

Assignment5

Part 1

A5 Requirements:

1. Requires **A4** cloned files.
2. Review subdirectories and files
3. Open **index.php** and review code:
 - a. Suitably modify meta tags
 - b. Change title, navigation links, and header tags appropriately
 - c. **See videos for complete development.**
 - d. Using **your** client-side validated form from **A4 (index.php)**, copy the file, rename it (**add_petstore.php**), and place it into your local **a5** subdirectory
 - e. **Turn off client-side validation by commenting out the following code:**
`<?php //include_once("../js/include_js.php"); ?>`
 - f. Add the following **server-side** validation and regular expressions-- as per the database entity attribute requirements (**and screenshots below**):
 - i. ***All*** input fields, except Notes **are required**
 - ii. Use min/max jQuery validation
 - iii. Use **regex** to only allow appropriate characters for each control:
pst_name, pst_street, pst_city, pst_state, pst_zip, pst_phone, pst_email, pst_url, pst_ytd_sales, pst_notes
Name (provided): letters, numbers, hyphens, and underscore
Street, City (provided):
no more than 30 characters
Street: must only contain letters, numbers, commas, or periods
City: must only contain letters, numbers
State:
must be 2 characters
must only contain letters
Zip:
must be between 5 and 9 digits, inclusive
must only contain numbers
Phone:
must be 10 digits, including area code
must only contain numbers
Email: See <http://www.qcitr.com/usefullinks.htm#lesson7>
URL:
no more than 100 characters
See <http://www.qcitr.com/usefullinks.htm#lesson7>
YTD Sales:
no more than 10 digits, including decimal point
can only contain numbers, and decimal point (if used)
 - iv. ***After*** testing jQuery validation, use HTML5 property to limit the number of characters for each control
 - g. Review other sub-directories and files, and modify them accordingly
 - h. Use git to push ***all*** your local **lis4381** files (which includes **a5**) and changes to **your** remote **lis4381** **Bitbucket** repo

Notes: See Ch15 (Regular Expressions)

Be sure to include the following **screenshots** (**see below**):

1. Before ***and*** after successful add
2. Failed validation

Helper Videos:

1. http://www.qcitr.com/vids/LIS4381_A5a.mp4
2. http://www.qcitr.com/vids/LIS4381_A5b.mp4
3. http://www.qcitr.com/vids/LIS4381_A5c.mp4

Part 2

README.md file should include the following items:

1. **Course title, your name, assignment requirements, as per A1;**
2. **Screenshots** as per below examples;
3. Link to **local lis4381 web app**: <http://localhost/repos/lis4381/>

Deliverables:

1. Provide **Bitbucket** read-only access to **lis4381** repo (Language PHP), include links to the other assignment repos you created in **README.md**, using **Markdown** syntax (**README.md** must also include screenshots as per above.)
2. **FSU's Learning Management System**: include lis4381 **Bitbucket** repo link

index.php

Assignment 5
Basic server-side validation
Server-side validation, and prepared statements (helps prevent SQL injection).
Displays user-entered data, and permits users to add data.
(See P2 to update and delete pet store data.)
(Table: petstore)

Pet Stores
[Add Pet Store](#)

Show: 10 entries

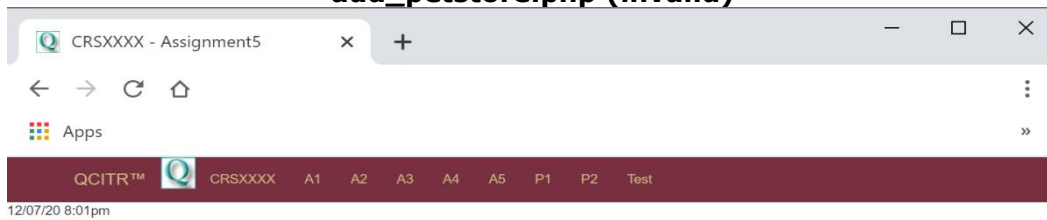
Name	Street	City	State	Zip	Phone	Email	URL	YTD Sales	Notes
Claws and Paws	123 Main St.	L.A.	CA	09010341	8185551234	clawspaws@aol.com	http://www.clawspaws.com	50000.00	notes1 Edit
Furry Friends	1435 Cherry Dr.	Chicago	IL	618665031	8471234567	furryfriends@hotmail.com	http://www.furryfriends.com	65000.00	notes2 Edit
Pet Palace	19803 Tennessee St.	Tallahassee	FL	000036790	8504445123	petpalace@gmail.com	http://www.petpalace.com	34000.00	notes4 Edit
Pet Partners	911 4th St.	NY	NY	001256728	2129676543	petpartners@yahoo.com	http://www.petpartners.com	87000.00	notes3 Edit
Purrs and Grrs	751 Top Lane	Phoenix	AZ	852122156	4809130015	purrsandgrrs@msn.com	http://www.purrsandgrrs.com	97500.00	notes5 Edit

Showing 1 to 5 of 5 entries

Previous 1 Next

© 2020 My Footer, Inc.

add_petstore.php (invalid)



Assignment 5

Basic server-side validation

Server-side validation, and prepared statements (helps prevent SQL injection).

Displays user-entered data, and permits users to add data.

(See P2 to update and delete pet store data.)

(Table: petstore)

Pet Stores

Name:

Street:

City:

State:

Zip:

Phone:

Email:

URL:

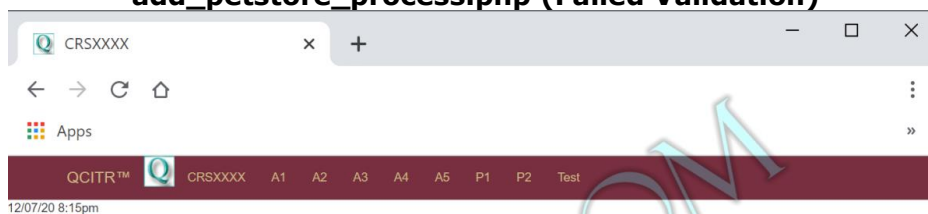
YTD Sales:

Notes:

Add

© 2020 My Footer, Inc.

add_petstore_process.php (Failed Validation)



Assignment 5

Basic server-side validation

Server-side validation, and prepared statements (helps prevent SQL injection).

Displays user-entered data, and permits users to add data.

(See P2 to update and delete pet store data.)

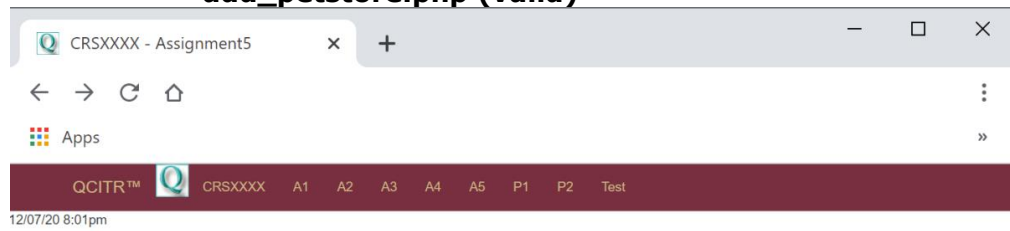
(Table: petstore)

Error!

Name can only contain letters, numbers, hyphens, and underscore.

© 2020 My Footer, Inc.

add_petstore.php (valid)



Assignment 5

Basic server-side validation

Server-side validation, and prepared statements (helps prevent SQL injection).

Displays user-entered data, and permits users to add data.

(See P2 to update and delete pet store data.)

(Table: petstore)

Pet Stores

Name:

Street:

City:

State:

Zip:

Phone:

Email:

URL:

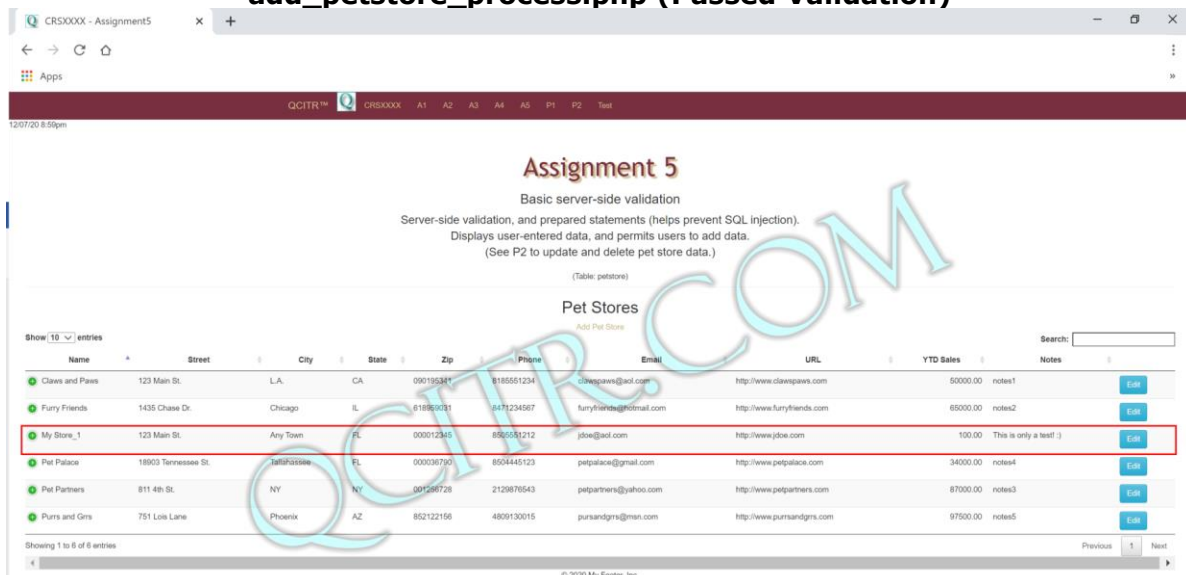
YTD Sales:

Notes:

Add

© 2020 My Footer, Inc.

add_petstore_process.php (Passed Validation)



Part 3

Questions (PHP/MySQL: Chs. 11, 12, 19):

1. The easiest way to perform a mathematical calculation, and add the values of all the elements in an array is to use
 - a. a for loop
 - b. a foreach loop
 - c. the array_sum function
 - d. the sum_array function

Note: To be clear, there is no difference in structure between a "stack" or "queue"--only in access. They are **both** arrays. They are only called by different terms due to the method of access. The same PHP array can be accessed using various functions, and ***all*** PHP arrays are created using the **array()** function.

2. Which of the following functions removes and returns the last (or next) element in a LIFO array (aka stack)?
 - a. array_shift b. array_unshift c. array_push d. array_pop
3. Which of the following functions will add an element to the end of a LIFO array (aka stack)?
 - a. array_shift b. array_unshift c. array_push d. array_pop
4. Which of the following functions removes and returns the first element in a FIFO array (aka queue)?
 - a. array_shift b. array_unshift c. array_push d. array_pop
5. Which of the following functions will add an element to the beginning of a FIFO array (aka queue)?
 - a. array_shift b. array_unshift c. array_push d. array_pop
6. Which of the following statements about associative arrays is NOT true?
 - a. You can use the count function to return the number of elements in the array.
 - b. You can use both integer and string indexes with an associative array.
 - c. You can use a foreach loop to access the values of an associative array but not the indexes
 - d. You can delete elements from an associative array.
7. A multidimensional array in PHP is essentially _____.
 - a. a two-dimensional array
 - b. nonexistent
 - c. an array of arrays
 - d. an associative array
8. If the key is omitted when adding a value to an array, and there were no integer indexes added, PHP assigns an index of _____ to the element.
 - 0 1 2 PHP assigns no value
9. Merge two or more arrays with the _____ function.
array_merge() array_plus() array_paste() array_values()
10. Multidimensional arrays are not limited to _____.
two dimensions associative arrays indexed elements recursive arrays

11. The ____ argument of the `array_splice()` function indicates the name of the array you want to modify.
`array_name` `array_value` `array_function` `array_mod`
12. The ____ function determines whether a given index or key exists.
`array_values()` `array_key_exists()` `array_key()` `array_set()`
13. The ____ function determines whether a given value exists in an array.
`array_search()` `in_array()` `find_array()` `array_values()`
14. The ____ function lets you add or remove elements anywhere else in the array.
`array_add()` `array_split()` `array_splice()` `array_values()`
15. The ____ function removes duplicate elements from an array.
`array_dup()` `array_values()` `array_unique()` `array()`
16. The ____ function returns a Boolean value of true if a given value exists in the array.
`array_search()` `array_boole()` `array()` `in_array()`
17. The ____ function returns an array of elements that exist in one array but not in any other arrays to which it is compared.
`array_compare()` `array_diff()` `array_search()` `array_match()`
18. To perform a reverse sort on an associative array by key and maintain the existing keys, use the ____ function.
`ksort()` `krsort()` `uk_sort()` `sort()`
19. To remove multiple elements using the `unset()` function, separate each ____ name with commas.
`variable` `array` `element` `argument`
20. To renumber an indexed array's elements, you need to run the ____ function.
`unset()` `array_values()` `array()` `array_num()`
21. To sort an associative array by key, use the ____ function.
`natsort()` `ksort()` `rsort()` `asort()`
22. You use the ____ function to return a portion of an array and assign it to another array.
`array_slice()` `array_set()` `array_assign()` `array_key()`
23. Suppose an array of country codes and country names is created with a statement like this:
- ```
$country_codes = array('DEU' => 'Germany', 'JPN' => 'Japan', 'ARG' => 'Argentina', 'USA' => 'United States');
```
- If you want to use echo statements to display a table of all of the codes and names, you could code those statements within a loop that starts like this:
- `foreach ($country_codes as $name[code]) {...`
  - `foreach ($country_codes as $code => $name) {...`
  - `for ($country_codes as $name[code]) {...`
  - `for ($country_codes as $code => $name) {...`

24. To delete a cookie, you
- delete the `$_COOKIE` variable
  - use the `delete_cookie` function
  - use the `remove_cookie` function
  - set the cookie's value to an empty string and its expiration date to a time in the past

25. The `$_SESSION` variable for a session
- is a regular array
  - is an associative array
  - is an object
  - is a cookie

26. When you use session tracking, each HTTP request includes
- a URL that stores the session ID
  - a cookie that stores the session ID
  - a cookie that stores the session data
  - a cookie that stores the session ID and the session data

27. Given the code:

```
$employee = array();
$employee["name"] = "John Smith";
$employee["age"] = 29;
```

which of these statements assigns "John Smith is 29" to `$message`?

- `$message = $employee[0] . " is " . $employee[1];`
- `$message = $employee[1] . " is " . $employee[2];`
- `$message = $employee["name"] . " is " . $employee["age"];`
- `$message = $employees[0]["name"] . " is " . $employees[0]["age"];`

28. In the object-oriented interface for `mysqli`, you use the `query` method to run a `SELECT` statement. In the procedural interface, you use this function:
- `query_mysqli`
  - `mysqli_query`
  - `execute_mysqli`
  - `mysqli_execute`

29. One disadvantage of using `mysqli` is that
- it doesn't provide an object-oriented interface
  - it doesn't provide for prepared statements
  - it can only be used with MySQL
  - it only provides a procedural interface

30. The `lastInsertID` method of the `PDO` class
- sets the ID for the last record that has been inserted into a database
  - sets the ID for the last record of several that are going to be inserted into a database
  - gets the ID for the last record that has been inserted into a database
  - gets the ID for the last record with an auto-increment ID that has been inserted into a database

31. The prepare method of a PDO object

- a. must be run before values are assigned to the parameters in a prepared statement
- b. assigns the values to the parameters in a prepared statement
- c. retrieves the rows of a prepared statement
- d. must be run after values are assigned to the parameters in a prepared statement

32. When you use the exception error mode with PDO objects, you can

- a. use the errorCode and errorInfo methods to find out whether an error occurred
- b. use the errorCode and errorInfo properties to find out whether an error occurred
- c. use try-catch statements to handle the exceptions that occur
- d. use try-catch statements to throw exceptions

33. After you run a prepared SQL statement with PDO, you can use the fetchAll method of the \_\_\_\_\_ object to get an array that corresponds with the rows in the result set.

mysqlStatement   PDOStatement   PHPStatement   mysqliStatement

34. When you use exception error mode, PHP throws a \_\_\_\_\_ object when a PDO error occurs.

PDOException   Error   Exception   PHPError

35. When you use the procedural interface of mysqli, you use \_\_\_\_\_ to do the database operations.

SQL code   functions   PDO statements   object-oriented interface

36. \_\_\_\_\_ provides an object-oriented approach to working with a variety of database servers.

PHP's MySQL extension   PDO   Dynamic SQL   PHP's mysqli extension

37. When you use a prepared SQL statement with PDO, you need to \_\_\_\_\_ the parameters to variable values before

you can run the statement.

- a. fetch or fetchall
- b. bind
- c. return
- d. use dynamic SQL statements in