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

Project, Part 2 – Client-side, dynamic

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

Due: Thursday, October 24, 2019, 10pm

Specification date: 2019/09/22

Synopsis

This part of the project will require you to use Javascript to add dynamic aspects to the client side of the site, focusing on form validation and maps; in part 3, you will implement the full dynamic server-side functionality. In part 2, your main tasks will be to add client-side form validation using JavaScript and HTML5/CSS3, and to add a dynamic map to your website.

You will also set up a web server on any server, such as a dedicated, shared, or virtual server on the Amazon Web Services Elastic Cloud Compute (EC2) platform, and configure SSL/TLS encryption for your website.

SUBMISSION

(1) You must upload a ZIP file containing all your files to Avenue by the deadline; see details on project specification 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.

Core Programming Tasks

You will extend part 1 of you project as follows.

Pages in your website

In part 1 of the project, you created 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 2 of the project, you should add a button allowing the user to search based on their location, which uses the HTML5 Geolocation API to retrieve the user's location.

- A sample 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 2 of the project, you should embed a live map using Javascript. You can use any mapping API; see below for some links. Your map should show markers with the search results on the map. When the user clicks the marker, a label should appear with some information about the result, including a link to the individual item page.
    - For part 2 of the project, since you will not be implementing the database and dynamic server-side components, your sample results page (including the map markers) may be hard-coded into your HTML source file.
- A sample individual object page, 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 2 of the project, you should embed a live map using Javascript. Your map should show the location of the individual object.
    - For part 2 of the project, since you will not be implementing the database and dynamic server-side components, you only need to create 1 sample individual object page, and it should include a few sample reviews and ratings hard-coded into your HTML source file.
- An object submission page, containing a form with which users could submit a new object. The form should have fields for the name of the object, a description of the object, and its location as a pair of latitude-longitude coordinates. The form should also allow users to upload an image for the object.
    - For part 2 of the project, you should add client-side form validation **using HTML5/CSS**. You should allow the user to set the location of the object using the Geolocation API.
    - For part 2 of the project, your form does not have to submit the results anywhere, since you don't have a server side yet.
- 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 2 of the project, you should add client-side form validation **using Javascript**.
    - For part 2 of the project, your form does not have to submit the results anywhere, since you don't have a server side yet.

Implementation

Your will need to write Javascript to implement your functionality for part 2. You may also need to modify some HTML or CSS code. Some requirements for your implementation are as follows.

- All requirements on your HTML and CSS code from project part 1 specification remain in place.
- Most of your Javascript should be contained in external .js files. You may use a small amount of Javascript inside an HTML page for page-specific functionality, such as attaching Javascript code to the onload/onsubmit/… events of HTML elements.

- A **larger than normal amount of comments** are required to be included in your Javascript code to demonstrate to your tutor (assignment marker) that you thoroughly understand the meaning of everything that you are using.

Server deployment

You will need to deploy your project on a live website. In particular, you will need to:

- Set up an account on Amazon Web Services.
- Set up an Amazon EC2 virtual server running Linux. I recommend a "t2.micro" instance, which should be powerful enough for what you need, and which is included in the AWS Free Tier (see below).
- Install web server software and database software on your virtual server. I recommend Apache (with mod_php installed) and MySQL.
- Enable SSL/TLS on your web server. I recommend using Let's Encrypt to get your certificate and automatically set up SSL/TLS.

We will provide some instruction sheets to help you with most of the above tasks.

The services required are all part of Amazon's "AWS Free Tier": new AWS customers receive a variety of services for free for the first 12 months, so if you have never been an AWS customer before, you should be able to obtain these services for free. However, if you are already an AWS customer and have used up your free tier services, you may need to purchase these services yourself. I estimate the cost to be approximately $30 or less. If this expense will cause you to face financial difficulties, please contact the instructor to make alternate arrangements.

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

There are no add-on programming tasks for part 2.

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 Javascript.

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 Avenue discussion group. As a last resort, you can email me (doq4@mcmaster.ca) with a question or to request a consultation meeting.

Submission Instructions

For part 2 of the project, you will upload a ZIP file containing your web site to 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.
  - An explanation of any unusual choices, or things you feel we should know about your project.
- All HTML, CSS, and Javascript for your website. You may organize these additional files into subfolders, as you like. Assuming you successfully set up your live server, 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.

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

You should limit your ZIP file to 30 MB.

If you are uploading a sample video, it only needs to be a few seconds long to demonstrate that it works, so you should be able to keep it small.

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.

Note that the Geolocation API may not work correctly on Chrome-based browsers when you are loading pages from the file:// scheme.

We will test your website at least on the most recent version of **Firefox** 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:

- 15 marks for the core functionality
- 5 marks for the quality of the Javascript code
- 5 marks for deployment

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

Mapping API

You may use any Javascript mapping API you wish. Three popular ones are:

- **Google Maps:** https://developers.google.com/maps/documentation/javascript/
  Widely used, with lots of example code available.
- **OpenStreetMaps:** http://wiki.openstreetmap.org/wiki/Deploying_your_own_Slippy_Map
  An open-source map. OSM doesn't directly provide a Javascript API, but there are several third-party Javascript APIs for OSM:
  - **OpenLayers:** http://openlayers.org
  - **Leaflet:** http://leafletjs.com/examples/quick-start/
- **Bing Maps:** https://www.microsoft.com/maps/choose-your-bing-maps-API.aspx

Many mapping API providers require you as a developer to sign up for an account and obtain an API key in order to use their API on your website. API keys typically have page view limits associated with them, but your project will not exceed any of the providers' free tiers. You should not need to provide any billing information to any of the mapping providers above. It is your responsibility to comply with the terms and conditions of any mapping service you use.

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.

Extensions and 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.

1. *Can I use AngularJS / YUI / Bootstrap / <insert favourite HTML/CSS/JavaScript framework here>?*
   ~~No.~~ Y**ES! While** web development frameworks are a very dynamic area, and it is very likely that the frameworks that are popular today will be abandoned or obsolete in just a couple of years; frameworks as a concept will still remain. As noted above, it is important for you to learn how to write your own code, and once you have those skills, you will be able to work with appropriate frameworks when the need arises. (If you have a very good reason to use a framework to do something super awesome in a way that doesn't defeat the purpose of the assignment, you can propose it to me and I will consider it.)

2. *Can I use jQuery?*
   Yes. ~~However, you may only use the core jQuery API, and not any additional jQuery projects such as jQuery UI, jQuery Mobile, or any jQuery plug-ins/add-ons.~~ If you have used jQuery in the past, you may be surprised to know how much Javascript has improved in the last few years, and many things that jQuery was used for in the past can now be accomplished using plain Javascript (http://vanilla-js.com).
3. *Which mapping APIs can I use?*
   See the section on mapping APIs above.