Distributed Programming

Web Programming Test Assignment

Submission deadline 10th September 2015, h 23.59

The project consists in developing a simplified version of a web application to fill a survey. No information must be saved into the database, neither temporarily, until the completion of the survey takes place. If the browser used for filling the survey is closed by the user, all the user's answers are lost and the user must restart filling from the beginning. All the answers are mandatory. The user must have the possibility to move back and forward between the survey pages and for each question the web application must remember the answers provided by the user with the possibility for the user to change them. At the end of the survey, a certain amount of money will be granted to the user (as a reward) depending on how many users already filled the survey, according to the following specifications.

1. All the users who access the website can see, without any authentication or registration mechanism, the first page of the survey which requires first name, last name and age. After entering this information, a button must allow the user to load the next page of the survey from the server

2. The second page allows the user to specify a list containing 3 sports. The sports must be typed by the user (for example basket, tennis, soccer and so on). The user can choose to specify one, two or three sports but at least one sport must be specified. In this page, there must be two buttons allowing the user to load the previous page and the next page from the server.

3. The third page of the survey requires the user to enter a valid email address to be tied with the answers provided, in order to associate the final reward (in money) with such address. In this page, a button must allow the user to load the previous page while another button must allow the user to store all the answers into the database, showing a summary page containing the answers provided by the user (including the email address) and the information about the reward associated with the filled survey. The amount of this reward must be saved within the database, associating it with the specified email address. In this page, some statistics about the answers provided by the users must be showed, in particular displaying the number of people who already filled the survey, divided by their age (you must consider the following ranges 0-17, 18-29, 30-49, 50+) and the total number of preferences for each sport. In this computation, also the answers of the current user (the user who just filled the survey) must be taken into account. If the provided email address is already stored in the database, the application should reload the third page of the survey, without assigning any money reward, signalling the problem to the user and saving no data into the database. In this case, the user must be allowed to change the email address. It is required to exploit an efficient way to compute and show the number of preferences for each sport, avoiding the complete re-computation of the statistics starting from the data of each single user every time this computation is needed!

4. The money reward is computed as 50 euros for the first user who successfully stores the answers into the database, 25 euros for the second one, 12.50 euros for the third one and so on, halving the reward each time a new survey is recorded into the database. If the amount of the reward is less than 1 euro, then it is fixed to 1 euro for each subsequent survey.

5. In the submitted project there must be three users who already saved some information in the database, in particular the first one must be 43 years old with sports "skiing" and "swimming", the second one must be 30 years old with sports "skiing", "tennis" and "volley", and the third one must be 19 years old with sports "new" and "volley".

6. Partial information inserted into the survey must be kept for no more than 2 minutes between one page load and the next one. If more than 2 minutes elapse from one page load and the next one, the user must be redirected to the initial page with an adequate informational message and with no previous answers available. Using HTTPS is mandatory every time the email address of the user is transmitted over the network, but not for the first two pages.

7. The general appearance of the web pages must include: a header in the upper part of the page, a navigation bar on the left side with all the links needed to perform the various operations, and a central part which is used for the main operation.

8. Cookies and Javascript must be enabled, otherwise the website may not work properly (in that case, for what concerns cookies, the user must be alerted and the website navigation must be forbidden, for what concerns Javascript the user must be informed). Forms should be provided with small informational messages in order to explain the meaning of the different fields. These messages may be put within the fields themselves or may appear when the mouse pointer is over them.

9. The graphical layout, even if simple, must be consistent, that is, the pages must be as much as possible uniform among all the different browsers.

10. Extra requirement (only for students having the 8-credits course): design an XML format that can be used for storing all the information available in the database of the web application, and specify the format using a schema. Store the schema in a file named `schema.xsd`, and store the file in the main folder of your application (the same where you have stored your `index.html` or your `index.php` page), so that it can be freely downloaded. Then, add an administration page to your application, and make it reachable from the application main page. The administration page must include a form that allows the administrator to download an XML file containing all the information available in the database of the web application. The administrator must specify the administration password in a text field of the form in order to be authorized to download the file. The file must be downloaded using an HTTPS connection. The administration password must be hardcoded in the application and it must be set to "admin".

Submission instructions:
The instructions already published in the Material folder of the course web page for the installation on the cclix11.polito.it, still hold. Furthermore, you need to submit your project (the same that you installed on cclix11) in a zip file named sXXXXXX.zip (without blank spaces in the name) to the following web site:
https://pad.polito.it/enginframe/dp1/dp1.xml (from inside the Politecnico network) or
https://pad.polito.it:8080/enginframe/dp1/dp1.xml (from outside).
In addition:
1. The sql script included in the zip file (submitted to pad.polito.it) to create the database must have a name with the following pattern: sXXXXXX.sql (where XXXXXX is your own student id).
2. The main page of your web site must be put in a file named index.html or index.php in your SECRET_FILDER such that the website can be accessed at the url http://cclix11.polito.it/~sXXXXXX/SECRET_FOLDER without adding any other resource name at the end of the SECRET_FOLDER.
3. DO NOT use absolute links

WARNING: The system that accepts your projects, works in an automatic way and it will stop accepting submissions at the scheduled deadline. For this reason, we recommend you DO NOT submit your work in the very last minutes before the final deadline.

In case of any doubt and question related to the project, please firstly visit the forum in the course website in order to check if other students have already asked the same question. Otherwise use the forum (not the teacher email) to ask your question so that the response will be available to all students.