Skip navigation

University of Liverpool - Department of Computer Science

- Computer Science
- University home
- > Computer Science
- > People
- > Ullrich Hustadt
- > COMP284
- > Assignment 1 Re-sit

COMP284 Scripting Languages (2017-18) -- Assignment 1 Re-sit: Perl and CGI

Your task for this practical assignment consists of two parts:

1. Develop a Perl script using `CGI.pm` for a web-based system 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/cgi-bin/cgiwrap/<your user name>/stat.pl`

   taking care that the access rights for the file `stat.pl` are neither too restrictive nor too permissive.

Requirements

The DBLP Computer Science Bibliography is an on-line collection of bibliographic information for over 4.2 million publications by 2.1 million authors. Your system is intended to produce statistics for small subsets of that data, more precisely, it is intended to determine how many publications authors have written on a particular topic.

1. The script should display a web page that contains a form with two text fields and a `Submit' button. The first text field should allow a user to enter keywords that describe an author or topic, called `query` in the following. The second text field should allow a user to enter a number that specifies the maximal number of publications that should be included in the statistics that the system will produce, called `maxHits` in the following
2. If a user presses the `Submit' button, the system should first check the input. The system should check that `query` is a non-empty string and should take measures against code injection via this input. The system also needs to check that `maxHits` is a natural number greater or equal to zero. If one or more of these checks fails, then the system should generate a HTML page containing an error message for each check that has failed and prompt the user to start again.
3. If the user's input passes these checks, then the system should retrieve information in XML format from the DBLP for up to `maxHits` publications matching `query`, this will be the *query result*. This can be done by retrieving the URL
   `http://www.dblp.org/search/api/?q=query&h=maxHits&c=4&f=0&format=xml;`

   Note that the query result is using UTF-8 encoding. You must make sure that your script and the output it produces correctly handles UTF-8 encoded Unicode characters.
4. For each publication the *query result* will include a list of authors. The system should count for each author in the *query result* how many publications in the *query result* he/she is an author of. The system should also determine how many publications are contained in the *query result*.
5. Once the system has completed the count, it should produce a HTML page that includes
   - a statement of the `query` and `maxHits` entered by the user, the XML data that was retrieved, in a form that makes XML tags visible and preserves space and line breaks, and a

statement of the number of publications that were retrieved.

- if and only if a non-zero number of publications was retrieved, two HTML tables, the first showing the ten authors with the most publications and the number of their publications (listed in order of the number of publications) and the second showing the ten authors with the least publications and the number of their publications (listed in reverse order of number of publications).
  Each table should have two columns, one for the names of the authors, one for the number of publications and one row for each author. The columns should have appropriate headings, the tables should have appropriate titles. You are permitted to use Perl's built-in **sort** function to produce those tables.

This HTML page should be displayed to the user as response to the URL the user has entered.

Additional requirements and comments

- As this is an assignment on Perl, no other scripting languages should be used. In particular, JavaScript should neither be used for input validation nor for error messages.
- 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.

Test data

Test data, together with the expected results, can be found at http://cgi.csc.liv.ac.uk/~ullrich/COMP284/tests-2017-18R/.

Submission

Submit your Perl script via the departmental submission system at https://sam.csc.liv.ac.uk/COMP/Submissions.pl (COMP284-1: Assignment 1R (Perl and CGI.pm)). Do not forget to also complete the second part of the assignment. 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. Note also that the University's definition of `lateness' for on-line submissions has changed and now counts Saturday and Sunday as `working days'.

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 Perl 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
- Creating web interface and handling user input: 25
- Retrieving data from the DBLP and computing the statistics: 33
- Creating HTML page with result: 20
- 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.

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/.