PHP

This assignment is intended for you to understand building an application using HTML, CSS, and PHP. You will not need to use any JavaScript for this assignment. You will be able to complete this challenge by using the content from the course and reading online resources. Please complete the requirements listed below.

Essential Requirements:

- **The FOLDER should be named "<**Yourname**>PHP".** Replace your name with your real name. For example, my name is Hao Zhang, my folder would be "HaoZhangPHP". This requirement will be applied to every assignment therefore, always remember to rename your file accordingly.
- **Write and complete the following comment in your HTML document:**

<!—

name:

date:

-->

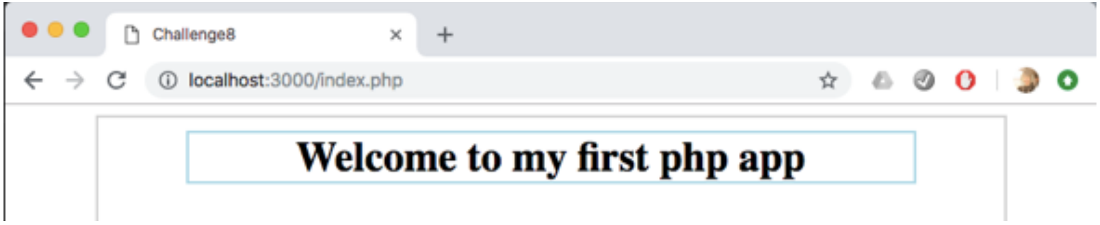
- Must be a well-formed HTML5 document that uses CSS for the page design and layout
  - **Including the 5 mandatory tags discussed in class**
- Use **External CSS** and use <link> in head of document.
  - **DO NOT** use Inline or Internal CSS.
- The output of your PHP code, after the code is finishing executing, should pass the validator.
  - Note: You will not be able to run PHP code through the validator, however you can take the output from the PHP code and check the output through the validator.

Overview:

In this assignment, you are to create a PHP file that:

- Is named index.php

PHP

- index.php should have a landing page with some text
    - Example: “Welcome to my first php app”
    - 

      Welcome to my first php app
    - http://www.professorwergeles.com/CS2830/Challenge8/
- Reads parameters sent through the $_GET Super-global Array
    - Read about $_GET at https://www.w3schools.com/php/php_superglobals.asp
- Displays information to the browser using echo/print
    - Read about echo/print at https://www.w3schools.com/php/php_echo_print.asp
- Performs the following 6 functions:
    - sayHi:
        - URL: …/index.php?myFunction=sayHi&myName=Nick
        - localhost: index.php?myFunction=sayHi&myName=Nick
    - showPic:
        - URL: …/index.php?myFunction=showPic
        - localhost: index.php?myFunction=showPic
    - showList:
        - URL: …/index.php?myFunction=showList
        - localhost: index.php?myFunction=showList
    - hamming:
        - URL: …/index.php?myFunction=hammingNumbers
        - localhost: index.php?myFunction=hammingNumbers&x=9
    - hammingSeq:
        - URL: …/index.php?myFunction=hammingSequence
        - localhost: index.php?myFunction=hammingSequence&y=9
    - Anagram:
        - URL: …/index.php?myFunction=anagram
        - localhost: index.php?myFunction=anagram&a=aan&b=ana
    - **Note:** You can click on the links above to see a live demo of each function
- Each function should error check to make sure all necessary parameters are correct
- If incorrect parameters are used, display a message to show an error.
    - See example screenshots below for an idea of how to handle errors.
- Each function should display an <h1> stating which function the user is currently on.

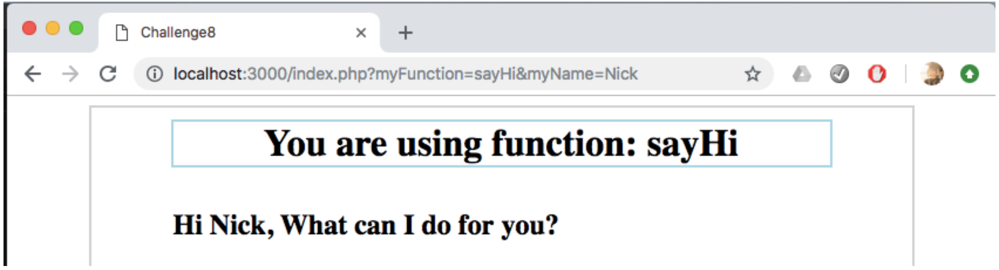
Note: DO NOT RENAME these functions or parameters! We will grade based on the given function names and parameter names and if you rename them, your app will not work.

PHP

Requirements:

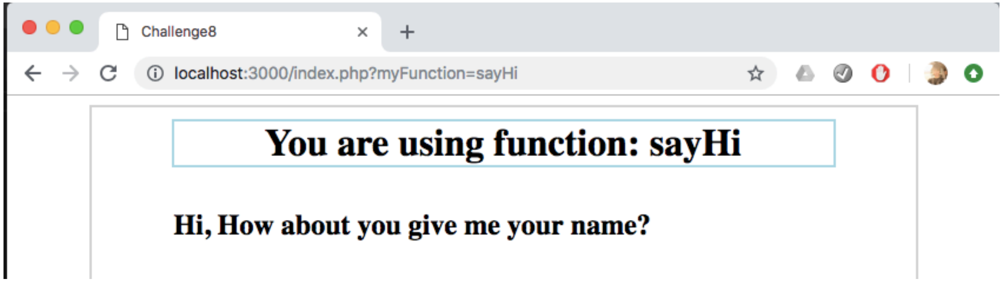
1. Must contain the **5 required tags**
2. Page should include the extra items to pass validation
3. When comparing strings then you should use a binary safe string comparison
4. Your code must properly display the error messages if input does not exist or the input is incorrect
5. Each function should display an <h1> stating which function the user is currently on.
6. Do not rename the functions or parameters otherwise your project will not be able to be graded
7. You must code I/O correctly, by closing the file when done, and checking for proper errors, like file does not exist, etc.
8. The layout and design of the user interface is up to you. Whatever you choose should be a well-organized, thoughtful, aesthetically pleasing, and a useable interface or an interface that logically makes sense. The layout should look like you made intentional choices and are in control of their placement. This means the layout should not be a disorganized mess that is a result of not knowing how to implement the user interface, layout, and/or code in a meaningful way.
9. You may expand the functionality beyond the basic requirements provided above if you choose. Usually expanding beyond the basic requirements gets rewarded.

Function sayHi:

- This function should display a message to the user reading: “Hi [user], what can I do for you?”
- Where [user] will be filled in with the parameter found in `$_GET[‘myName’]`
    - 

      You are using function: sayHi

      Hi Nick, What can I do for you?

- If myName is null, display an error message.
    - 

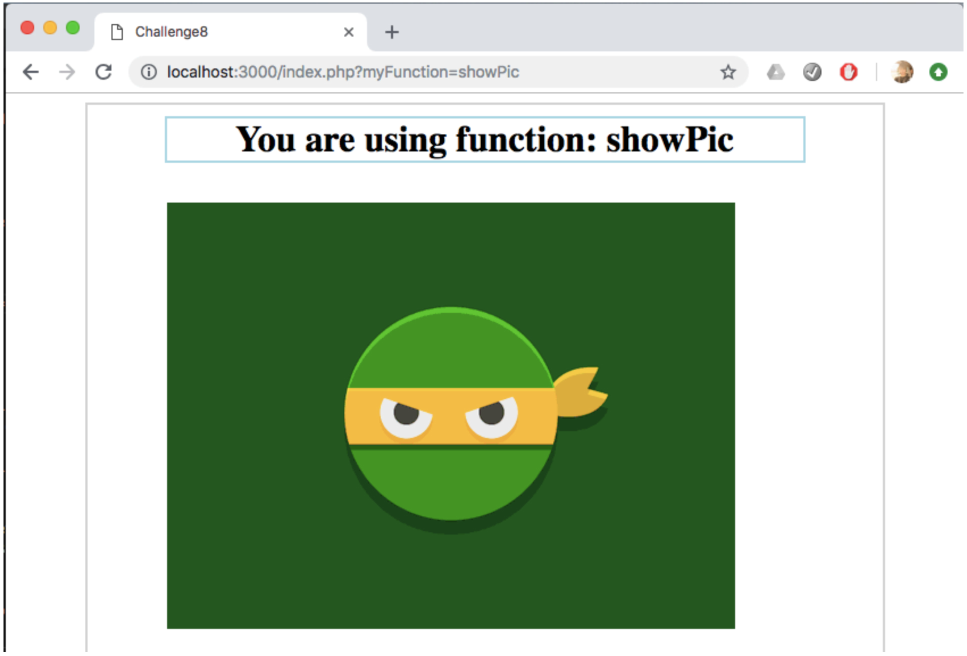
      You are using function: sayHi

      Hi, How about you give me your name?

Note:

- If you look at the URL bar you can see I am passing parameters using the "?" such as:
    - `index.php?myFunction=sayHi`
- If I want to pass multiple parameters, I can use the "&" to append them such as:
    - `index.php?myFunction=sayHi&myName=Nick`
- These parameters will all show up in the super global array called `$_GET`.

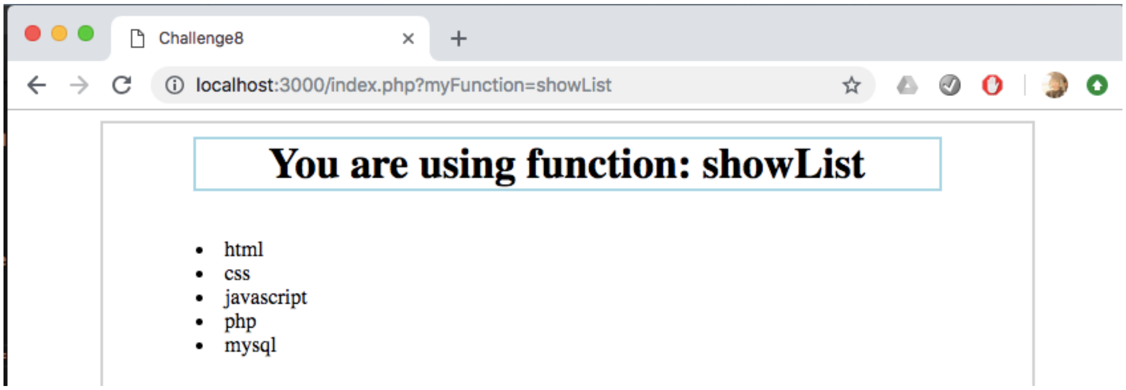
Function showPic:

- This function should display the image "cool.gif" that is attached to this assignment.
    - 

      You are using function: showPic

Function showList:

- This function should display the contents of “list.txt” attached to this assignment in the form of an unordered list.
- You will need to read in the text from list.txt line by line in a loop and display it to the user.
- Be sure to use `fclose()` when you are finished with the file to prevent memory leaks.
- We will be grading with a different list.txt so make sure your function will be able to support more than 5 items.

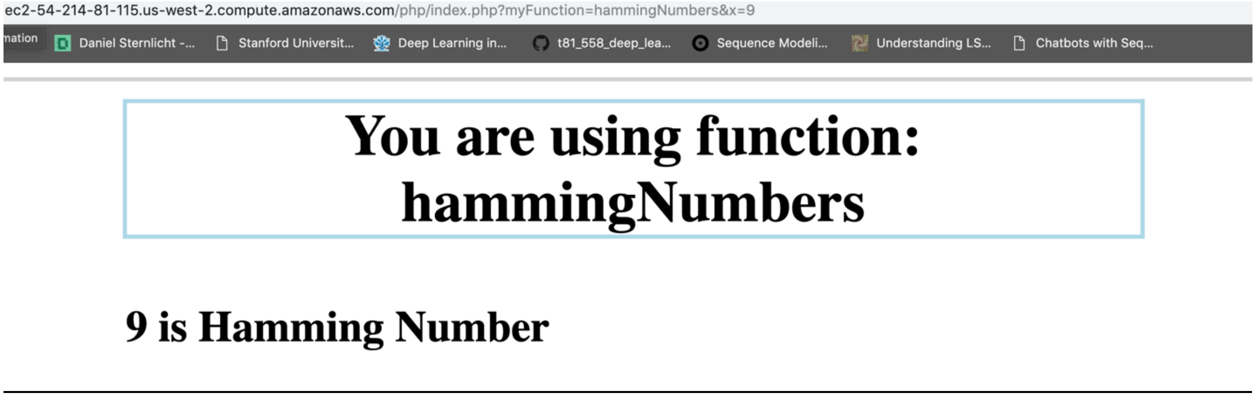


You are using function: showList

- html
- css
- javascript
- php
- mysql

Function hamming:

- Hamming sequence is also called regular number, please read about it here: https://en.wikipedia.org/wiki/Regular_number
- This function should receive input of a number (exampler: number is [x])
- The function will decide whether the number given is a hamming number or not
- The result will print out the number and state whether it’s a hamming number or not a hamming number.
- Example output: Number [x] is not hamming number or Number [x] is hamming number

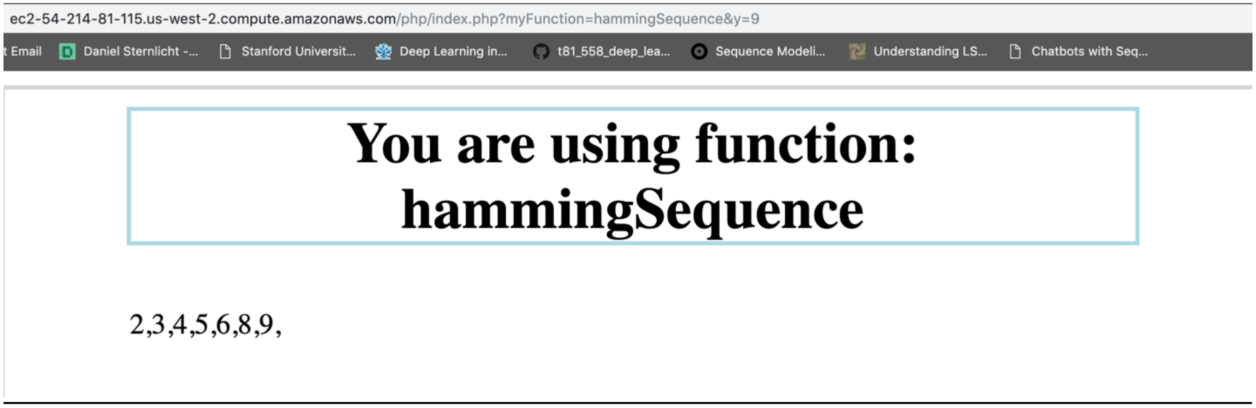


You are using function: hammingNumbers

9 is Hamming Number

<u>Function hammingSeq:</u>

- This function should receive a number as input (example: number is [x])
- The function will decide whether the number given is a hamming number or not
- If it is a hamming number then it will print out the sequence of hamming numbers until it reaches the input number. (example output x = 24: [x] is hamming number, sequence = 2,3,4,5,6,8,9,10,12,15,16,18,20,24)

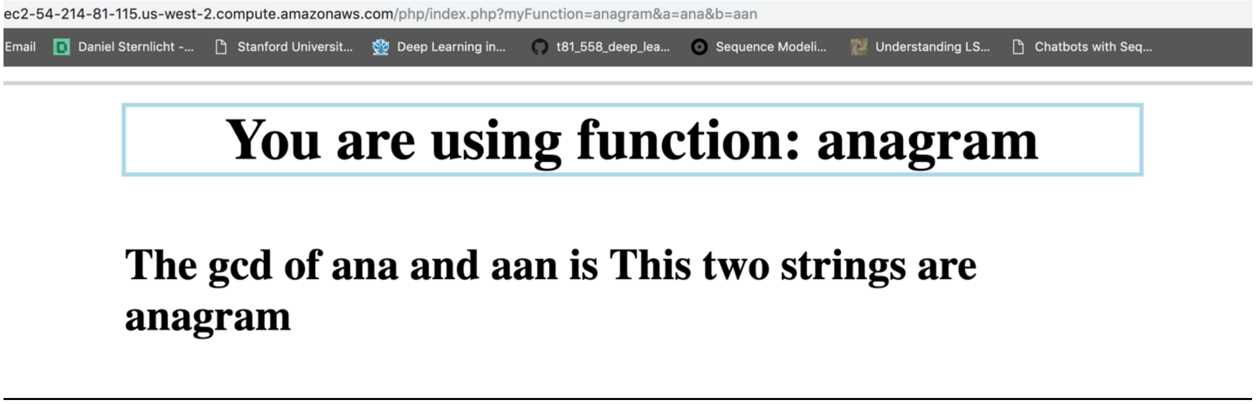


You are using function: hammingSequence

```
2,3,4,5,6,8,9,
```

<u>Function annagram:</u>

- This function should receive 2 words separated by a comma as input (example: [x],[y])
- The function will decide whether the words are an anagram or not
- If the two words are an anagram, the function will print an output “an anagram!” or “not an anagram”



You are using function: anagram

The gcd of ana and aan is This two strings are anagram

PHP

Upload to Amazon EC2

- When you are done with the code, make sure you have your completed files uploaded to your ec2 instance
- Make sure to make a new directory called "WebTrainingPHP" so that the index file doesn't clash with other index files from previous challenges
- These simple problems cannot be used as an excuse for not getting your challenge done

Tips and Debugging

- Most likely you will run into compile errors, unless you are some kind of PHP compiler wiz, so you will need to know how to debug.
- You can upload any code you have to your ec2 and run the command:

```
php index.php
```

- This will tell the PHP compiler to compile your code and list any errors it encounters.
- I suggest coding one function at a time and trying to compile after you have written a small piece of code.
- Perhaps start with just getting the landing page to work. Then slowly add each function.