Project “PollWeb”

Version 1.0

Preamble

The course projects are inspired by real-world needs, and they usually refer to similar sites already on the web. Students must follow the specifications given by this document, but they can also refine them through an interaction with the teacher and the analysis of similar websites. The final website must be completely original, well organized and easily accessible to all the users.

Site Specifications

The *PollWeb* site is a simple system for conducting online surveys. Each survey has a *manager* user, a title, an opening and a closing text, and consists of a series of questions. Each question, in turn, can be of one of the following types:

1) *Short text*: a line of text;

2) *Long text*: multiple lines of text;

3) *Number*: numbers only;

4) *Date*: only dates, optionally with associated time;

5) *Single choice*: a series of options are presented and the user must choose exactly one of them;

6) *Multiple choice*: a series of options are presented and the user can choose one or more of them.

All the question types above have some common attributes: *code* (used to uniquely identify the question), *text* (the text of the question, optionally in HTML), *note* (an explanatory note that might appear after the question text to guide the compilation), *mandatory* (if set, indicates that the user must necessarily answer the question, or choose (at least) one of the answers in the case of multiple choice questions). Other attributes can be added to each question type to further specify it, for example

- *Minimum and maximum length* of a text field;
- *Minimum and maximum value* for numeric fields (and maybe also for dates);
- *Regular expression* that must match a text field;
- *Minimum and/or maximum number of selectable options* for multiple choices.

These attributes are all optional, but the more you add, the more realistic your project will be.

Tip: a very important part of this project will be, clearly, the definition and implementation of a data structure that can model the surveys by aggregating questions that, in turn, can be defined, each with its own characteristics, in the most efficient way, and associated with responses in an equally effective manner.

The following list contains a brief description of the contents and functionalities required by this site. Obviously, any further refinement or enrichment of these specifications will increase the value of the project.

- The system must manage three types of users: *participant*, *manager* and *administrator*. The types are cascaded so, for example, a manager can also be a participant. The data associated with each user must include at least a valid email address.
- The *administrator* is unique and we assume he/she is pre-registered in the system. Managers must be registered by administrators. Later we will see the registration procedure for participants.
- Each *manager* user can create new surveys, to which he will be associated. When creating a survey, all the textual fields (see above) must be provided, and then the questions are defined, as described in the following points.
- To compose the survey, the manager adds the individual questions one at a time, in a sequence. It must be possible to review the sequence of questions already entered in a summary screen, where the most important features of each question are also presented.
- The manager e must be able to modify, delete or rearrange (move up or down in the sequence) the questions.
- Each type of question should have its own insert/modify page, although the various pages should maintain a certain uniformity, at least for the attributes common to all types.
- The survey can be *reserved* or *public*. In the former case (*reserved* survey), the manager must manually enter the data (name, email, password) of the participants who can access the survey. *Optionally*, this data could also be imported from a CSV file. *Please note: each survey will have its own participants, so users included in a survey will not be able to access other surveys, unless they are registered (by repeating the data) also in the latter.* On the other hand, if the survey is *public* anyone can access it, i.e., there will be no authentication procedure.
- Once satisfied with the definition of the survey, the manager can *activate* it, making it fillable. Upon activation, the URL that the participants can use to access the survey will be presented. *Optionally*, in the case of reserved surveys, upon activation the system could send an invitation email, with access URL and credentials, to all registered participants.
- By accessing the survey URL (the one presented to the manager at the time of activation), participants of *reserved* surveys will be asked to authenticate themselves, and then proceed to the compilation. In *public* surveys, the participants will go directly to the compilation. *Please note: in reserved surveys, each user will be able to participate only once, so after a successful compilation, his credentials must be disabled.*
- The survey will take place by showing the user all the questions, in the sequence set. You should choose the input technique most suited to each type of question. *Optionally*, you can also present a paged survey, grouping questions.
- Once the answers to the survey have been submitted, the server will carry out all the necessary checks on the validity and on the constraints set on the individual questions, and if successful it will save the answers. In case of error, the user will be asked to repeat the compilation.
- The manager should be able to *close* the survey at any time, inhibiting access to other participants.
- The manager should be able to *export the answers* in CSV format and/or in other useful formats. You can choose the data format to make this file as clear and informative as possible. *Optionally*, the manager may also be able to view the individual answers directly on the web.

Technologies

- The basic structure of the site must be created using HTML5 (*with XML or Polyglot syntax*).The validation of all the site pages with respect to the chosen HTML flavour is an important part of the development and **must** be reported in the documentation.
- The site layout must be realised using CCS style sheets. The layout can be freely based on third party layouts available on the web or shown during the lectures. In this case, the degree of **customization** of the layout will be taken into account in the final project assessment. **A responsive layout is not required, but strongly suggested.**
- For the *client-side* programming, JavaScript is the **required** language. It is possible to include libraries developed by third parties, provided that they have suitable cross-browser portability and that they are described in the project documentation. However, **any abuse of these technologies is not recommended**, especially when they can be replaced with a suitable use of HTML, CSS, etc. **In general, your site should be functional also with JavaScript disabled**. The site without scripts may be less "friendly" or allow the access to "core" functionalities only, but sites whose dynamics is entirely script-based are not allowed. However, **scripts can play a more important a role in the functionalities whose users are restricted and pre-determined** (e.g., in the *back-end* functionalities for administrators, but not in the public *front-end* or in the login procedure).
- For the *server-side* programming, Java *(servlets, JSP)* is the **required** language. Any DBMS and template engine can be employed, if required. Again, it is possible to rely on external libraries.
- In general, the site must work and have a good *rendering* on Internet Explorer/Edge, Mozilla Firefox and Google Chrome, and *possibly* be compatible with older browsers (in this case it should at least *degrade well)*and with the latest versions of other browsers, like Opera. Browser compatibility **must** be explicitly stated in the documentation.

Project Development and Documentation

The specifications may not be exhaustive or completely defined. Every feature added or refined, also through an interaction with the stakeholder or the site end users, will be adequately assessed. All the design choices must be discussed and motivated.

The final project, developed following the guidelines given by the present specification, must be a fully functional website, whose contents and features will be assessed during the examination. The specification parts marked as *optional*, if not developed, will not make the project insufficient but, on the other hand, will not allow your project to reach the highest mark. If you choose to implement an optional feature, the result should not be necessarily perfect or complete: it must only show your commitment to deal with an advanced issue.

The documentation (**in electronic format**) which accompanies the project **must** contain at least the following information:

- Indication of software dependencies (which libraries are required on the client and server side?).
- Indication of the functionalities that have been developed or not developed. Detailed description of any extra or optional functionality added to the project.
- Diagram showing the site structure.
- Relational schema of the database (if any).
- Analytical description of the site layout, also indicating its static/dynamic components.
- Description of any advanced technology (language, framework, plugin, library, etc.) used in the project, specifying the reason of its adoption and the actual contribution given to the project development.

- Description of *any* cross-browser programming/rendering problem encountered, list of compatible browsers.
- Screenshots of the most important website pages (*optional*).

The actual contribution of each group member to the project **must** be declared in the documentation (indicating, for example, the members that mainly worked on server and client side programming, the layout designer, etc.). During the examination, each group member will describe its part of the realisation.

Project Evaluation

To evaluate the project, the following issues will be considered (in order of importance):

1. Compliance with the specifications.
2. Technical correctness.
3. Organizational clarity and correctness of the contents.
4. Accessibility and standards compliance.
5. Appropriate use of static and dynamic contents.
6. Design quality.
7. Adequacy of the documentation.

This evaluation will be combined with the result of the project discussion.

Additional Information

This specification is available in PDF format on the website of the Web Engineering course at the address http://people.disim.univaq.it/dellapenna. Additional information on the specifications can be obtained directly via email by writing to giuseppe.dellapenna@univaq.it.

Please note that projects should be carried out by *small* student groups (three members is the recommended number). Exceptions to this rule must be agreed with the teacher.

Summary

This summary should be compiled and sent to the teacher, in electronic format, *before* the examination. A copy of this form should be also attached to the project documentation.

Project name: ____

Authors:

First Name	Last Name	ID	Contribution / Role in project development

Client-side technologies/libraries/frameworks:
(HTML5, CSS 2/3, Javascript, JQuery, Bootstrap…)

____

____

Server-side technologies/libraries/frameworks:
(e.g., JSP, MySQL, JPA, template engines…)

____

____

Compatible browsers:

Browser	Version	Compatible	Degrading	Not compatible	Not tried
Internet Explorer Edge					
Mozilla Firefox					
Google Chrome					
Opera					
Safari					

Project discussion date: ____