

Web Miniproject Assignment

Internet, intranet and Web
Technologies and protocols for application communications
Info et Réseaux en Apprentissage, Sup Galilée, Paris Nord

November 10th, 2014

This document describes all the mandatory requirements of the course project, from initial design to submission, as well as assessment criteria used for grading it. The keywords “*must*”, “*should*” and “*may*” indicate if a requirement is compulsory, encouraged or optional.

Project’s main goal is to create an interactive web application to demonstrate your mastery of web technologies you are learning during the class. You will work in groups of two or three people for designing and realising a small web application of your choice.

1 Design

1.1 Inception

First step is to choose an appropriate project. It *must* stimulate your interest and your creativity. The project can be a version of famous web site, scaled down in its functionality, or a brand new idea. Beware of excessive ambitiousness: a good job on a small project is better than an incomplete and large project.

Except your personal enjoyment and taste, the project *must* use the following web technologies: XHTML 1.0 Strict, CSS, JavaScript, PHP, and a SQL-based database. More details about technical requirements are in §3, while information about its assessment are in §6.

1.2 Design proposal

Then the project *must* be designed, preparing a design proposal.

The proposal *must* contain a written description of the overall aim of the web site and of its feasibility, it will help in starting to work in a organised way. The little document of maximum one page *may* be written in English or French, and *should* kept as simple and effective as possible. Some questions to make the analysis easier:

- What is the real-world problem you hope to solve with this web site?

- Who is your customer/target audience? Describe your target audience (age range, education)
- What value your web site offers? Why do your customers need your web site? What does your web site offer that is different from the competition?
- Describe typical interactive tasks a user may perform on the site How much content you need to prepare?
- Do you have existing written content for the site or will you need to create entirely new content? If so, how much content will you need to write?

The proposal *must* include also a conceptual global site map, showing the web pages planned for the site. You *should* draw low-fidelity page wireframes and hexadecimal color scheme palette. Finally, a little database diagram *should* be included to the proposal.

All the drawings *may* be realized with drawing software or with hand-sketch and scan (or smartphone photos, too). These artifacts (site map, wireframes, color palette, DB diagrams) *may* be added to the written document or simply listed as external files. The proposal *must* be packed up in a single archive and submitted via email no later than 12:00 p.m. of November 10th.

You *should not* continue working before receiving approval from instructor about your proposal, in order to avoid waste of time caused by possible changes requested to your design.

2 Architecture

The web application *must* be structured in the three-levels architecture, composed by storage, application, browser. Nevertheless the project *should* implement the application avoiding coupling between business logic and presentational aspects. Storage management is not a key element of the project, therefore data design is considered marginally.

Except this requirements, the concrete design is left to you. You can choose any approach in-between the following two extremes.

- client-centric: everything in the browser, except a little interface to the storage level at server-side;
- server-centric: everything in the server, except the web documents and some little scripting.

3 Technologies

Storage layer Data management *must* be implemented with a relational DBMS, such as MySQL, PostgreSQL, or SQLite.

Server-side scripting Application components to be executed in the server-side *must* be in PHP language version 5. Independently of the specific authors' architecture it *must* implement at least an interface to the storage layer using the PDO (PHP Data Objects) library.

Client-side scripting The application *should* include at least one dynamical interface, to be implemented in JS scripting language. It will offer as well a fall-back static alternative for being supported by any browser.

Client-side documents Final documents *must* be marked up with the strict variant of XHTML version 1.x and styled using CSS in version 2. The syntax of both the documents and the styles *must* be validated. The application interface *must* use at least two different pages and at least two different visualization styles.

Brand-new features of recent versions of these languages, namely XHTML version 5 and/or CSS 3 *may* be used in addition to the first described.

Server/client data The application *may* also use JSON format for the communication between the server and the client layers.

Reuse In general, third-party components reuse *must not* be used – no ready-made CSS stylesheets, no HTML templates, and no PHP frameworks. Nevertheless, if you already have previous experience with JS, you *may* use existing frameworks; in this case, obviously, the expected functionalities will be higher. Moreover, you *may* use pieces of third-party code. In other words, copy-paste is allowed, but you *must* really understand what your doing.

4 Presentation

Final Report A small report, of a length between one and two pages, *must* describe your work. It *must* be written, again, in English or French. The first part of the document will motivate the main choices regarding the implementation and the testing of the project. The second and last part you will point out the strengths and weaknesses of the resulting work, mentioning the main challenges faced.

Comments On the other hand, small-scale descriptions of the implementation details *must* be placed directly as comments into the source code.

Presentation You *must* package your project in an archive (ZIP, Tar+GZip or Tar+BZip) and send it with to an email message, using “[IWEB14]” as a prefix subject, to the instructor by 12:00 p.m. of Wednesday the 3th of December (last TP session). Late submissions will not be assessed.

After submission, you *must* present your project with a live demo of 5/10 minutes. Such a presentation will be given after the written exam, indicatively from 11:00 to 11:45 a.m. of Monday the 8th of December.

5 Communication policy

Q&A You *should* ask questions to your colleagues and to the instructor. As well as you *should* answer others' questions. If you do one of the two, you *must* do it publicly.

Collaboration You *must not* explicitly collaborate with people of different working pairs

6 Assessment

The project will be assessed considering the various requirements so far described: functionality, use of technologies, documentation, working practice.

Each of those aspects will receive a score of 20 points maximum, which will be weighted to determine its actual contribution to the project mark, according to the concluding table. The project mark, in turn, will constitute one half of the final assessment of the course.

Aspect	Reference	Final mark
Use of technologies	§2, 3	4
Functionality	§1	3
Presentation	§4	2
Communication	§5	1
Total		10