Distributed Programming I

Web Programming Test Assignment

Submission deadline: July 1, 2018, 11.59p.m. (on pad.polito.it and on cclix11.polito.it)

Build a simplified version of a website for managing online booking of a shared shuttle for people transportation. For simplicity, assume there is a single shuttle with a fixed capacity (but the capacity should be configurable in the code by means of a define or variable) and do not consider the time of transportation, which is considered fixed as well.

Assume also that the itinerary of the shuttle is determined by the alphabetic order of the addresses to be visited, i.e., the shuttle will first visit the first address in alphabetic order, then the second one, and so on until the last one.

1. On the home page of the site, anyone can view, without any registration, the full list of addresses to be visited by the shuttle, in alphabetic order (which is also the itinerary to be followed by the shuttle), and, for each segment of the itinerary, the number of booked passengers. If there are no passengers for a segment, a message has to be shown.

2. Each user can sign up freely on the site by providing a name, which must be a valid email address, and a password, which must contain at least a lowercase alphabetic character, and at least another character which must be either an uppercase alphabetic character or a digit. In case of invalid username or password, the user must be notified by the client, before sending the data to the server, and signing up must be forbidden by the client.

3. Each user can view, in his or her personal page, accessible only after authentication, the full itinerary of the shuttle (from the first address to the second one, then to the third one and so on until the last address). Moreover, the user can also see, for each segment of the itinerary, the number of passengers who will be on the shuttle in that segment, along with the usernames of the users who have booked them and for how many passengers each user has booked.

4. Each user can make a single booking for a number of people who will travel together (from a minimum of 1 to a maximum which is the capacity of the shuttle), from a certain departure address to a certain destination address. Each of these addresses can be chosen either by clicking on one of the addresses already present in the system or by entering the string of the address, if it is new. The booking operation and related addition of new addresses must occur within a **single** interaction with the server, i.e. the creation of any new addresses must occur in the server together with the booking. In the response, the server will indicate if the booking operation has been successful or not. In case of success, the new addresses have to be created and the application must show the full itinerary of the shuttle as already described for the personal page but highlighting in red colour the departure address and the destination address of the booking performed by the authenticated user. If the operation is not successful, the application must show a message that notifies the user about the failure, and the new addresses must not be created. An authenticated user may delete his or her own booking by means of a dedicated button. No partial modification or delete is possible.

5. Example:

Initially there are 4 addresses already added by the users: AA, BB, DD, EE.

Let's assume the shuttle has capacity 4 and the initial itinerary is:

AA → BB: total 2; user u1 (2 passengers)

BB → DD: total 3; user u1 (2 passengers), user u2 (1 passenger)

DD → EE: total 2: user u3 (1 passenger), user u2 (1 passenger)

User u4 requests the booking of a travel from AL to BZ (two new addresses entered by the user) for 2 passengers: the request is refused because in the segment BB → BZ the shuttle would have no enough capacity for all the requested passengers. Remind that addresses have to be visited in alphabetic order, and, starting from address BB, there are already 3 passengers on the shuttle, which has no capacity for 2 more passengers.

User u4 requests the booking of a travel from AL to DD for 1 passenger: the request is accepted. The new itinerary becomes:

AA → AL: total 2: user u1 (2 passengers)

AL → BB: total 3: user u1 (2 passengers), **user u4 (1passenger)**

BB → **DD**: total 4: user u1 (2 passengers), **user u4 (1passenger)**, user u2 (1 passenger)

DD → EE: total 2: user u3 (1 passenger), user u2 (1 passenger)

User u1 deletes his booking. The new itinerary becomes (note that the itinerary does no longer start from AA):

AL → BB: total 1: user u4 (1 passenger)

BB → DD: total 2: user u4 (1passenger), user u2 (1 passenger)

DD → EE: total 2: user u3 (1 passenger), user u2 (1 passenger)

User u1 requests the booking of a travel from FF to KK for 4 passengers: the request is accepted. The new itinerary becomes:

AL → BB: total 1: user u4 (1 passenger)

BB → DD: total 2: user u4 (1passenger), user u2 (1 passenger)

DD → EE: total 2: user u3 (1 passenger), user u2 (1 passenger)

EE → FF: total 0: empty

FF → **KK**: total 4: **user u1 (4 passengers)**

6. In the application deployed onto the Labinf server there must already be 4 users with usernames *u1@p.it, u2@p.it, u3@p.it, 4@p.it*, and passwords P1, P2, P3 and P4 respectively. The state must be the one at the end of the above example scenario.

7. Authentication through username and password remains valid if no more than two minutes have elapsed since the last page load. If a user tries to perform an operation that requires authentication after an idle time of more than 2 minutes, the operation has no effect and the user is forced to re-authenticate with username and password. The use of HTTPS must be enforced for sign up and authentication and in any part of the site that shows private information of an authenticated or signed up user.

8. The general layout of the web pages must contain: a header in the upper part, a navigation bar on the left side with links or buttons to carry out the possible operations and a central part which is used for the main operation.

9. 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.

10. The more uniform the views and the layouts are by varying the adopted browser, the better.

Submission instructions:
You need to submit your solution **both** on cclix11.polito.it **and** on pad.polito.it (as explained below).
The instructions already published in the Material folder of the course web page for the installation on 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 first 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.

The forum has to be used exclusively for requests of clarification about the text of the assignment and not for requesting help about how to solve it or how to solve specific problems encountered during the execution of the assignment.