CS 4WW3/6WW3 – Web Systems and Web Computing

Project, Part 3 – Server-side

Worth: 20% (4WW3), 10% (6WW3)

Due: Nov 28, 2019, 10pm

Specification date: 2019/09/22

Synopsis

This part of the project will require you to use PHP and MySQL to implement the full dynamic server-side functionality of your website and deploy it on your virtual machine. You will use Amazon S3 buckets to store user uploads.

You must upload a ZIP file containing all of your files to Avenue by the deadline. You must also ensure your website is functioning correctly on Amazon Web Services, and submit the URL for your live server in the README file in the ZIP upload. See below for details on the contents of your ZIP file.

Core Programming Tasks

You will extend part 2 of your project as follows.

Pages in your website

In part 1 and part 2 of the project, you created sample versions of the following pages. Indicated below are the changes or additional functionality you must incorporate.

- A search form that allows users to search for objects by name (entered into a text box) or rating (selected from a drop-down box).
    - For part 3 of the project, your search form should submit the search query to the web server, and return an appropriate dynamically generated results page.
- A **dynamically generated** results page showing the results of a search (a) on a map, and (b) in a tabular format. From the results table, users should be able to link to a more detailed screen for individual objects.
    - For part 3 of the project, these results should be dynamically generated based on the search query submitted from the search form above. Both the table and the map markers should be dynamically generated using PHP by searching the database appropriately.
- **Dynamically generated** individual object pages, with details about the object itself, its location on a map, as well as a list of all reviews and ratings that have been entered by users.
    - For part 3 of the project, clicking on an entry in the search results page should take you to a dynamically generated page for each individual object. As before, this should

include a live map showing the location of the object. It should also include all previously submitted reviews and ratings.

  - For part 3 of the project, logged-in users should also be able to submit a review and a rating for an individual object, which is inserted into the database and displayed for future users.

- An object submission page, containing a form with which users could submit a new object.
  - For part 3 of the project, your object submission page should submit the user's data to a PHP script which, after performing appropriate server-side validation, adds the data to the database.
  - Only logged-in users should be able to submit a new object.
  - The object submission page should include the ability for users to upload an image. You should store this image on an Amazon S3 bucket. You may use Amazon's PHP libraries for interacting with S3, or another library if you prefer.
- A user registration page, containing a form in which users are asked to enter the information required to sign up for an account.
  - For part 3 of the project, your registration page should submit the user's data to a PHP script which, after performing appropriate server-side validation, adds the data to the database.
  - Although many websites include email confirmation or email password reset features, you do not need to do this.

You should pre-populate your website with 5+ objects so that there is some data to see.

Implementation

You will need to design an SQL database that is suitable for the data in your application. It is likely you will need at least three tables: one for users, one for objects, and one for reviews.

You will need to write PHP code that implements your functionality for part 2. You may also need to modify some of the HTML, CSS, or Javascript code you wrote for parts 1 and 2.

Some requirements for your implementation are as follows.

- All requirements on your HTML, CSS, and Javascript code from project part 1 and part 2 specifications remain in place.
- A **larger than normal amount of comments** are required to be included in your PHP code to demonstrate to your tutor (assignment marker) that you thoroughly understand the meaning of everything that you are using.
- You must use the PHP Data Objects (PDO) API for database access.
- You should extract common page elements (headers, menus, sidebars, etc.) into separate PHP include files to maintain a common look-and-feel without repeating too much code between pages.
- Your website should include protections against malicious user-entered data, including protection against cross-site scripting attacks and SQL injection attacks.

Server deployment

You will need to deploy your project on a live website. In addition to what you completed in part 2, you will need to:

- Set up an Amazon S3 bucket for storing user-uploaded images.

If you use Github or some other public repository for storing your source code, it is very important that you do not put a copy of your AWS password or access keys in your Github repository: there are automated scripts that malicious parties use to scrape Github repositories for AWS credentials, and students in this course in the past have had their account broken into and had charges run up because of this.

Add-on Programming Tasks

CS 4WW3 students can receive up to 5 extra bonus marks for completing the add-on task.

CS 6WW3 students must complete the add-on task.

Add-on 3: AJAX

As above, on your individual object page, users can submit a rating and review for an object. The simple way to do this is for the user to click submit, the browser navigates to a new page as requested via a GET or POST, the server inserts the data into the database, generates a new HTML response of the full page, returns this to the browser, and the browser loads the new page.

You will use AJAX for this procedure so that the user's browser doesn't load a new page when the user submits a review. Instead, the submission of the rating and review should be done using AJAX. The server should insert the data into the database and return a success/failure code to the browser, and then the browser runs Javascript to add the new review to the HTML code in the appropriate place, without reloading the whole page.

Restrictions

You may not use any client-side technologies that require plugins, such as Java applets, Flash, or Silverlight.

See the FAQ at the end of this document for information about which external frameworks you are allowed to use.

Getting Help

To achieve top marks in this assignment, you will need to supplement what you have learned in lectures and tutorials with details from textbooks or online resources. Fortunately, there are many resources available on the Internet for web development. On Avenue, I have posted links to useful sites about PHP.

During the workshops, your tutor will be able to provide guidance on the core programming tasks. You should be prepared to explain what you are trying and why you are stuck, rather than just asking "how do you do this?".

You are welcome to ask questions of your peers either offline or on the Facebook discussion group. As a last resort, you can email me (doq4@mcmaster.ca) with a question or to request a consultation meeting.

Submission Instructions

SUBMISSION

(1) You must upload a ZIP file containing all your files to Avenue by the deadline; see details below for the contents of your ZIP file.

(2) You should also have your files copied to your web server and set up so that they appear as a live webpage.

(3) The README file in your ZIP file will include a:

- **a. link to your live server and**
- **b. git repository (add doq4@mcmaster.ca as a contributor).**

(4) Submit on Avenue.

Your ZIP file should contain the following:

- README.txt: A text file containing:
  - Your name and student number.
  - The URL of your live website.
  - If you are CS 4WW3 student doing an add-on for extra credit, say which add-on task you are doing.
  - An explanation of any unusual choices, or things you feel we should know about your project.
- datamodel.sql: A text file containing the CREATE TABLE statements used to create your database
- All PHP, HTML, CSS, and Javascript for your website. You may organize these additional files into subfolders, as you like. You do not need to include any images, videos, fonts, or third party scripts or APIs in your ZIP file, as we will not be running any of the code from your ZIP file, only reading the source.

As noted above, you should have a live website working on a server, which remains unchanged (except for the possible submission of additional objects or reviews) from the due date (Nov 28) until the end of exams (Dec 20).

We **will only mark the contents of the part 3 ZIP and your live website. We will not go back to your part 1 or part 2 ZIP.**

You should limit your ZIP file to 30 MB.

Marking

Your website should work in any modern mainstream desktop browser: Chrome, Firefox, Internet Explorer / Edge, Opera, or Safari; as well as Safari on iOS and Android Chrome.

We will test your live website at least on the most recent version of Chrome on the desktop, but we may also use another modern browser to check compatibility. For the mobile version, we will test your site using the (desktop) Chrome Developer Tools mobile simulator at the “iPhone 7” size as indicated above, but we may also test on an actual mobile phone or emulator to check compatibility.

Your website must follow the WCAG guidelines for accessibility.

Marks will be allocated as follows:

- 20 marks for the core functionality
- 10 marks for the quality of the PHP code
- 3 marks for database design
- 5 marks for security and deployment
- 5 marks for add-on #3 (bonus for 4WW3 students, required for 6WW3 students)

Please see the separate assessment criteria sheet for the detailed marking criteria.

Academic Integrity and Copyright

Except where allowed below, the work you submit for this assignment must be your own.

A good way for beginners to learn about creating web pages is to examine the source of other high-quality pages available on the web and to learn how they do various things. It is unethical to plagiarise such code verbatim, but it is quite OK to examine them, understand how they work and apply the same techniques and approaches in your web site. Different web pages can provide different things; one web site might illustrate the use of a font or colour scheme that you find effective, while another might use CSS to achieve a page layout that you like. Others still might use JavaScript to achieve dynamic effects, such as drop down menus. Other web sites of the same genre might provide some ideas regarding content and functionality.

It is generally acceptable to use small snippets of code you find on the Internet—not more than 5 lines—without attribution. You should not use larger pieces of other people’s code in this assignment.

You may not use any external frameworks, libraries, or templates in developing your website, except as indicated in the FAQ below. You are permitted to use a CSS reset stylesheet if you want.

If you want to use other people’s images, fonts, or videos in your website, you may do provided you abide by all copyright restrictions. For images, this means you should only use images that you make yourself, or images that are either in the public domain or licensed under terms that allow reuse (e.g., certain Creative Commons licenses). For fonts, this means you should only use fonts that are licensed for web embedding, or web-font services such as Google Fonts. Note that when using properly licensed images in building your own website, it is generally considered polite to locally host the image to avoid placing bandwidth load on the original image provider.

We may use similarity-detecting tools to help identify submissions that share code.

Late Assignments

Late assignments will be penalized at 20% per day to a maximum of 5 days. Students with extenuating circumstances should contact the instructor well in advance of the deadline. The McMaster Student Absence Form (MSAF) can be applied to academic work worth less than 25% of the final grade, i.e., a lab report. You must submit the MSAF online at http://www.mcmaster.ca/msaf/. I will automatically grant a 3 calendar day extension upon submission of an MSAF.

Frequently Asked Questions

See the part 1 specification for FAQs about HTML and CSS and part 2 specification for FAQs about Javascript.

1. *Can I use Joomla / cake / this fancy PHP ORM framework?*
   ~~No~~. **Yes!**
2. *Can I use this fancy AJAX framework?*
   ~~No.~~ **Yes.** You may use jQuery for AJAX, however.
3. *What libraries am I allowed to use for interacting with S3?*
   You can use Amazon's PHP libraries for interacting with S3, or any other library of your choosing.