University of Liverpool - Department of Computer Science

COMP284 Scripting Languages (2017-18) -- Assignment 2 Re-sit: PHP

Your task for this assignment consists of two parts:

1. Develop a web-based system using HTML, PHP, and MySQL that provides the functionality stated in the Requirements section below.
2. Make the system that you have created accessible and usable via the URL
   http://cgi.csc.liv.ac.uk/~<your user name>/books.php
   taking care that the access rights for the file books.php are neither too restrictive nor too permissive. Making the system usable includes setting up the corresponding database on the departmental MySQL server and filling it with the appropriate data.

Requirements

The five members of a research group, Anne, Bob, Connie, Dean and Eve, want to make some of their books available for loan to other people in their group. The books, their owners, and the number of copies of a particular book owned by a particular researcher are as follows:

Book description	Researcher	Number of copies owned by that researcher
E. M. Clark, O. Grumberg and D. A. Peled: Model Checking. MIT Press, 1999.	Connie	1
M. Fisher: An Introduction to Practical Formal Methods Using Temporal Logic. Wiley, 2011.	Eve	1
J. A. Goguen and G. Malcolm: Software Engineering with OBJ. Kluwer, 2000.	Anne	1
M. Huth and M. Ryan: Logic in Computer Science: Modelling and Reasoning about Systems. Cambridge University Press, 2004.	Bob	2
M. Huth and M. Ryan: Logic in Computer Science: Modelling and Reasoning about Systems. Cambridge University Press, 2004.	Connie	1

M. Huth and M. Ryan: Logic in Computer Science: Modelling and Reasoning about Systems. Cambridge University Press, 2004.	Dean	1
J.-J. Ch. Meyer and W. van der Hoek: Epistemic Logic for AI and Computer Science. Cambridge University Press, 1995.	Connie	2
W. Stallings: Network Security Essentials: Applications and Standards. Prentice Hall, 2000.	Anne	1
W. Stallings: Network Security Essentials: Applications and Standards. Prentice Hall, 2000.	Bob	1
I. Witten and E. Frank: Data Mining: Practical Machine Learning Tools and Techniques. Morgan Kaufmann, 2005.	Dean	1
I. Witten and E. Frank: Data Mining: Practical Machine Learning Tools and Techniques. Morgan Kaufmann, 2005.	Eve	1

That is, in total there are 7 different books, some with more than one copy, some with copies owned by more than one researcher, that the five researchers are willing to loan to others. For example, Bob, Connie and Dean each have copies of `M. Huth and M. Ryan: Logic in Computer Science: Modelling and Reasoning about Systems. Cambridge University Press, 2004', Connie has one copy, Dean has one copy and Bob has two copies of that book.

To keep track to whom a book has been loaned, a web-based library system is to be used. The system should allow a user to

1. select the book description of a book via a drop-down/pop-up list or selection menu;
2. select the name of one of the researchers who owns a copy of the book selected via the first list/menu via a **separate** drop-down/pop-up list or selection menu;
3. enter the name of the person who borrows the book via a text field;
4. enter the e-mail address of the person who borrows the book via another text field;
5. after selecting/entering the data above, submit a borrowing request by pressing a `Submit' button.

Ideally, the user of the system is able to enter all this data via a single web page (not a sequence of two or more pages). However, a sequence of web page can be used if this is the only way that you are able to realise this system.
Also, ideally, the menus are populated with data from the database.
On submission of a borrowing request, the user should be shown a confirmation whether the borrowing request has been successful or unsuccessful. This confirmation should include the book description, the researcher, the name and the e-mail address of the user.

- The system should ensure that what the user enters as a name satisfies the following constraints: A name is an arbitrary non-empty sequence of characters that only consist of the characters a-z, A-Z, hyphen and apostrophe; contains no sequence of two or more hyphens or two or more apostrophes; and starts with a letter or an apostrophe. If these constraints are satisfied, then we call the name valid. If these constraints are not satisfied, then the system should display an error message and the booking request must be unsuccessful.

- The system should ensure that what the user enters as e-mail address satisfied the following

constraints: An e-mail address should only consist of the characters a-z, A-Z, dot, hyphen plus exactly one occurrence of @. If these constraints are satisfied, then we call the e-mail address valid. If these constraints are not satisfied, then the system should display an error message and the booking request must be unsuccessful.

- If name and e-mail address are valid, then a borrowing request must be successful if the selected researcher is still in the possession of a copy of the selected book, that is, not all copies of the book owned by that researcher have already been loaned by the researcher to someone. If a borrowing request is successful, then that copy of the book is now on loan and the book description, its owner, the name of the person who has borrowed it and the e-mail address of that person must be recorded in the database.

- A borrowing request must be unsuccessful if all the copies of the selected book owned by the selected researcher are already on loan. If a borrowing request is unsuccessful then no information is recorded in the database. Also, it should then be possible to select the same or a different book from the same or one of the other researchers. When doing so your system would ideally be programmed in such a way that there is no need to enter name and e-mail address again.

To keep track of the books that are still available and books that have been borrowed, underlying the library system must be a MySQL database storing for each copy of each book whether it is available or has been borrowed and for any borrowed copy of a book the name and e-mail address of the person who has borrowed it. The **PHP Data Objects** (PDO) extension of PHP must be used to implement the interactions between the library system and the database.

Additional requirements and comments:

- If all copies of all books have already been borrowed, then the system should inform the user about this and not show any drop-down menus or text fields.
- The description of the system above suggests that a user goes through a sequence of five steps in order to submit a borrowing request. However, if you use a single page design for your system, then there is little that prevents a user from skipping a step or skipping all steps before pressing the `Submit' button. Your system should make sure that a request is only processed once the `Submit' button has been pressed and should produce appropriate error messages if the request does not contain all the necessary data.
- The system would ideally help the user by only listing in the first menu book descriptions where there are copies left that can be borrowed and, once the user has selected a particular book description in that menu, would in the second menu only list the names of researchers who still having a copy of that book that can be borrowed.
- It will also be helpful to the user if entries in the two menus are listed in a `sensible' order, that is, book description should be ordered by the surnames of the authors and researchers should also be ordered by their names.
- Note that the user still has great freedom when it comes to entering a name. You need to make sure that name strings are correctly stored in the database.
  You should also make sure that code injection is not possible via any of the inputs of the system.
- A user can borrow more than one book and more than one copy of the same book, but a particular copy of a book can only be on loan to one particular user.
- There is the possibility that two users nearly simultaneously try to borrow the same book from a particular researcher and that there is only one copy of that book left. Depending on how you implement the interaction with the database, this could lead to a situation in which your system determines for both users that there is still a copy of the book available and then records for both users that he/she has borrowed that copy, with only the information on one of the users remaining in the database but both users receiving a confirmation that their borrowing request has been successful. This is obviously an incorrect behaviour by the library system and needs to be avoided. The borrowing request of exactly one of the two users must be successful and the borrowing request of the other has to be unsuccessful.
- As this is an assignment on PHP, the use of other scripting languages should be kept to a minimum. In particular, the use of JavaScript must be restricted to the pre-defined `submit()` function, as shown in Practical 5 (Exercise 3). Also, input validation should be done in PHP not in

HTML5.
- Your code should follow the COMP284 Coding Standard. This includes pointing out which parts of your code have been developed with the help of on-line sources or textbooks and references for these sources.

A script that deals satisfactorily with these additional requirements and comments, in addition to providing the basic functionality required, will receive higher marks.

Submission

Submit your HTML/PHP code and a dump of your MySQL database (as separate, individual files; not as part of an archive file) via the departmental submission system at https://sam.csc.liv.ac.uk/COMP/Submissions.pl?module=comp284 (COMP284-21: PHP). Do not forget to also set up the database on the departmental MySQL server, to make books.php accessible via the departmental web server, and to correctly connect the two. Also, make sure that the file or files that you submit are identical to those made accessible via the departmental web server.

Deadline

The deadline for this assignment is

Friday, 10 August 2018, 17:00

Earlier submission is possible, but any submission after the deadline attracts the standard lateness penalties. Please remember that a strict interpretation of `lateness' is applied by the Department, that is, a submission on Friday, 10 August 2018, 17:01 is considered to be a day late (analogously for submissions that are delayed further).

Assessment

This practical assignment will address the following learning outcomes of the module:

- rapidly develop simple applications, both computer and web-based, using an appropriate scripting language;
- document and comment applications written using a scripting language.

This assignment will contribute **30%** to the overall mark of COMP284. Failure on this assignment may be compensated by higher marks on other assignments for this module.

Marks will be awarded according to the following scheme:

- The PHP script is accessible via the required URL, works without producing script errors, all required files were submitted, the files accessible via the web are identical to those that were submitted, and the access rights of the files in your filestore must be such that no other user can view their contents in the filestore: 10
- Input/Output handling: 46%
- Database and database operations: 32%
- Formatting, commenting, and quality of code: 12%

Marks are given according to the extent to which the system behaves in the expected way and produces correct results, and, to a lesser extent, how well the code is written. Code that has no observable effect will receive no marks.

The mark for a submission that is not set up correctly on the departmental web server and MySQL server will be capped at 9.

As stated above, the University policy on late submissions applies to this assignment as does the University policy on academic integrity, which can be found at http://www.liv.ac.uk/student-administration/student-administration-centre/policies-procedures/academic-integrity/.

Department of Computer Science, University of Liverpool
Ashton Building, Ashton Street, Liverpool L69 3BX, United Kingdom
+44 (0)151 795 4275
Maintained by Ullrich Hustadt, u.hustadt@liverpool.ac.uk