

PROG2001 – WEB DESIGN AND DEVELOPMENT

A-02 : ANIMAL-SERVE – WEBSITE DEVELOPED USING CGI AND PHP

OVERVIEW

In this assignment, you (and your partner) will create a simple two-page web site that uses a form to provide the parameters to a web server. You will implement both the CGI and PHP versions of this website...

This is a partner-based assignment and you can have partner if you wish.

OBJECTIVES

This assignment supports the following course objectives:

- To demonstrate the operation of server side technologies (CGI and PHP)
- To demonstrate the ability to create user web forms

ACADEMIC INTEGRITY AND LATE PENALTIES

- Please refer to the SET Policies document regarding [Academic Integrity Information](#)
- Please refer to the SET Policies document regarding [Late Policy](#)

EVALUATION

- Please refer to the assignment weighting in the *Instructional Plan* for the course as well as the assignment's Rubric in the course shell.

PREPARATION

Review Module-03, Module-04 and Module-05 lesson content as well as each module's code samples – they will help you in this assignment.

REQUIREMENTS

1. In this assignment – you need to create two (2) HTML forms that are identical, except where the CGI implementation uses the form GET method and the PHP implementation uses the form POST method.
2. Each of the implementations start with this form which must:
 - a. Ask for the user's name and validate input using JavaScript. (i.e. is a mandatory field and not blank – can accept any characters)
 - NOTE: when the user initially visits your ZOO – the “name” prompt, entry textbox and a button should be the only elements visible to them
 - b. After getting the user's name and validating it, you may then display and ask the user (by name) to choose an animal from a list of six animals (you choose which six)
 - use a combo-box / drop-down / select list for this control
 - make the first option (default selected) in the dropdown blank
 - you will need to validate that the user has then actually chosen an animal from the list before submitting
 - c. Once validated, the user's name and the animal selection are to be sent to the server as appropriate key value pairs, in each of the two HTML pages, respectively.
 - HINT: This means that the `<form>` which asks for the user's choice of animal will also need to contain the user's name input on it ... hmmm ... how can it be *hidden*?
 - d. Name these HTML files `cgi-zoo.html` and `php-zoo.html`.
3. The root folder of your web site on the server is to contain the two HTML pages as well as the CGI executable and PHP server-side page.
4. Create a sub-folder (called theZoo) and have it exist in your site.
 - a. This directory is to contain images of the 6 animals you chose as well as 6 short descriptions (text files) corresponding to the animals.
 - The animal descriptions can be made up of a bullet-list of facts about the animal, a short paragraph description or story of the animal, etc.
 - NOTE: A good idea here is to use HTML formatting within the text files – so that when an animal is selected and its information is displayed – the text is nicely formatted.
 - b. Each server technology composes the response, with:
 - A salutation to the person (using the name entered in the HTML form)
 - The picture of the selected animal being displayed (no larger than 400 pixels x 400 pixels)
 - The description of the animal – to be placed beside the picture

- c. Remember when referencing your image files and text files to avoid using absolute file addressing within your website
- 5. For the CGI solution, the server file must be called **animals.exe** and can be created using either C, C++ or C#. For the PHP solution, the server file must be called **animals.php**
- 6. You are not required to make the pages overly fancy, but a reasonable amount of effort should be made to align the content well and also use CSS techniques (inline, internal or external styles can be used) to increase the user experience. Use some of the *best practice* design principles discussed in class to make the user's trip to your Zoo enjoyable.
- 7. Good programming practices must be exhibited as per SET standards. Make sure to comment all source files appropriately
- 8. Ensure that your *Animal-Serve* application runs properly and consistently within the Microsoft Edge and Chrome browsers

FILE NAMING REQUIREMENTS

As mentioned above in the Requirements, the solution for this assignment must be developed and delivered using specific filenames in order to make marking and testing straightforward.

The filenames are:

- `cgi-zoo.html` and `php-zoo.html`
- `animals.exe` and `animals.php`

SUBMISSION REQUIREMENTS

When submitting your solution to this assignment, hand-in a single ZIP'd file containing:

1. The two HTML *Animal-Serve* launching pages (HTML files)
2. The cleaned Visual Studio solution for the CGI executable file (no need to submit the executable)
3. The PHP server-side source file
4. The supporting folder (correctly named to match your code) with appropriate files (animal images and textual descriptions).
5. If you're doing this assignment with a partner, then please ensure you include both partner names in your file header comments (in HTML, CGI source and PHP source)
6. Also remember that this solution will be tested using Microsoft Edge as well as Chrome
7. Please ZIP up these files and submit to the appropriate eConestoga Dropbox by the deadline
 - a. Please give your ZIP submission the filename *lastName-firstInitial.zip* (e.g. if you are Sally Jones – then your ZIP should be named `jones-s.zip`)
 - b. If you are working with a partner, then include both your names in the ZIP filename (e.g. if Sally Jones is working with John Smith – then your ZIP should be named `jones-s-smith-j.zip`)

NOTE: If working with a partner, only one partner need submit the solution

Possible Hint that Might Help ...

When you begin to implement your CGI solution – you may find that your solution has issues finding and properly opening your animal text file? Here are some hints that might help:

- remember that your CGI executable will be present in the same directory as your cgi-zoo.html file and the images and text files will be located in a sub-directory called theZoo
- also remember that when accessing a file (to open, read, close, etc.) you are making an OS API call - and it has nothing to do with the web root or localhost ...
 - you need to open the file based on wherever the .txt file is located on the **file-system**
 - and you cannot depend on the .txt file being in the same location on my test system as it is on yours ...
- so a good way to write portable, robust code is to revisit some of the extra references in the CGI module and examine some of the other CGI environment variables to see if there is anything that you can do with the values of any of the variables to determine where the CGI executable is and modify it into pointing at the desired .txt file
- Please note that you may find a different a way to open, read and close the .txt file -- so this **HINT** may not be needed or apply to you!

Something to Test:

1. After developing and debugging your CGI and PHP solutions – ensure that there is no dependency on the absolute file-paths used for the text files and/or image files.
 - a. Do this by moving the entire website (from the root down) to a different location in your file system and adjusting the *published* location within your IIS
 - b. For example – move the website to a sub-sub-sub-directory on the C: drive – but still within your *web space* (e.g. C:\localWebSite\WDD\A-02\Animal-Serve) – and open the pages (e.g. <http://localhost/WDD/A-02/Animal-Serve/cgi-zoo.html>)