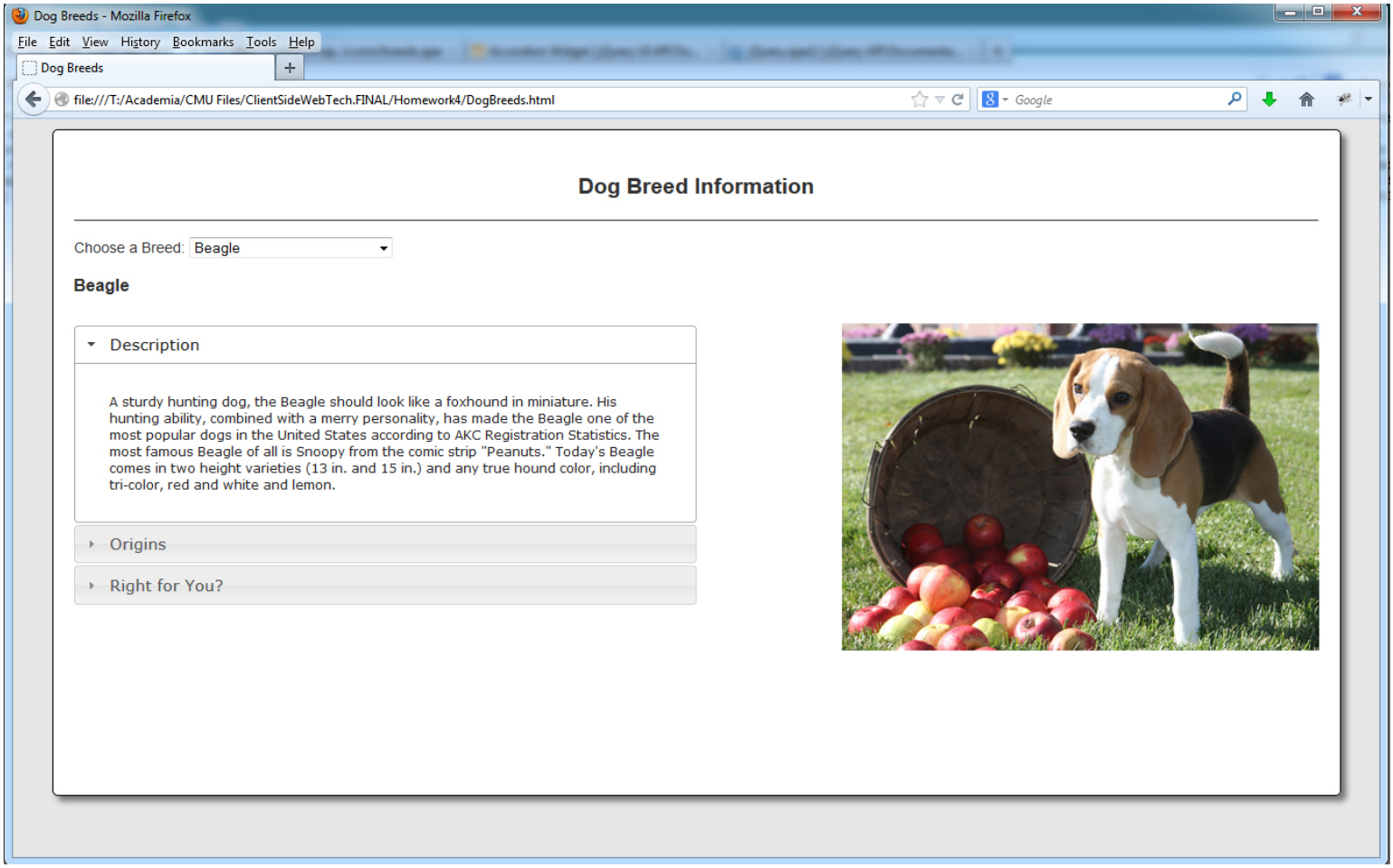
Client-Side Web Technologies – Homework Assignment 4

Due: March 5 @ 11:59 PM

Value: 15% of total grade

Overview

In this assignment you get some practice using jQuery, the jQuery UI plug-in, AJAX, and JSON. You will use these technologies to create a very simple yet useful webpage that gives information about several different dog breeds. Here is a screenshot of how it should generally look:

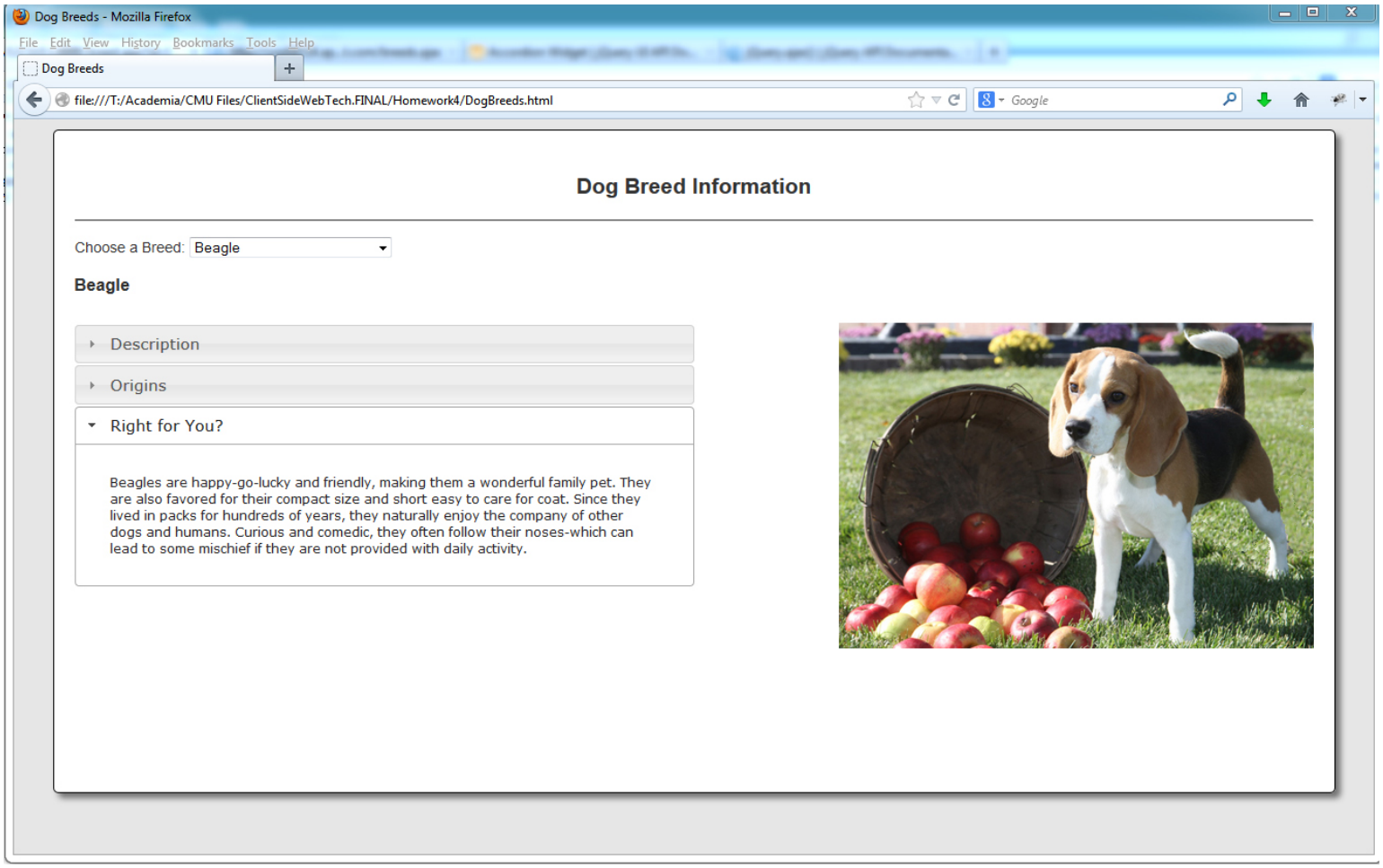


Feel free to use whatever fonts and colors that you want. Unless it would conflict with one of the requirements that are stipulated in the assignment, you are free to get as creative as you like with CSS.

The page must contain a select element, as shown, that upon loading of the page must be populated with data retrieved via an AJAX call described in the next section.

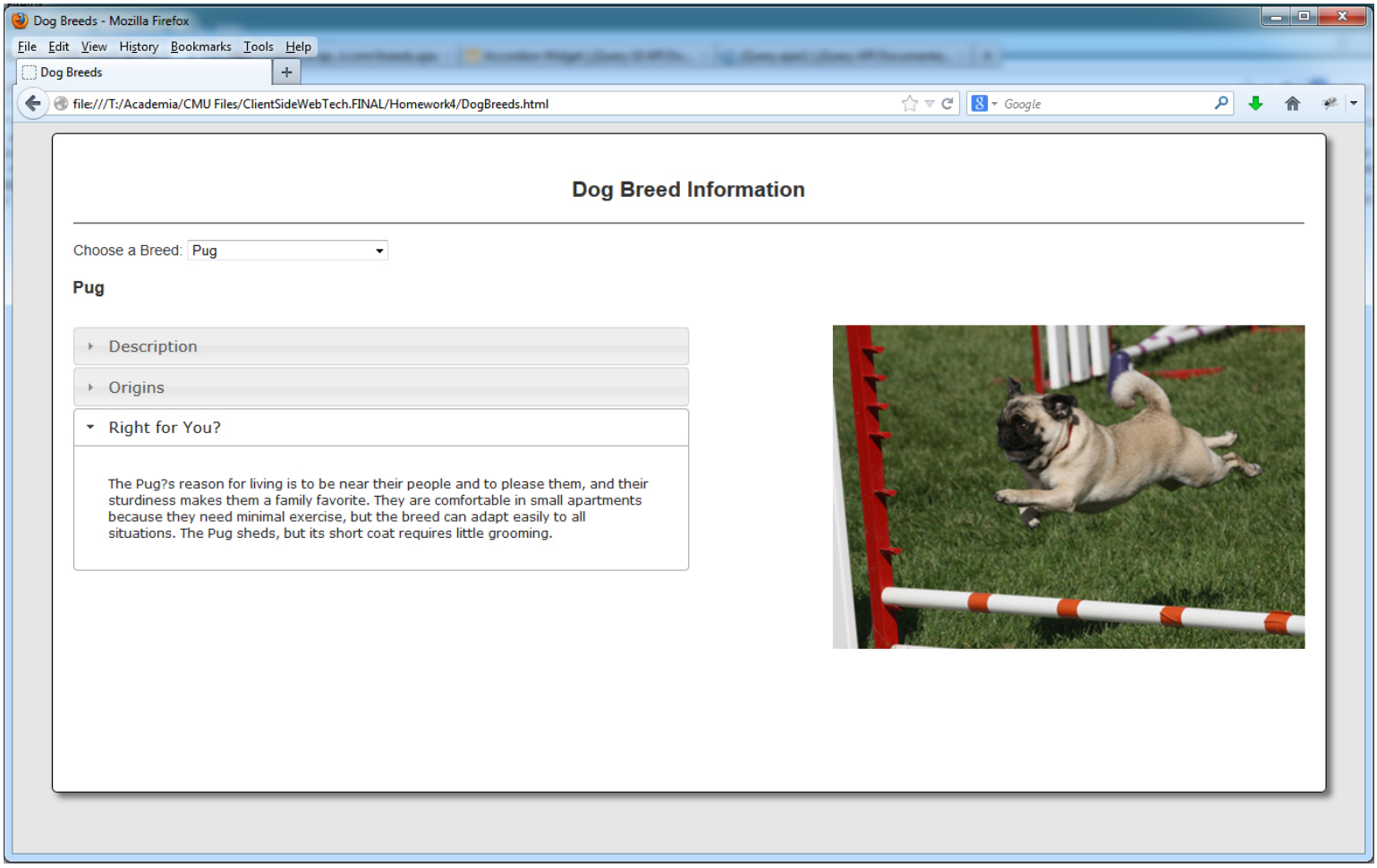
When the user selects a breed using the select element, another AJAX call is made to obtain breed information that is used to update the jQuery UI accordion widget's data as well as the breed name and image.

The accordion widget should expand different sections as they are clicked:



Notice how the “Right for You?” section is now expanded so the other sections collapse. The jQuery UI accordion widget should take care of all of this for you so it should not be too difficult to make use of it in your page.

Selecting a different breed will update all three sections in the accordion as well as update the breed name header and image:



When selecting a different breed, the currently expanded accordion section should stay expanded (i.e. it should NOT reset to the initial state that it had when the page first loaded).

Requirements

There are several requirements. These are outlined below.

Layout

Your page should be laid out similar to how it is shown in the screen shot. There is a heading at the top titled “Dog Breed Information”. There is a select element under the heading with the label “Choose a Breed”. Under that is where the breed’s name should be shown. Under the breed name is the accordion widget with three sections – “Description”, “Origins”, and “Right for You?”. To the right of the accordion is the breed image. Your layout should not be affected by resizing of the browser viewport (with the exception of the left and right margins of your main section).

Populating the Select Element

The select element must be populated with the option data received from the following URL:

http://csw08724.appspot.com/breeds.ajax

A GET request to this URL will return JSON. The JSON will be a serialized array of objects. Each object contains two properties: “id” and “name”. The “id” property is a number that represents the id of the breed. The “name” property is a string that represents the name of the breed. Here is a subset of what the JSON looks like:

```
[
  {"id":1,"name":"Beagle"},
  {"id":2,"name":"Vizsla"}
]
```

You must use this JSON data to create the options for the select element. You must make an AJAX call to do this whenever your web page loads. Do NOT hardcode the JSON into your solution. Your code should work even if I add more breeds (which I WILL do to evaluate your assignment).

Upon initially loading the select element options, you must make an additional AJAX call to fetch the data for the first breed option (as if a user had selected the first breed). This is discussed in the next section.

Selecting a Breed from the Select Element

When a user selects a breed from the select element (or in the case of the page initially loading) you must retrieve the necessary data from the following URL and update the page accordingly:

http://csw08724.appspot.com/breed.ajax

Notice that it is very similar to the one above but uses “breed” instead of “breed**s**”. There is an additional requirement to this URL as you MUST provide an “id” query string parameter. The id used will be one of the ids from the select element. For example, to retrieve data for the Beagle one would make a GET request to this URL:

http://csw08724.appspot.com/breed.ajax?id=1

A GET request to this will return a JSON object. This object will have several properties:

- a “name” property that is a string to use to update the breed name header below the select element
- a “description” property that is a string used to update the content of the “Description” accordion section
- an “origins” property that is a string to update the content of the “Origins” accordion section
- a “rightForYou” property that is a string to update the content of the “Right for You?” accordion section
- a “imageUrl” property that is a string that is the relative path to the image on the server to use for the breed image source
- a “extraImageUrls” property that is an array of strings representing relative paths to additional images on the server for the breed (**this is only used if you are attempting the bonus option**)
- an “id” property that is a number representing the breed id

Here is what the JSON object from the above URL would look like:

```
{
  "name":"Beagle",
  "description":"A sturdy hunting dog, the Beagle should ...",
  "origins":"In the 1500s, most English gentleman had ...",
  "rightForYou":"Beagles are happy-go-lucky and friendly ...",
  "imageUrl":"img/beagle.jpg",
  "extraImageUrls":["img/beagle_2.jpg","img/beagle_3.jpg","img/beagle_4.jpg",
    "img/beagle_5.jpg"],
  "id":1
}
```

The image path is relative to "http://csw08724.appspot.com/" so the full URL for the beagle image is http://csw08724.appspot.com/img/beagle.jpg. The same is true for the extra images.

The Accordion Widget

Documentation for the jQuery UI accordion widget can be found at:

http://jqueryui.com/accordion/

As stated above, when the user selects a breed the currently expanded section should stay expanded. When the page first loads the "Description" section should be expanded. Also, since you will be dynamically changing the content inside the accordion sections, you may need to consult the accordion's API regarding how it determines section height. A requirement of this assignment is that each accordion section is large enough to display its content without scrollbars.

Submission

To submit your assignment, add all of your files to a ZIP archive, name the ZIP file ***Homework4_[andrewid].ZIP***, and upload to Blackboard under Assignment 4 by the due date/time above. For example, my ZIP file name would be ***Homework4_jmussits.ZIP***.

Grading Rubric

This assignment is worth 150 points (15% of your total grade). The following is how the assignment will likely be scored:

- The page is laid out correctly **[25 points]**
- The select element is populated correctly when the page loads **[25 points]**
- The breed name header is updated correctly when the page loads and when a breed is selected **[15 points]**
- The "Description" accordion section is updated correctly when the page loads and when a breed is selected **[15 points]**
- The "Origins" accordion section is updated correctly when the page loads and when a breed is selected **[15 points]**
- The "Right for You?" accordion section is updated correctly when the page loads and when a breed is selected **[15 points]**
- The image is updated correctly when the page loads and when a breed is selected **[15 points]**
  The accordion widget is implemented correctly, including section heights done correctly as well as expand/collapse logic done correctly **[25 points]**

Bonus Option

This assignment offers an optional **15 point** bonus and is described below. The 15 points will be added to your assignment 4 score even if you achieve the maximum possible points, so you could get a **165/150** for this assignment.

To complete the bonus option you must use the image as well as the extra images for a given breed to create a running slideshow of the 5 different images (1 + 4 extra).

So, instead of just updating the image using the 1 image contained in the "imageUrl" property when a breed is selected (or the page is first loaded), you will use all 5 images to create a slideshow. Every 5 seconds (timing is of course not guaranteed in Javascript but don't worry about that here, just try for 5), the image should change. The slideshow should cycle through all 5 images and start over. The order of the images is up to you. The slideshow should continue until another breed is selected, in which case the slideshow would start playing the images for the newly selected breed.

Be sure to use some sort of animated effect to transition between pictures. The type of effect is up to you. You could accomplish this in various ways. For example, you could use CSS Transitions or you could make use of some of the jQuery UI effects.

Cycling through the images as described is worth **10 points**. Animating the transition between pictures is worth **5 points**.

A demonstration of how the bonus option could be implemented successfully will be given in class on Wednesday, February 26th.