**GrC 338 Content Management and**

**Web Development**

**Spring 2016**

**Howard Vogl, Eric Johnson**

**Project Website**

Purpose

For this project you’ll design and publish a website for a mobile device using Adobe Brackets. Your website should contain at least four pages. The actual design of the website is up to you, but as a minimum requirement include the following pages or their equivalent.

* About Page
* Professional Interests Page
* Personal Interests Page
* Contact Page

*Important:*

*The purpose of this lab is to gain further understanding of the basic components of a website. Therefore, for this project you are not allowed to use a web development application like Dreamweaver or templates from any source.*

*It is expected that you will start with a blank html page. It is permissible to examine the design and code of other websites as inspiration for your work, but is not permissible to copy an entire webpage or a substantial portion of a webpage with or without modifications and submit it as your own work. Doing so will result in disciplinary action that may lead to expulsion. If you are unsure as to whether using attributes is acceptable ask the instructor.*

Content

The first step is to gather the necessary copy and images. *Remember—content drives design!*

**Sketch**

Start by sketching out the design of your pages on paper. This is a quick and easy way to establish your design. Consider that some elements such as the navigational system and footer should be in the same place for all pages.

**Wireframe**

Once you have completed your sketch create a wireframe of the pages using Photoshop or a similar tool. A wireframe will show the size and position of different elements on your pages.

Using Photoshop make a new file for the mobile page.

iPhone6 375px x667 (retina 750 x 1334). Feel free to use the dimensions of a different mobile device.

There is a sample wireframe in the project file named Mobile-Site-Design.psd

Your wireframe should include:

* Size and spacing of elements
* A color scheme
* Type selection for html elements

Once you have finished your wireframe create templates for your site

There is a sample template in the project folder named website-mobile-template

* Create a directory to work locally.
* In that directory make a new HTML file and name it **template.html**
* Link the supplied reset style sheet to your template page
* Make a new text file and name it stylesheet1.css and link the style sheet to the template
* Make a subfolder for images and place your images in that folder
* In the template page create a preliminary layout based on your wireframe.

Set up the basic structure for your pages by importing and marking up your content. For the template include design elements that would be common to the entire site such as header, footer, main content area, navigation system, etc. You can modify other pages as needed.

Building a navigation system

Creating a navigation system can be a complex task. The key is to start with an unordered list. To create the navigation menu put the pages you wish to navigate to in an unordered list.

****

Next is styling the menu. The website project example starts with a div that contains the element selector nav. Feel free to use this menu as an example; however, build yours from scratch.

When you view your menu in a browser you should see a clear structure of links. If not, fix before moving on.

Making your life easy

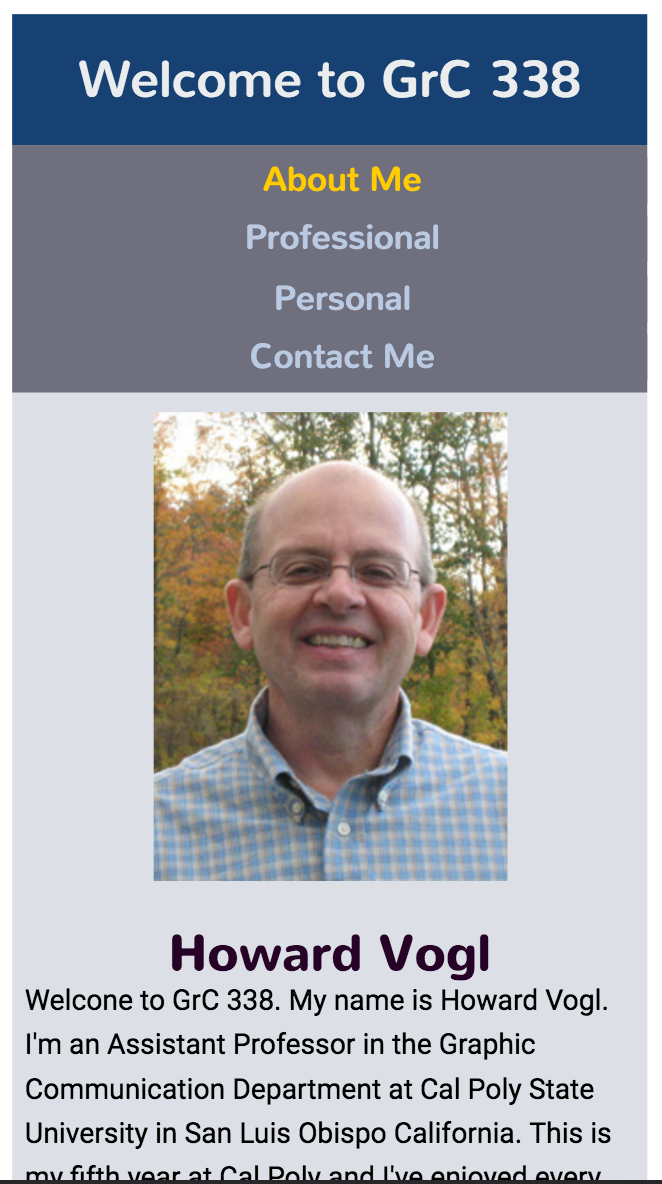
* A good way to save time is to have your template saved with the proper filenames in the list items section of the menu. For example, <li><a href="about.html">About Me</a></li>. This way when you make copies of the template and rename the pages the page links will already be created.
* Also, whenever you create an opening tag, immediately create the corresponding closing tag. If it’s a div tag immediately add a comment after the closing div. This will save you a lot of debugging later.

<div class=”tophead”>

</div><!--end of tophead div-->

* In addition comment and indent your html and CSS.
* Another way to save time is to continually validate your work as you go.

When your template is completed validate the file before making any copies!

GitLab

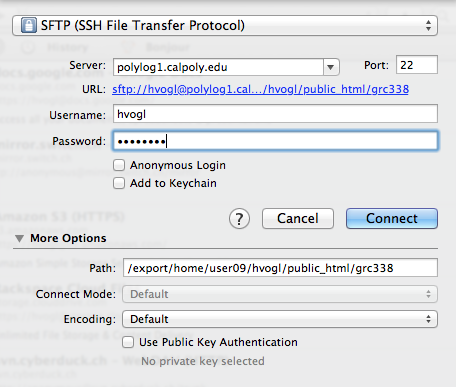
Using Git will be a portion of this project.

Create a local Git repository and link it to the remote host using the URL provided by the instructor. Make commits and push them to the remote site. Make sure that any work you commit is related and fully tested before making a commit. Also, don’t forget to include short descriptive messages with every commit.

* Make copies of the template and rename each page according to proper html file naming conventions and change the title of each page. Remember that your home page will be named index.html.
* Completely check your website to make sure you can navigate to each page and return home in one click.
* Then when you are satisfied modify the content on each page.
* Elements such as text and images will change from page to page. However, the navigation system should remain consistent throughout your website.

Uploading your website to the web server

* Launch Cyberduck
* Open Connection
* Set connection type to: SFTP (port 22)
* Server: polylog1.calpoly.edu
* Enter: Username and Password
* Connect to server



Upload the contents of the local folder.

To view

* Open your web browser and enter the URL to your personal site
* For example

<http://www.calpoly.edu/~hvogl/grc338/>

**Validate by URI**

* When your project is uploaded and you can view it on the server, validate it using the W3C validator. Validate by URI tab.
* **This is how I will check your page**

**Grading Criteria**

To be graded I must be able to link to your website through the web server by URL.

If your home page doesn’t open from your URL you will not receive credit for the project

Meaningful content and good design 10 to 0

* Deductions for lack of meaningful content and poor design simply copied from the demo files.
* Minimum required pages 10 to 0

Unique Images on some pages 10 to 0

* Appropriate resolution & file format 10 to 0

Functional navigation system

* All pages are linked and user can return home in one click 20 to 0

Page markup validates by URL 10 to 0

(-2 for each error)

Good use of CSS to layout pages 10 to 0

Code properly commented and indented 10 to 0

Use of GitHub host 10 to 0

Late -5% per day