P2: Project Design

(25% of the project grade)

Keep in mind that these documents might be updated during the semester.
Last updated: 12.08.2022.

Due Date

Monday, October 10th at 23:59. We do not accept late submissions (i.e., they will not be graded).

Overview

P2 counts 25% towards the project grade and is about designing your website graphically and plan the directory-structure. In addition, you should describe the implementation details around the design, how the final site will look like and how you will make it work. Reuse your stylesheet for the project documents in this course.

Description

The idea of design is to make a design that is clear enough for the users. This is easy to say, but it is hard in practice. This means that the design should be so clear that other people can implement it without questions. Another issue is to think user-oriented, like “Why should the navigation bar be on top and not at the bottom? Will the accessibility be easy for the user if the navigation bar is at the bottom?”.

Create the design

Your design should say what you will create, and it should be clear, complete and concise. The design includes the organization and navigation of the site. What pages will there be? How will they be connected? What sequence of steps will a user need to take for a common task?

For example, if users most often visit your site to find out what when the next game is, describe the path they will follow to get to that information. The design also covers the layout and appearance of each page.

Think a bit about your target audience and remember that some of them may have certain disabilities, thus, design with accessibility issues in mind (see reading below).

Also, remember that not every user will have precisely the same size screen you have, so think about page layouts that are flexible and will still look okay in a variety of sizes. Document the decisions you make.

You may change your mind about the design during coding. All the changes in the design when coding will be documented in P3_changes and will be delivered along with the code. See the P3 Project Coding on Blackboard. Design Resources can also be found on Blackboard, under Resources.

Mockups

Mockup-images are pictures of the design for each page or perhaps of common elements such as the logo or the navbar. The mockups can be created in a graphic/image editor tool (Photoshop, Pixlr, Paint.net, Illustrator, PowerPoint, Figma, etc.). Save or export them as web images (formats such as JPEG or PNG) and upload them to a folder on your folk.ntnu.no so you can link them correctly in the P2 document.
If you have not received necessary images and text from your clients, such as for a logo or the history of the organization, or you have an imaginary client, feel free to use dummy images and text. In this phase, focus on the layout of your pages: sizes, alignment, fonts, colors, borders, and so on.

*NOTE: As mentioned in the **P3 description** the website has to be responsive, but a mobile version is **not mandatory**. This is optional, so feel free to make mockups for a mobile version if you want.*

Planning your work

It is important to have a plan for your project to share the work amount between the teammates. In addition, the plan gives a general overview of the responsibilities, priorities, progress, and deadlines for the project.

The plan must cover all of the following:

1. A complete “to-do” list of all files you need to create, including HTML files, CSS stylesheets, part-HTML and JavaScript files used for modularity, GIFs for buttons, JPEGs from a digital camera, etc.

2. An organizational scheme for the files. However, this may not be the same as your organizational scheme for the website! For example, often all the navigational buttons will be put in a folder called "buttons," but that scheme obviously has nothing to do with how the content of the website is organized. Thus, a common technique for listing files and folders is the indented list, shown in the illustration.

List of Files and Folders

- homepage.html
- about.html
- images/
  - background_image.png
  - background_border.png
  - design_30.png
- scripts/
  - randomImages.js
  - rollOverMenu.js
- styling.css

3. A person that is assigned to be responsible for each file in the complete list. That way there will be no confusion about who is doing what, and everything will get done. An exception to this may be the external style sheet - defining CSS for the project may be a lot of work, and it is more equitable to share this task with the partners.

4. A deadline for each file to be completed. This is not the same as the coding deadline! It's true that everything must be completed by the coding deadline, but (as we described earlier) you want to be able to keep track of whether you are on schedule or not. Resolve to check on each other's progress. **Include time for testing (P4).**

An idea for organizing this information is to create a table with the columns: file name, file description (one-line description), deadline, and assigned person (items 1, 3, and 4). This is shown in the example.

Modularity

Rather than duplicating code for elements of your site that appear on multiple pages, you should handle repeated items by using shared files of JavaScript code.
Repeated elements include things like:

- banners, navigation bars, and footers,
- navbar button rollovers in navbar code,
- stylesheets (there should not be a separate style file for each page, but a single "main" CSS style sheet).
- JavaScript functions - when the same JavaScript function is used on multiple pages, it should be defined in an external JavaScript file (rather than being included and re-defined in each file through the use of SSI).

Depending on the project, achieving modularity may be a large task. Therefore, the implementation of **modularity should be shared by the team members**, since assigning it to one partner only may result in an uneven workload.

Deliverables

The final product of the design phase is a web page that contains a structured essay - a combination of text and images of your mockup pages. The HTML document should contain these components:

1. **Header:** A title, the authors' names, and date.

2. **Administrative Details:** The name of your client, and his or her position in the organization.

3. **Purpose, goals, and audience:** A summary paragraph to remind the reader about the purpose, goals, and audience of your website. Make it brief and succinct.

4. **Navigation Structure:** A brief discussion about the site's navigation structure using the terminology learned in class and the notes on slides “HTML Cont.” in the Site map and Storyboard sections (pages:47-50). This is not a sketch of your navbar (or whatever), but instead a map or description of which pages connect to which other pages. (It is sometimes better to describe a clique in words than to try to draw it.) The goal is to explain why the choice of a particular structure makes sense for your website.

5. **Page Layout and Appearance:** A description of design decisions that are universal to your site. You should specify in writing the fonts (types and sizes), the color scheme (for text, background, links, shadows, borders, etc.), navbar position in the page and its appearance, banner, footer, the social icons bar, or background images (if any of them is applicable). In addition to the textual description of these elements, you should include a mockup image of this generic page that has only these common elements and no other content. Finally, you should describe how these universal design decisions fulfill certain goals of the website and its audience.

6. **Content:** This is the longest section of your document. For every page, you will need to give a text description of the content of the page as well as an image of the mockup of the page. The text describes what goal(s) this page fulfills, and how design decisions specific to it contribute to fulfilling such goals. The reason why you should describe the content is that you might not have yet received the actual content from the client and in your mockup of the page, you will be using fake text, fake images, etc. While the content can be fake, you should try to use meaningful headings and captions to convey its meaning. Finally, you should describe the design decisions that are unique for this particular page (if any).

   Example:
   Images of e-board members are stored as thumbnails to reduce the loading time of the page. They appear in a grid format on the left side of the page. Each thumbnail is of size 150x150. When clicked, they will open as larger images (320x480) on the right side of the page, surrounded by text describing the member's duties and reasons for joining the org. The chosen thumbnail will have a surrounding border in gray color, to provide visual feedback to the user for the selected content.

7. **Minimum Requirements:** The website must include four distinct JavaScript applications (for example, rollover effects on images or menus, user-defined functions, form validation, animations, etc.). The four applications must all be different from each other. **Clearly, indicate how each of you will be fulfilling these minimum requirements.**

Please note that each of these items must be constructed entirely by you and not copied from external websites on the Web (though you can modify the code from our class materials). Once you fulfill the minimum requirements with your own code, you are allowed to get the code from the web (with source attribution) to incorporate certain advanced components (e.g. image galleries with transitions), but they will not count as part of the minimum requirements.

8. **Plan:** Provide a clear, specific description of your plan for building the site. Outline the tasks to be accomplished, the deadlines for those tasks, and who is responsible for them. Include a list of files and folders, which will help each partner link to files her partner has built. Include a complete list of everything you need to get from your client (text, pictures, sound, etc.) and when you expect to get them. To the greatest extent you can, try to have all of these things by the time your design is done so that nothing will delay your coding.

Example

Here is an example of a P2 design document:
http://www.idi.ntnu.no/~michailg/IT2805/exampleproject/p2_designexample.html

NOTE: These examples were drawn from earlier semesters when the course was somewhat different, but may be helpful. Notice how the design document is close to the final website, but there are changes that were described in the P3_changes document when the code was submitted. A description of P3_changes.html can be found in the P3 document.

Grading

Here are the aspects that you will be considered for this assignment:

- Your document, P2_design_GROUPNUMBER.html, should include all the components described above.
- The mockups should reflect the text description of the pages and be as close as possible to how the coded pages will eventually look like. We will look at how well the mockups convey the purpose and goals of the website as well as whether they are appropriate for the intended audience. For example, if one of the target audiences is children, how well do the colors, fonts, images, and the general layout speak to such an audience (it's cheerful, bright, not overcrowded, etc.).
- The document is well structured, easy to read, follows conventions, etc.
- **Your HTML code for the document page is valid.**

Delivery

Ensure to include all of the requested information in the file before delivering. Save the HTML file as P2_design_GROUPNUMBER.html and publish it on your website. It is recommended to save P1 to P4 files in the same folder called Documents to have a

structured web-hierarchy. Make sure to include a link to your website in the blackboard delivery.

Deliver the HTML file on Blackboard by **Monday, October 10th at 23:59**. We do not accept late deliveries and you cannot change the file after the deadline.