420-D08-SU

Web programming

Practical Project Instructions

**Evaluation weight:** 30% of the final grade

The project must be done in teams of 2 people. If you don’t have a partner the teacher will match singles. If there is an odd number of students in the class, only 1 team with 3 members will be allowed. Each member of the team must produce 3 web pages through individual work. The site will be deployed on the school’s Sentora Web server. You will present your project to the group on the last day of the course. Both members of the group must participate in the presentation.

**Select a theme**

For your project you must select a theme fitting the guidelines below. Make sure your concept includes an offer of at least 6 different products or services. Example: web store selling products likes shoes, computers, cars, travel, … Web site describing services offered: web design/hosting/marketing services, show promoter with at 6 or more upcoming shows, gym or school with different classes like yoga, weight lifting, aerobic, …

**Keep it simple!**

Avoid grandiose ideas because of the time constraint for this project. Aim for a small project perfectly executed, with small and clean code files, and meeting all guidelines below, nothing less, nothing more. Extra pages and features won’t make a difference in the final grade.

**You must do your own HTML and CSS**

The goal is not to show how good you are at finding pre-made CSS and HTLM code elsewhere! Anyone can copy someone else’s code or even a whole web site. We can easily see if you copied large chunks of code just by the size and content of the files. Imagine instead you are working for a company producing new CSS libraries and original web site templates. You must demonstrate how good you are at crafting your own CSS styles and HTML. It is allowed to copy and modify simple examples from W3schools.com or exercises and examples shown in the class. In that case you must write your own comments in the files to show you understand what the code is for.

You cannot use technologies other than HTML5 and CSS3, **so no JavaScript, and no frameworks such as Bootstrap. The site does NOT need to be Responsive, only one desktop version needs to be designed.**

**Don’t get loss in styling**

There are points for the visual aspect of your site but the priority is clean code, well structured, without errors, and following all the guidelines below. The goal is not to show how good you are at finding awesome pictures, fancy animations, stunning backgrounds and the like. Again you should aim for a small project perfectly executed. Given the time allowed, simpler but very good looking pages typically leads to an higher score.

# Content and Structure Requirements

1. You must produce a website containing at least the following **6 pages**
   1. index.html - Home Page (Welcome Page)
   2. catalogue.html - Catalogue Page
   3. product.html - Product Page
   4. contact.html - Contact Page
   5. table.html - Table Page
   6. gallery.html - Photo Gallery
2.  **Home Page (index.html)** must contain at least:
   1. a header with a logo
   2. a main navigation pane
   3. multiple titles (main or subtitles)
   4. an introduction text
   5. at least 3 photos, showing product categories for example
   6. a substantial footer
   7. links to social networks
   8. a contact address
3.  **Catalogue Page** **(catalogue.html)** showing a list of products with thumbnail pictures must include
   1. at least 6 different products
   2. a small picture for each
   3. basic characteristics of the products (name, price, description, etc.)
   4. for one of the products, a link to the Product Page belowwhen clicking the picture
4.  **Product Page (product.html)** giving complete details of a ONE product. You don’t need to make a page for each of the 6 products. Choose 1 of the products to detail and make a product page for that one only. The Product Page must contain at least
   1. the name of the product,
   2. the price,
   3. a detailed description,
   4. other relevant information depending on the nature of the product, for example size/dimensions/weight
   5. The Product Page must be accessible from the Catalogue Page by clicking the photo of the product you elected to detail
5.  **Contact Page (contact.html)** with a form for customers to contact the company. The form must contain at least:
   1. 3 text fields
   2. radio buttons
   3. checkboxes
   4. a drop-down list
   5. a multi-line text field
   6. a submit button.
   7. The form must be styled and should have a modern look.
   8. The elements of the form must be correctly aligned.
6.  **Table Page (table.html)** with a table comparing the detailed characteristics of the products**.** For example, comparison of plans/packages, ingredients of a recipe, etc.
   1. The table must be styled with the aid of CSS (example: rows with alternating colors, highlighting the headers of rows and columns…)
7.  **Photo Gallery Page (gallery.html) with 12 photos.** Not the same as your product page ! The photo gallery must showcase your business as if you were a visitor taking a tour inside the company. For example pictures of your employees, pictures of the shipping area, pictures of the building, and yes you should also put a few pictures of your best selling products, the best ‘showroom’ models, not all products !
   1. You must place miniature thumbnails of the photos in the gallery, to be used as links.
   2. When clicking on the picture, thepicture is shown full size.
8.  You must insert a **Google Maps map** in your website. It could be on the contact page and/or the footer for example.
9. You must use at least **one custom font**.

**DESIGN Requirements**

1. You must draw 6 wireframe diagrams showing the design of each page. These can be roughly hand drawn on paper, no need to be good looking. You can write anything on the paper, the goal is brainstorming and showing your creative process. The final HTML/CSS coded pages can look different from the desing.

**CODING Requirements**

1. You cannot use technologies other than HTML5 and CSS3, **so no JavaScript, and no frameworks such as Bootstrap.**
2. You cannot use pre-made HTML templates, other than simple example found in W3schools and shown in class.
3.  **You must produce the CSS and HTML code yourself.**
4.  HTML code must be properly structured, indented, with the main parts well-commented. Your code must look professional, easy to read,
5.  the code must have a consistent style and follow the course coding style guide, for example lowercase tag names, good choice of class names, etc…
6.  All links are working
7.  All pages have the required metadatas in the <head> section, including an icon, a title and a description. The title and description should be tailored to each page.
8. All pages of your website must be valid HTML5 and CSS3**.** Use the HTML W3C Validator and CSS W3C Validator to validate your code before submitting.

**STYLING Requirements**

1. Formatting must be done only with the aid of CSS. No tables used for formatting sections of the screen (use div’s instead), no long strings of <br>
2. All CSS code must be in one .css file, no inline or in-file CSS code in the final version. You can test and develop with in-line styles, but merge everything in one css file at the end of the project.
3. CSS code must be properly structured, indented, with the main parts well-commented.
4. All of the pages must adopt the same visual style (one should not have the impression of being on another site when changing the page). Take care to ensure the quality of the visual aspect; your site should be attractive and should have a modern look.
5. You must use at least one custom font.
6. The site must display correctly with the following browsers:
   1. Google Chrome
   2. Microsoft Edge (latest Internet Explorer)

# Evaluation Sheet

|  |  |
| --- | --- |
| **Criteria** | **Points** |
| Adherence to instructions and content/structure requirements (tick all requirement boxes in this document) | 8 |
| All pages are valid HTML5 and CSS3, verify with W3C Validator | 6 |
| Good HTML page structure, use of tags like <nav>, <header>,<footer>, <div> with classes, etc… | 6 |
| Formatting is done only with CSS, no tables to split the screen, no long list of <br> tags. All CSS in one well-structured and well commented CSS file used for all pages. | 4 |
| Good use of CSS  Judicious use of selectors and properties, good choice of classes, no overuse of id’s, no un-necessary code duplication, insertion of a custom font, etc… | 5 |
| The site displays correctly in Google Chrome, Microsoft Edge and Apple Safari (to be confirmed if an Apple device is available for testing) | 3 |
| General appearance of the site (visual quality) | 5 |
| Coherence of the ensemble comprising the site | 3 |
| Source code (HTML/CSS) is sufficiently commented, indentation is applied, consistent code style, professional looking | 6 |
| General assessment  Originality, effort, improvements, research… | 5 |
| **Wireframe diagrams (1 for each page, 6 total)** | 4 |
| Presentation (individual grade) | 3 |
| Deployment on Sentora Web Server. Team was able to upload the site on the server. | 2 |
| **Total** | **60** |

# Submitting you project

**Setup and test your Sentora Web server domain and FTP client.**

* On the day before submitting the project, we will setup during class time your Sentora server domain, to host your web site on the school public server.
* See instructions in document “**Procedure for Sentora Server Hosting.docx**” found in Moodle
* Upload your current (unfinished) project with Filezilla to test your setup.
* Verify the web site displays correctly on the Web Server. If some pages or some files don’t work, but they were working fine on your local directory, **check the filenames and files paths for uppercase/lowercase differences.**

**Zip the whole project and put in the Dropbox drive**

* The practical project must be submitted on the last day of the course
* Submit your 6 wireframe diagrams on paper to the teacher. **Write the name of each team member on each page.**
* ZIP the whole project directory as an archive (zip file) named:  
  “**final-project-firstname1\_firstname2.zip**”.
* Copy/Paste in the Dropbox directory **FINAL-PROJECT-ALL.**
* **Upload your final version with Filezilla (FTP client)** on the Sentora Web server. See separate instructions in Moodle “**Procedure for Sentora Server Hosting.docx**”.

**Project Presentation**

* You must do a public presentation of your project on the last of the course.
* Both members of the team must talk. Each member will be graded individually for the presentation.
* The presentation should last between 5 and 10 minutes max for the whole team.
* See presentation guidelines below in this document

# Recommended Project Methodology

1. Choose a theme suited to the requirements
2. Decide the hierarchy of the site (site map) with main navigation path
3. Agree with your partner on:
   1. Directory names and folder structure
   2. All file names (so links can be created even before the file actually exists)
4. Design the model of the site with wireframe diagrams.
   1. Design the page template: header, nav, footer, main content div.
   2. After the template is done, design each page content div only, ignoring the template header, footer
   3. The design is a creative brainstorming process, you can do the diagrams with a pencil and graph paper. The diagrams DO NOT NEED TO BE GOOD LOOKING AND WELL DRAWN. You can write anything on the sheets to help your creative process. You should submit all and without cleaning up, including coffee stains!
   4. Optionally you can use tools like <https://moqups.com> , Photoshop or others
5. Write HTML content and basic CSS dimensions and positioning. Construct the “skeleton” of the site in HTML.
   1. Start with the template page with header, nav, footer. Use CSS styles for block dimensions and positioning, no need for colors or fonts yet.
   2. Decide the size of the remaining main content block.
   3. For each page, analyze the diagram model and distinguish the different blocks. Design only the main content of the page, ignoring the surrounding header and footer.
   4. Write the HTML code for each page, ensuring that each requirement listed above is fulfilled.
   5. Use CSS styles for block dimensions and positioning, no need for colors now.
   6. Validate the HTML code with W3C Validator.
   7. Indent and comment the code files
   8. Near the end of the project, make 6 copies of the template file and merge-in the content div from the 6 individual pages
   9. At the end of this stage, your web pages should look a lot like your wireframe diagrams, showing content and structure without much styling.
6. Detailed Formatting with CSS
   1. Define fonts sizes, colors, family. If appropriate, generate and insert custom fonts
   2. Define color scheme
   3. Background images
   4. Hyperlinks colors
   5. Define the styles of the titles (h1, h2, etc.)
7. Verify the validity of the source code (W3C HTML and CSS validators)
8. Ensure compatibility with browsers
   1. Google Chrome
   2. Microsoft Edge
   3. Firefox
   4. Safari

# Presentation Guidelines

1. You should present as if you were giving a presentation to your client (and not to your teacher), who has contracted you to realize the project.
2. Speak in the plural: “we” instead of “I”. You are part of a team. The client does not need to know who did what; they want an integrated solution.
3. In advance, agree amongst yourselves regarding who will say what and the order in which you will present.
4. Do not mention your strengths or your weaknesses (you will do this during the question period).
5. Do not obstruct the view of the projection (one person can sit down while the other speaks, or handle the computer for their colleague).
6. Make a demonstration of the particular functionalities of your project .
7. If the functionalities of the site do not work perfectly, show those that work and specify that the others are still in development or that certain functionalities still need to be refined.
8. Speak loudly and articulate well.
9. When presenting each page, do not read the content of the site. You should present without going into too much detail.
10. Do not show your code (leave that for the question-and-answer session if other students are interested).

**1.2 Example plan for the presentation**

1. Introduction
   1. Say hello and present the members of the team (first name / last name).
   2. Mention the name of the project / website.
2. Description of the project
   1. Type of project (online store website, corporate website, blog, reworking a site, etc.)
   2. Objective of the site (sales, increasing the number of members, information, etc.)
   3. Constraints of the project:
      1. Technologies used and why.
      2. Compatibility with multiple browsers (compatibility from which browser versions)?
3. Presentation of each page
   1. Start with the home page (talk about the choice of corporate color scheme, about the general design).
   2. Highlight the challenges from the point of view of user experience.
   3. Highlight certain constraints of the project (our client hoped for … and we resolved this by … in order that the users will be able to …)
   4. Present the site on multiple browsers in order to demonstrate its compatibility
4. Conclusion
   1. If all of the constraints from the scope statement were fulfilled, say so; if some are not fulfilled, say what percentage remains to be done.
   2. Thank the audience.
   3. Open a question-and-answer session.