CS 336 – Web and Database Applications

Midterm

Fall 2016

Due Monday, October 17

Dr. Pyatt

Objectives

1. To demonstrate your knowledge of HTML, CSS and JavaScript as it pertains to Web design and development through the creation of a prototype for an interface, webpages and web site.
2. To demonstrate your knowledge and application of essential programming elements like: variables, functions, methods, events, classes, cookies, and objects.
3. To demonstrate your ability to problem solve, design, test and create an innovative solution to a coding challenge.

Task

1. Build Interface (see below)
2. WebSite Design (3-5 pages)
   1. Example of Site Structure
      1. Application Tree
         1. index.html
         2. CSS Folder
            1. .style
            2. .style
         3. JS Folder
            1. .js
            2. .js
         4. Img
            1. Background1.img
            2. Backgroung2.img
         5. Video (use video format that is most compatible across all web browsers)
         6. Other (as appropriate)
3. WebPage Design (see below)
   1. Build a simple 3-5 page web site that includes the following:

Interface

* Interface must include essential HTML, CSS and JS elements and interactive design. For example, interface can include *buttons*, *images*, *video*, *audio*, *tables* and other *essential* features.
* Interface must incorporate responsive design (i.e., responsive in terms of user interactions with objects). For example, there should be an option to change the form or style of the interface through clicking on a button, or through certain conditions being met for a given event (e.g., if the day is Tuesday, the interface will change the background color, or image). An example of a simple interface was the one we studied and built in class called VirtualNote.

Interface Examples

1. Create a simple game which runs on one of your pages (e.g., puzzle game, quiz game, tic-tac-toe, hangman, etc…).
2. Create a simple interface which functions as a music player.
3. Create a simple interface which functions as a daily journal or note capture application.
4. Create a simple interface which functions as a personal calendar.
5. Create a simple interface which performs calculations (e.g., given certain variables, makes a calculation give a set of conditions). For example, days until your birthday, days between dates).
6. Create a simple interface which functions as a weather application which pulls in real-time weather information that is customized for you.
7. Create a simple interface that another person you know would find useful. For instance, maybe you have a friend who needs help with a math class, they don’t understand matrices. You could create a simple application which helps them build matrices to solve trinomial equations.
8. Create a simple interface which helps keep track of bills. For instance, there could be functions where you create “bills” and enter certain criteria. The application has temporal awareness in that certain reminders are provided based on the day of the week or date.
9. What ideas do you have at this point?

Pages (Home, Interface, Design)

1. Home Page
   1. Project Name
   2. Background
   3. CSS Style and Theme
   4. Buttons, links, text and other essential features which make this an inviting and responsive space.
2. Interface (see details above)
3. Design Page - your design page will need to include the following:
   1. Abstract of your application/project
   2. Design Goals
      1. To build an interface which can be used to …
      2. To use storage in the form of cookies to …
      3. To use JavaScript to …
   3. Design concept (this is a simple diagram of your envisioned site/project). This diagram should show relationships between pages. You should also have a structure you share that describes the envisioned functions, methods, events, variables, and other programming essentials specific to your design concept.
   4. Description and examples of HTML Tags (at least 3) that were used in your design.
   5. Description and examples of CSS used in your design.
   6. Description and examples of JS used in you design.
   7. Description and examples of extensions (code snippets, methods or functions) which were used in your design.
   8. Testing
      1. Alpha – this is where you debug and work through your code. Document issues that arise as well as those you have resolved.
         1. Issue 1
         2. Issue 2
         3. Issue 3
      2. Beta – this is where you get feedback from potential user/friend/family member or me, regarding your design (i.e., +, -, delta).
         1. A paragraph about how your test was conducted.
         2. Recommendation 1 and Design change
         3. Recommendation 2 and Design change
         4. Recommendation 3 and Design change
   9. Maker’s Mark video
      1. This is a simple 1-2 minute video you create which illustrates and describes the site and project. In this video you can discuss the essential aspects of your project and design which demonstrate your knowledge of HTML, CSS, and JS.

Required Elements

Cookies/Storage

You will need to utilize cookies for your site to capture user input or data in a way that your site page(s) respond to the input data. Your cookie functions must demonstrate: set; check; get; as was studied in class.

HTML Tags

You will need to utilize, at a minimum five essential HTML 5 tags throughout your application design. In particular, you will need to use <iframe>, <table>, <id\_>, <a href>, <div>, <button>. For an exceptional project, you will need to demonstrate, at a minimum, ten tags used throughout the application.

CSS and JS

You will need to create a CSS folder which holds, at a minimum, two .style files which are referenced in your application. These are your external CSS files. You will also need to include in your application pages internal styling as appropriate.

You must develop at a minimum, two .js files which are stored in the JS folder, and get called from one of your pages to accomplish something useful (e.g., you can have a JS script file which, when invoked from your interface changes a behavior, fires a function or event, gets/sets/stores a variable).

Bonus – build a CSS page template which can be applied to other pages on your site. Incorporate bootstrap functionality, and/or other publically accessible libraries to add responsive design to your pages.

Classes, Objects and Variables

You will need to utilize classes, variables and objects throughout your application design. For example, you may create a button class, or a divider class, for which you control behavior or properties through calling the element id.

Functions, Methods and Events

You will need to create and utilize, in your application design, functions, methods and events. For instance, at a minimum, you will need to use *if statements*, *loops* and *variables* to control behaviors and properties of application and interface elements.

Extensions

Although not required, but recognized and encouraged for exceptional design, you may incorporate extensions in the form of JS libraries or JQuery libraries like bootstrap and font-awesome.

Attributions

It is expected that the application you design is exclusively your own code. That said; you may utilize existing functions or methods which you reference from an authoritative source. Be sure to document such uses.

