Inf553, Project description: Step 3
PubMed Web application

Théo Bouganim, Pierre Bourhis, Ioana Manolescu

1 Outline

In this step, you must **create a Web application that exposes the PubMed database content**, as follows.

2 To do

Building on, and extending the Web application development lab, you need to write a Web application that allows to do the following:

1. Shows a **start page** with the distinct journals in which papers were published. As there are many journals, the start page is in fact a *first page* with the 200 journal titles in alphabetic order, having a "Next page" link at the bottom, which leads to the next page of 200 journal titles; this page should have a "Previous" and a "Next" page, etc., until the last page of at most 200 titles, which only has a "Previous" link. Using the Next and/or Previous links, it must be possible to navigate over all the journal titles. Each journal title should be a link to the journal's page (see below).

2. When the user clicks on such a link, they should arrive on a **journal page** titled "Journal: *journal name*". In this page, we should see a list, with an item for each different author that has published an article in this journal. The authors should appear in the alphabetic order of their names. Each author name should be a link (see below).

3. When the user clicks on such a link, they should arrive on an **author page** titled "Author: *author name*". In this page, we should see a list of all the publications of this author, ordered first by the year (the most recent year first), and then by the journal name (in increasing alphabetic order). For each article, we should see:

    - The article title;
    - The year;
    - The link to the paper in Pubmed (pubmed_link field; this link should be clickable and lead to the respective Pubmed page);
    - The title of the journal (this should be a link to the journal page in your application (see 2. above);
    - The list of authors (each author name should be a link to the author page (see 3. above);
    - The grant information of the article (if any), e.g., the page may show: "Grant info: " followed by the text of the article grant information;
    - The conflict of interest information of the article (if any), e.g., the page may show: "Conflict of interest: " followed by the text of the conflict of interest information.

All pages should be dynamically generated (on demand) by your code, i.e., no static Web content should be present in the .zip file. Implementing a page cache (to avoid generating the same page more than once) is OK (you can assume that the database does not change while your application runs).

3 What to turn in

You need to turn in Moodle by **Nov 29, 2023**

A **.zip** of a **directory**[a] named by the student author(s)' name(s), e.g., Durand-Johnson.zip. The directory should contain all the application code, together with a README.txt file specifying what we need to do to run the application. The code should assume the database is called **pubmed**. The README should specify where we need to set the Postgres user and password information.

[a] Be careful not to zip together a set of files, but a directory containing everything.

We will try to run the applications and also look at the code. Readability and good organization will be appreciated.