COMP1536 Website Development Project Specification

Purpose

The goal of the team project is to design an interactive web site using industry best practices. You will be applying website development principles and techniques taught in this course, and developing the website from scratch using HTML, CSS, Javascript and PHP. The final site should be professional in its presentation and quality.

Content

The content of the website can be on any topic that is of interest to the team as a whole – hobby, sports, family, place, clubs, or even a real company. In addition to providing users with static information, the website will incorporate some dynamic features as described below.

Requirement

The website must meet the following requirements:

1. Design and coding requirements
    a. Have a consistent site-wide design and presentation. Refer to lecture material and / or http://www.webstyleguide.com/ which has good tips on site design, user interface design, etc. You should incorporate these guidelines as much as you can into the design of the website.
    b. Provide clear and consistent menu navigation as well as a site map.
    c. Be HTML 5 standards-compliant. Your entire site must validate as HTML 5 at http://validator.w3.org/
    d. Use external CSS styling, including both screen and print styles. All CSS stylesheets must validate at http://jigsaw.w3.org/css-validator/
    e. Be accessible. You site must be error-free as evaluated by the WAVE accessibility evaluator at http://wave.webaim.org/ or some other equivalent tools

2. Basic functional requirements
    a. User Registration - Users must be able to register and deregister themselves on the website
    b. User authentication - Users will be able to log in to and out of the website. Once logged in, the site needs to display the logged on user's name somewhere on the page.
    c. Server side processing - The website must have a page for authenticated users to process information passed to the server, and return the processed data to the users. The project group can decide the server functions to be implemented. Sample code to allow users to post and comment in a forum will be provided. Students can either integrate this function with their website or create their own set of functions.

3. Project groups
    Teams of 4-5 students will be formed in D2L. The projects have been set up for students to self-register into groups. Prior to beginning work on the project, teams should meet to:

    a. Decide on a team leader responsible for coordinating the overall team efforts and guide the team toward timely completion of deliverables. The responsibilities of each team member, including the team lead, should be discussed and agreed upon by the team as a whole.
    b. Establish a shared goal in terms of the project mark. One person may be satisfied with a mark of 70% while others may wish for 95%. It is important to establish a common goal that everyone can commit to.

c. Establish meeting guidelines including punctuality, frequency, duration, expected preparation work for each meeting, documentation of decisions, etc.
d. Establish working agreements including mode of work (individually or in pairs), backup plans for failure to deliver by teammate, etc.

Responsibility and peer evaluation

All students in the team are expected to contribute their share of the work to complete the project on time and to a satisfactory quality level. There will be a mandatory peer evaluation component to this project. The evaluation will be used as a deduction factor to be multiplied to the final project mark for the student. Students who participate fairly will not have the final project mark lowered. No project marks will be increased after peer evaluation.

Project Milestones

There are 5 milestones for this project, any updates to due dates below will be announced in D2L, so make sure you check this site on a regular basis. All milestones (except code) are to be documented in a track change Word document. Milestone documents should be presented in a cohesive, professionally format with title page and table of contents. The documents should NOT be just in point forms and simply answering the questions listed in each Milestone requirement (below) but presented as a unified well thought-out document. Each milestone document submission (except the first one) should include previous milestones in an Appendix. All contents should be organized logically. Grammatical and spelling mistakes will be penalized. Milestone document should be in ONE document. Project files should be zipped up in a ZIP document with appropriate subfolders.

Each milestone starting with Milestone 3 must include a Quicklink to your website in the Comments Box.

Milestone #	Weight	Due (Friday 8.30am)	Goal
1	20%	Week 2	Topic, requirements, workplan
2	20%	Week 4	Site map, page design
3	20%	Week 6	Skeleton site with layout, tables and forms
4	20%	Week 10	Deployed Javascript-enabled site
5	20%	Week 13	Server side programming

Milestone 1: Project Proposal

1) Topic:
The project team will brainstorm and come up with a website to be created. Use team member experiences or research the web for ideas. Submit a 1-2 page single spaced proposal with appropriate section headings that addresses all of the following:
   a. What is the purpose of the site? What needs does it serve?
   b. What are the goals for the site? List the reason for creating the site (e.g. business reasons, communications, easy access, etc.)
   c. Who is the target audience? Age, demographics, etc.?
   d. What does the site have to accomplish? List the goals for the site.
   e. What type of content should be included? Describe the type of text, graphics, and media needed for the site.
   f. What are the success factors for the website?
   g. Evaluate and briefly critique two similar sites that already exist on the web and how you will take that consideration into your site.

2) Functional requirements
   a. One of the project requirements is to have a form included in your website for the user to submit input. Think of an appropriate form for your site (e.g. question form, membership signup, etc.) and propose it for your site.
   a. Another project requirement is to have a 'list' included in the content of the website (e.g. glossary, catalogue, etc.). Think of some appropriate content that would display well in a table.
   b. Don't forget to include item 2 under the Requirements section above.

3) Work Plan
   How will the team be working together to complete the project? What will the team do to make sure deadlines are met?

The proposal should be the result of thorough group discussion and thought, and written professionally.

Milestone 2: Site map and page design

Come up with a design for the website aimed at enhancing user interaction. Use MS Visio (or any other suitable prototyping tools) to document the following:

1. Project site map and navigational structure
   - Create a site map to show the hierarchy of pages and relationships between pages. Reference: Felke-Morris 8th edition page 207-208.
2. Page layout
   - Create wireframe page layouts for all the pages. Also, provide a wireframe page layout showing the print layout
   - Specify whether the layout will be fixed-width or fluid, and the type of column layout
   - Indicate where the logo, navigation, text and images will be located
   - Provide a sample or document the proposed color scheme for the site; use an online color guide preferably
   - Include reasons and best practices that you follow behind your design.

Submission:

- Revise any changes to previous milestones based on the feedback from your instructor with track changes enabled, and add to the end of this milestone document in an Appendix.
- Include the items listed above in this milestone document.

Milestone 3: Skeleton site with layout, tables and forms

In this milestone, you will start coding. However, before you start to code the HTML/CSS, do the following prep work:

1. Technical design
   - Project folder: Create a folder named after your TEAMCODE (e.g. G1 for Group 1). All your project files and graphics for your website will be organized in this folder and subfolders as needed.
   - Filenames: Review your sitemap and the pages you need to create. Decide on the filenames you will use for these pages. Filenames should be meaningful. The home page should be named **`index.html`**
   - Styling requirements: See guidelines at http://www.webstyleguide.com/wsg3/7-page-design/index.html, http://www.webstyleguide.com/wsg3/8-typography/index.html), and other course material. Referring to the page layouts you created for Milestone 2, decide on how the following are to be styled:

    - Fonts and colors to be applied across all the pages
    - Common elements such as headings, lists, paragraphs
    - Other elements such as logo, navigation, footer, etc. if necessary
    - Note that the main navigation should be configured as an unordered list. Use the CSS **link** and **hover** pseudo-classes for the main navigation hyperlinks and in-page hyperlinks
    - The table (from Milestone 1)
    - The form for user input
  - Class design: Design the classes, naming them by semantics (e.g. maincontent) rather than style (e.g. centeredtext). Use advanced selectors whenever it makes the styling design clearer. Remember the purpose of CSS is to separate style from content so that any subsequent style changes are localized to the style sheets.
2. External style sheet
  - Create an external style sheet named **`base.css`** in the **`style`** subdirectory. Document your design using CSS commenting.
  - Note the style should be consistent across pages by using **`base.css`**. If there is a justifiable need to have page specific styling, use separate css files without duplicating any of the classes defined in **`base.css`**. Do not use inline CSS.
3. Prototype page
  - Code a prototype page for your site based on the design documents. Apply classes and id's where appropriate, using external CSS to format content and layout.
  - Save and test the page iteratively. Modify both the webpage and **`base.css`** as needed. Test the page carefully using Firefox and IE, improving the page as needed. Make sure the page is fully tested before you start developing other pages.
4. Remaining pages
  - Using the completed prototype page as a basis, code the rest of the pages on your site including the appropriate html markup.
  - Update the relevant page(s) with the html **`form`** markup, using **`post`** with the **`action`** attribute invoking PHP server-side processing subsequently. For now, the **`action`** attribute should send the form data temporarily to: **`http://webdevfoundations.net/scripts/formdemo.asp`**
5. Testing and code submission
  - Conduct A/B testing and document the results to justify your design
  - Test your site thoroughly in both Firefox and IE. Make sure to validate all html and css files.

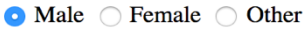
6. Submission
  - Revise any changes to previous milestones based on the feedback from your instructor with track changes enabled, and add to the end of this milestone document in an Appendix.
  - Zip up (don't rar) the entire website and submit it to D2L. The zip file should be named:

    `TEAMCODE _TeamLeader_Milestone#.zip`

  - A Quicklink in the Comments box in D2L the URL of your website
  - Include the following in this milestone document:
    - URL where all work completed so far can be viewed
    - List of items completed for this milestone
    - Any additional work on top of what is required for this milestone, if any
    - Key issues, including those outstanding, that was encountered in building the site
    - Any deviations from the proposal (Milestone 1) and design (Milestone 2) and why
    - Documentation of work
      - Screenshots of the front page as well pages containing the required display table and the form

        - base.css
    - A/B testing

Milestone 4: Deployed Javascript-enabled site – validated, tested

1. Define validation requirements
    - For each form, determine the input requirements for each field. Be detailed and thorough in the validation requirements; the same requirements will be used later for server side validation as well. For example, for 

, the validation requirement is that only three values are allowed. This requirement must subsequently be explicitly validated on the server as it is very easy for a malicious client (or user) to submit invalid data (see http://en.wikipedia.org/wiki/Cross-site_scripting).
    - Document each input field using this template:

FORM on page: _____		
Field ID	**Data Format or RegExp**	**Explanation**

2. Implement validation requirements
    - Implement client-side validation to the form(s) according to the defined requirements.
    - Validate the form field-by-field as the user enters the information, or upon submission, or a combination of both. Provide user-friendly feedback. Avoid alert boxes; provide context-sensitive feedback using highlighted fields or on-form error messages.
3. Test form
    - Ask a classmate to try using the form and observe closely. Note down questions, comments, mistakes, and hesitations and record them in the template below. Make any necessary improvements.

TEST DOCUMENTATION for FORM on page: _____		
FIELD LEVEL TESTING		
Field ID	**Problem**	**Improvements made**
FORM LEVEL TESTING		
Form Flow	**Problem**	**Improvements made**

4. Javascript or jquery third party widget
    - Find an appropriate open-source third party javascript or jquery widget that can be used to improve the users' experience of the website. Examples include data pickers, color pickers, photo displays, etc.
    - Document in the report why this widget was chosen and the advantages of doing so
5. Testing with Javascript disabled
    - Test the form(s) with Javascript disabled. While validation will not work, verify that submission still works properly.
    - Test with the inclusion of the widget in the step above that the form still works when Javascript is disabled.

6. Publish site and test
   - Include a statement in your milestone report on success/problems faced with the publish and test
7. Submission
   - Revise any changes to previous milestones based on the feedback from your instructor with track changes enabled, and add to the end of this milestone document in an Appendix.
   - Zip up (don't rar) the entire website and submit it to D2L. The zip file should be named:

     TEAMCODE _TeamLeader_Milestone#.zip

   - A Quicklink in the Comments box in D2L the URL of your website
   - Include the following in this milestone document:
     - URL where all work completed so far can be viewed
     - List of items completed for this milestone
     - 2 tables: validation requirements and testing
     - Any additional work on top of what's required for this milestone, if any
     - Key issues, including those outstanding, that was encountered

Milestone 5 – Server side programming

Use PHP & MySQL to implement the following:

1. User registration
   Users can register and deregister themselves on the website
2. User authentication
   Users will be able to log in and out of the website. You will be provided information on how to access a MySQL database with valid user id / password. Once logged in, the site needs to display the logged on user's name somewhere on the page.
3. Server side functions
   The website must have a page for authenticated users to submit information to the server for server side processing. The server functions should have been specified in previous milestones and implemented on the server side using PHP.
4. Submission
   - Revise any changes to previous milestones based on the feedback from your instructor with track changes enabled, and add to the end of this milestone document in an Appendix.
   - Zip up (don't rar) the entire website and submit it to D2L. The zip file should be named:

     TEAMCODE _TeamLeader_Milestone#.zip

   - A Quicklink in the Comments box in D2L the URL of your website
   - Include the following in this milestone document:
     - URL where all work completed so far can be viewed
     - List of items completed for this milestone
     - Any additional work on top of what's required for this milestone, if any
     - Key issues, including those outstanding, that was encountered

Presentation

- Prepare a 2 minute presentation on:
  - Major features of the website

- Major design of the website