

Distributed Programming 1

HTML, Javascript and PHP homework

To be submitted by 26 June 2013, 23.59

Develop a simplified version of a web site to make reservations for train journeys. For simplicity, consider only direct trains without intermediate stops. The site must have the following characteristics:

1. There are two categories of users: regular users (travelers) and a site administrator.
2. Any traveler can freely register to the site (providing first name, last name and a valid email address). The administrator has a pre-registered account with username "admin" and password "ADMIN".
3. Once registered to the site, a traveler can create his or her own trip. The traveler looks for a train by providing departure station, arrival station, and a departure time. If there are results that meet the requirements (i.e. that match departure and arrival stations and that depart at or after the indicated departure time), the traveler can select a train and add it to the list of trains for his or her trip. When a train is added to the list, the system must verify that it is compatible with any other added trains, that is, the traveler cannot be on two different trains simultaneously during the trip. If there is no compatibility, the operation has no effect and the traveler must be notified of the failure. The traveler can cancel at any time one or more trains that are included in the list of his or her trip. A page with the list of trains booked must show the total time the traveler will spend on train. In addition, there should be a button "Validate journey" that verifies, on request, that for all the trains that compose the trip the arrival station of the previous train is identical to the departing station of the following train (prior to this test it is suggested to order the trains by departure time so that checking will be simpler).
4. A list of trains must be pre-loaded into the database. The list must respect the following constraints: there must be at least four different stations, there must be at least two pairs of trains with overlapping travel period, there must be at least a combination of trains that connects all stations consecutively with compatible schedules for a trip.
5. The administrator, once authenticated to the system, has the ability to: a) display a list of all the trains in the trip of a traveler, b) display the list of all the available trains and related information (train number, departure station, arrival station, departure time, arrival time), c) cancel a train, d) add a new train. In the case where the change involves trains booked by a traveler, the journey of each traveler should be updated accordingly, and in this case the traveler must be notified of the fact that there was a change (not necessarily which one) through a special message at the next login .
6. The authentication with username and password must be made when required, and remains valid, if the user (traveler or administrator) has no idle periods longer than 5 min. If a user attempts to perform an operation after an inactivity period of more than 5 minutes, the operation has no effect and the user is forced to re-authenticate with username and password.
7. The general appearance of the web pages must contain: a header at the top, a navigation bar on the left side with links to perform the various operations, and a central part in which the main operation is performed.
8. Cookies and Javascript must be enabled, otherwise the site may not work properly (both for cookies and Javascript, alert the user and block the access to the site). Site's forms must provide short explanation messages about the meaning of the various fields.
9. Make sure that the site appearance, even simple, is as uniform as possible in most common browsers.

Additional requirements only for the students who have the 8-credits course:

1. While the traveler is writing the departure station name and at least the first two characters have been entered, the application must suggest possible completions of the station name using AJAX and XML.

2. While the traveler is writing the arrival station name, the application must suggest the possible completions (i.e. only the names of the stations that are directly reachable from the selected departure station). In order to make completion suggestion as prompt as possible even with high-delay connections, the application should try to anticipate the user, i.e. when the traveler adds a new train to the trip the application must download from the server the list of destinations that are directly reachable from the destination station of the added train and keep this list in memory. In fact, it is likely that the traveler will select this station as the departure station of a new entry and in that case the information about possible destinations will be already available.
3. The loading of a whole page must be avoided when possible, using the AJAX technique. In particular, the page downloaded when a traveler logs in should be updated without a new full page download when the traveler adds a new train to the trip or deletes one.

Instructions for submitting

Instructions for installation on the machine cclix11.polito.it have been published in the “Material” folder of the course. You must also submit everything installed on cclix11.polito.it, packed in a zip file, on the site pad.polito.it.

In addition:

1. The database included in the zip submitted to pad.polito.it must be in a file named sXXXXXX.sql (where XXXXXX is your student ID number “matricola”)
2. The home page must be a file named index.html or index.php
3. DO NOT use absolute links

Warning: the submission system is *automatic*. Submission will be closed automatically at the deadline. Submitting the solutions in the last minutes is strongly discouraged.

In case of doubts and questions about this assignment, first check the forum pages in the didattica.polito.it course website to see if somebody else already asked your question, otherwise use the forum (not email to teachers) to post your question so that the answer is available for everybody.