INFO/CS 2300: Project 1

Build a website with PHP and promote something.

Overview

Project 1 is your chance to demonstrate your knowledge of HTML, CSS, PHP, and to give us a chance to get to know you as well. Your first project is to build a small 3-4 page website about something about which you are interested or passionate.

Learning Objectives

- Refresh skills and knowledge from INFO/CS 1300.
    - Refresh and practice your visual design skills.
    - Refresher for prior HTML, CSS, and PHP knowledge.
    - Brush-up on web programming standards, conventions, and this class' expectations.
- Improve visual design skills through the use of personas and wireframes.
- Reduce redundant HTML code using PHP and functions.

Deadlines & Receiving Credit

Milestone	Points	Grade	Deadline
Project 1, Milestone 1 (p1m1)	10	Feedback	Tues. 2/13, 5:00pm
Project 1, Final (p1final)	100	Rubric	Tues. 2/27, 5:00pm

Slip days permitted for all deadlines. Maximum of 2 slip days per deadline.

Submit all materials, including designs and reports to your GitHub repository for this assignment.

Description

For Project 1 you'll build a 3-4 page website to promote a topic of your choice. It's probably best to pick something you find interesting or are passionate about. It can be something that you do as a hobby, something related to your academic career, or something you just find enjoyable. Or it can be just plain silly as long as you meet the requirements. Your site should include some basic information about what you are promoting (why you're promoting it), and images to describe the thing that you are promoting. Examples include (but are not limited to) a sports team, clubs, organizations, individuals, items (technology, fashion/clothing, book), photography, music, yourself as a person, etc.

While this course is mostly about using server-side technologies, we're still interested in your ability to create visually interesting and user-friendly websites. You are expected to apply basic web design theory, such as the visual design principles you learned in 1300 (contrast, repetition, alignment, proximity, etc.). Your web page should be organized appropriately using the information architecture skills you also learned in 1300. Your site must be coded in valid HTML, CSS, and PHP. You are expected to include most of the PHP elements we've covered in class. However, it's up to you to determine how you will incorporate these elements.

This project has two milestones: Milestone 1 and the Final submission. For milestone 1 you'll design your web page using sketches and wireframes inspired by personas. You'll also compose a plan on how you will program your website. For the final milestone you'll code up your website based on your designs and coding plan. Your final website should be polished.

Tip: Read this entire document before starting Milestone 1 so that you understand the all of the requirements.

Requirements

1. The site should have 3-4 pages. You can have more than this, but we won't necessarily look at more than 4.

2. The site should be well-designed, have a consistent "look and feel," and have clear navigation.

3. You should thoroughly plan your design using sketches and wireframes. Your designs should be informed by the provided personas.

4. Your site should display reasonably well (not necessarily identically) across Firefox and Chrome. We will check your website on any one of these browsers.

5. The site should have at least 3 images.

   - The images should have a reasonable file size. Scale the images to reduce the file size if necessary. The GIMP is one such tool.
   - If you took the photograph or created the illustration, it must be noted in a comment in the HTML. If you acquired the image from any other source, a credit must appear near the image. When the scale of the image makes this impractical, a credit at the bottom of the page is acceptable. See the **Media Credits** in the syllabus for additional details.

6. You should create a plan using pseudocode before you write any code.

7. The site must demonstrate the PHP concepts we've covered thus far.

   At a minimum this includes:

   - Effective and appropriate use of variables.
   - Use of a least one array.
   - Use of a least one user defined function (see requirement below).
   - Use of a least one conditional expression and statement.
   - Use of a least one loop.

8. Your site should include at least one non-trivial user defined PHP function that accepts input, then displays the output in a user-readable way (table, HTML formatting tags h1, h2, etc.) or returns a result for use by the code that

called it. Examples include the output of repeating elements in HTML forms or lists or making a particular calculation. Make sure any PHP code that generates HTML properly sanitizes that output (i.e. `htmlspecialchars()`). The graders will be looking for the XSS vulnerabilities you learned about in 1300.

9. Every page that a user can access must be validated (i.e the stuff you see in the browser). This means that files that you include in the user-accessible pages with PHP, such as header.php, do not need to validate on their own.

10. Your code should be well formatted and readable. It should follows the standards, conventions and expectations of this class. See Lab 1 for an example.

    Tips:

    - Use indentation to keep nested content clear.
    - Keep your code efficient, neat, and organized so that the TAs can easily read it and understand it.
    - The usage of explanatory comments in code is encouraged and expected.

11. All code must be your own work for this project. You may not use code from other class, including INFO/CS 1300.

12. Your website should be under 10MB unless you have an exceptional reason. Reduce the size of your images if necessary. Do not use Git LFS. The grading staff will not support or grade any Git repositories that use Git LFS.

13. The grading staff will grade using the PHP server in Atom. If you use other servers with different configurations, like MAMP, you do so at your own risk. These differences in configuration can sometimes cause errors.

14. You may assume that your web page will be viewed in a desktop browser. You do not need to design a web page for mobile browsers. You do not need to use media queries.

Coding Standards & Design Guidelines

These are the standards, conventions, and expectations introduced in Lab 1. You are expected to follow these for all assignments this semester.

Design Guidelines

- A multi-page site should be well organized and include proper navigation.
- Have effective information organization and navigation structures.
- Use visual design principles well, and have an engaging, pleasing design.
- Designed and implemented effectively for the target audience of the site.
- Follow the rules of good usability from the user's perspective.
- Your design should work on different screen sizes and display reasonably well across different browsers.
- External content is cited. See the syllabus for details.

Coding Standards

- All code is your own work, unless the assignment states otherwise.
- Use Validated HTML5 and CSS3. You must have 0 errors; warnings are permitted.
- Main page is named index.html or index.php.

- The HTML is well structured for your site's content (i.e. use of `<header>` , `<main>` , `<section>` , `<footer>` , `<p>` , `<ul>` , `<em>` , `<strong>` tags)
- Use only CSS for positioning and style.
- External styling via CSS. **No inline or internal styling** (i.e. `<style>` or `style=""`).
- Multiple CSS files are okay as long it's for legitimate structural reasons.
- Your code (HTML, CSS, JavaScript, and PHP) is well written, formatted, properly indented, readable, and well-documented.
- Your code is documented with comments where appropriate.
- Images are located in an **images** directory. Scripts in the **scripts** directory. PHP includes in the **includes directory**. CSS in the **styles** directory. See Lab 1 for an example.
- No broken or dead links. Remember that some computers use case sensitive file and folder names!
- You have tested your website in Firefox and Chrome.

Milestone 1

For Milestone 1 you'll sketch your design and provide a wireframe for each page of your website. Your design should be informed by one of the personas listed in the personas folder. You'll formalize your design by authoring a wireframe for each page of your website. Lastly, you'll make a coding plan with pseudocode.

Clone your Assignment Repository.

Clone `git@github.coecis.cornell.edu:info2300-sp2018/YOUR_GITHUB_USERNAME-project-1.git` . Replace **YOUR_GITHUB_USERNAME** in the URL to **your actual GitHub username**.

Design & Planning Document

For Milestone 1 you'll submit a Design & Planning document: **design-plan/design-plan.md**. This file should be in the *design-plan* folder of your assignment repository. You are required to submit this file in Markdown format.

Markdown is a popular mark-up language. It's a good idea for you to learn it, especially since it's used for documentation for many programming projects. All of the course documents this semester are written in Markdown. Please feel free to review any of these documents as examples.

For your reference, here's a few references for Markdown:

- https://guides.github.com/features/mastering-markdown/
- https://github.com/adam-p/markdown-here/wiki/Markdown-Cheatsheet

When writing your Markdown file in Atom. Open the command palette and search for **Markdown Preview: Toggle**. This will open up a panel in Atom where you can preview your formatted Markdown file.

1. Personas

You have 4 personas to choose from located in the personas folder. You may pick any persona you like. However, we **strongly recommend that you pick Abby** for reasons we'll discuss in lecture throughout the semester.

- **Abby** - personas/abby.pdf
- Patricia - personas/patricia.pdf
- Patrick - personas/patrick.pdf
- Tim - personas/tim.pdf

Make a note of your selected persona in your Design & Planning document: **design-plan/design-plan.md**.

2. Sketches

Before you code anything you should design your web page first. Sketching is the best way to jump start this process. Sketches are quick and help you generate ideas that will improve your overall design. You should sketch on paper. Sketching using a computer program takes too long and misses the point of sketching. When sketching, keep your selected personas in mind. You'll want to make sure that you're designing your web page for your persona.

For Milestone 1 create a few sketches. This should probably be around 2-3 sketches. You're welcome to submit more if you like. Your sketches don't need to be polished. The lines don't need to be straight. You don't need to write text; squiggly lines are okay to *represent* text. You should only use one color: black and white. The idea here is to generate ideas.

Take a picture of your sketches (I recommend the Microsoft Office Lens app) and place the image files in the *design-plan* directory and link them in your Design & Planning document. Refer to the example image if you're unsure about how to include images in Markdown.

3. Wireframes

After you've generated some ideas with some rough sketches, it's now time to start to finalize your design. You should generate polished wireframes for each of the pages of your web site. When authoring your wireframes, always keep your persona in mind.

Wireframes should be higher fidelity than your sketches, but should not be full mock-ups. This means you should still create your wireframes on paper. Avoid authoring your wireframes in tools like Illustrator, Photoshop, or Balsamiq. Wireframes are one color: black and white. Wireframes should probably have more text than your sketches, but it's still okay to use some squiggly lines. Sketches don't usually take much care for spacing, alignment, and white space; wireframes should make your design decisions for spacing and alignment very clear. This means you need to take some care when authoring your wireframes. Your lines should be pretty straight. Your should design should probably fit neatly onto a grid as we discussed in 1300.

Your wireframes should make it very clear what your final website will look like. You should take a photo of each page's wireframe and include it in your Design & Planning document.

4. Coding Plan & Pseudocode

After you have concrete idea of what your web site will look like, it's time to plan out coding your website.

You should plan out your main php pages and what *includes* you'll use to reduce redunant code (e.g. *header.php*, *footer.php*, etc.).

You should also plan out the code for your user defined PHP function using pseudocode. Recall that your site should

include at least one non-trivial user defined PHP function that accepts input, then displays the output in a user-readable way (table, HTML formatting tags h1, h2, etc.) or returns a result for use by the code that called it.

Place your coding plan and pseudocode in your Design & Planning document.

Grading - Feedback

10 points.

This is a feedback milestone. Your grade will be based on whether you submitted a complete assignment. If it looks like you tried, even though there are some mistakes, you'll get full credit. If your submission is obviously incomplete, you'll get a 0.

We'll provide feedback on your milestone to help guide your work for the final submission. **Our feedback is designed to help you learn more**; our feedback is not a "pre-grade". This feedback is designed to catch large problems (which we sometimes miss). **This does not resolve you of the responsibility of meeting the project's requirements, even if we miss something.**

Your feedback will show up as comments for the assignment in CMS. Please read it and use it to improve your learning and your project.

Submission

See **submit-*m1*.md**

Late submissions will not receive feedback until a least one week after the deadline. Using slip days has consequences. Not receiving timely feedback is one of these consequences.

Final Milestone

For the Final Milestone you will code the web page you designed and planned in Milestone 1. Your entire web page should be authored in HTML, CSS using PHP. You may optionally use JavaScript, however it is not required.

Grading & Submission

The final milestone is graded via Rubric.

See **submit-*final*.md**

Rubric

100 points.

I reserve the right to change this, but this should be close to the final version.

1. **Planning & Design (40%)**

   - Complete and thoughtful Design & Planning document.
   - Design is informed by a persona.
   - Sketches used to generate ideas.
   - Quality wireframes for each page. The wireframes *generally* align with the programmed web site.
   - The site has an appropriate information architecture; the site has a clear navigation structure.
   - The design follows visual design principles.
   - The design pleasant and polished.
   - Color and design support the content of your site in your delivering the site's message.

2. **Content (10%)**

   - Does the site have at least 3-4 pages of content?
   - The site should have appropriate and complete content. No placeholders allowed.
   - Are there a reasonable number of images on your site? (At least 3 across your site, you can have more if you wish. Not all pages need 3 images, but you must have 3 images divided among your pages. No credit if the images are not properly credited.)

3. **PHP Interactivity (40%)**

   - Site effectively utilizes PHP to reduce redundant code (i.e. *includes* and functions).
   - Does the site make use of the following PHP elements? (At least one of each)
     - Variable
     - Array
     - Loop
     - Conditional Expression and Statement
     - A non-trivial user defined function (you cannot simply wrap standard PHP functions in a method declaration or implement a method that increments a variable; it should do something more than that.)
   - How well does the PHP work? Are there any PHP errors or warnings?
   - Does the PHP functionality fit with the site?
   - All code is your own work for this assignment. Code written by others or in a previous class is not permitted.
   - All PHP generated HTML is properly sanitized.

4. **Coding Standards (10%)**

   - The code follows the standards, conventions, and expectations of this class. (see above)
     - Does the code validate?
     - Are the HTML, CSS, PHP, and JavaScript (if any) well-formatted, commented, and readable?
     - External styling via CSS. **No inline or internal styling**
     - Are the files well organized? (styles folder, images folder, no redundant code)
   - Website renders correctly in Firefox and Chrome.
   - Entire website is less than 10MB. (Does not include design-plan folder)