used.	1
Successfully validated HTML5 document and CSS when applicable. Validation must be successful to receive the full points.	5
TOTAL POSSIBLE POINTS:	80