NOVA SCHOOL OF SCIENCE & TECHNOLOGY
DEPARTAMENTO DE INFORMÁTICA

A Web Server for managing information about curricular units
(TPC2 – Redes de Computadores)

1. Introduction

In this assignment, one continues to study, through a concrete example, how the basic HTTP protocol is used for supporting client- server interactions where the server stores information in a database and dynamically generates HTML pages in response to GET operations from the client. In this exercise, the use of JavaScript in a HTML page is introduced, as an illustration of how the execution of the code in the client (browser) allows a more efficient rendering of the page that shows the response to the client.

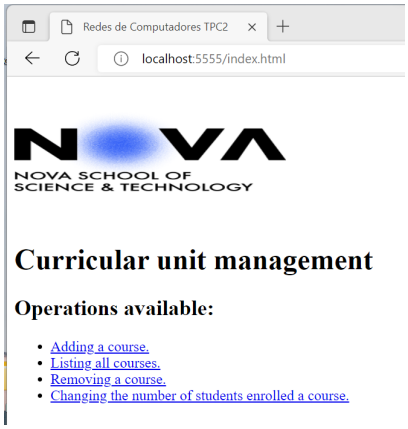
In this assignment, you must implement the JavaScript code and the HTML pages that support the management of information about curricular units (UC). The information about UCs is stored in a SQLite3 database called *curricularUnits.db* with a single table *Courses* with the following columns:

```
course_id integer PRIMARY KEY

name text NOT NULL

students_enrolled int
```

The management of the information is made exclusively through a browser. The web server is the same used in practical session 5. In general, information included in that exercise is useful for the current assignment and will not be repeated in this document.

In the following, the URLs to use are shown in the format *BASE/remaining_of_the_URL* . BASE can be for example http://localhost:port, where port is the number of one of the students of the group.

The application should have an entry page accessible through the URL *BASE/index.html* with the following aspect, where each item allows the invocation of one of the operations supported.



The suggested HTML code of the page is as follows:

```
<!DOCTYPE html>
<head>
    <title> Redes de Computadores TPC2 </title>
</head>
<body>
    <img src="nova_4.png" alt="Logo da FCT NOVA" width="300" height="150"/>
    <h1> Curricular unit management </h1>
    <section>
     <h2> Operations available: </h2>
     <ul>
        <li> <a href="http://localhost:5556/addCourse.html">Adding a course.</a></li>
        <li> <a href="http://localhost:5556/cgi-bin/listAllCourses.py">Listing all courses.</a></li>
        <li> <a href="http://localhost:5556/removeCourse.html">Removing a course.</a></li>
        <li> <a href="http://localhost:5556/changeStudentsNumber.html">Changing the number of students enrolled in a course.</a></li>
     </ul>
    </section>
</body>
</html>
```

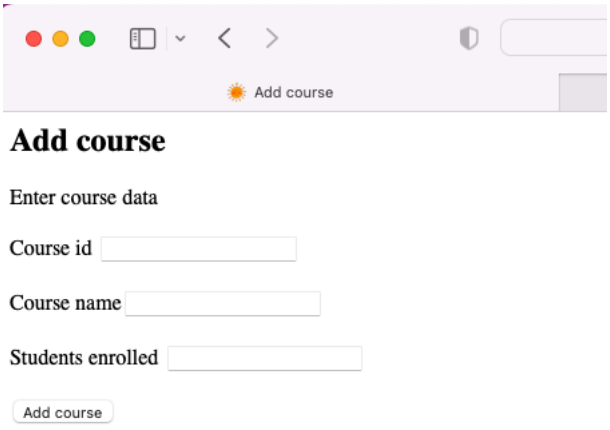
Most browsers allow the visualization of the HTML code rendered. For example, in Firefox / Windows one can click with mouse's right button over the rendered page and select *View Page Source*

2. Operations supported

In the following, a specification of the operations supported by the server is presented, screenshots about the desired user interface are also shown. These screenshots are only indications about what information should be asked to the user and which elements should be present in the answer sent by the server; you are free to modify the page if you preserve the inputs and outputs. Operations 2.1 to 2.4 don't require the use of JavaScript. In the implementation of operation 2.5 use of JavaScript is mandatory.

2.1 Adding courses to the database:

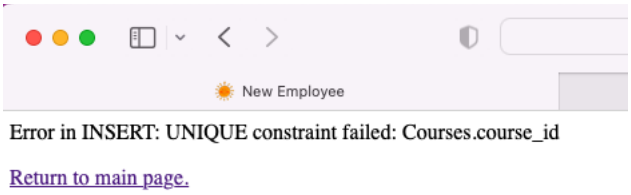
- The user should introduce the URL *BASE/addCourse.html* in the dialog box of the browser.
- The browser should present the following page



- The user should introduce the complete information about the course

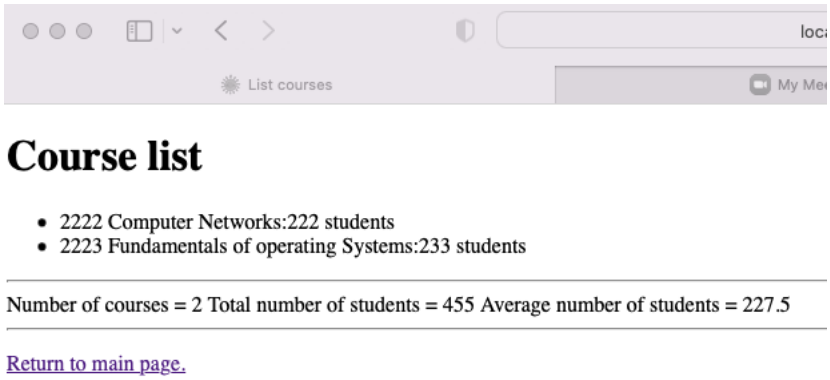
- The browser should show the following if the insertion was successful

- The browser should show the following if the insertion was unsuccessful because the curricular unit already exists



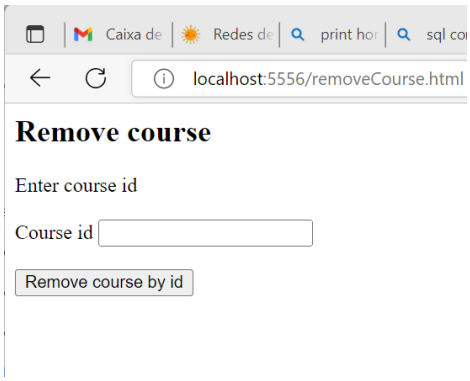
2.2 List information about all the courses

- The user should click the corresponding option (second) in the main page, directly invoking the Python script *listAllCourses.py* present in the *cgi-bin* folder.
- The browser should present the following page, including information about all the courses, finalized with a line with the number of courses, total number of students enrolled, and the average of students enrolled in one course. Don't forget to handle the case where there are no courses in the database.

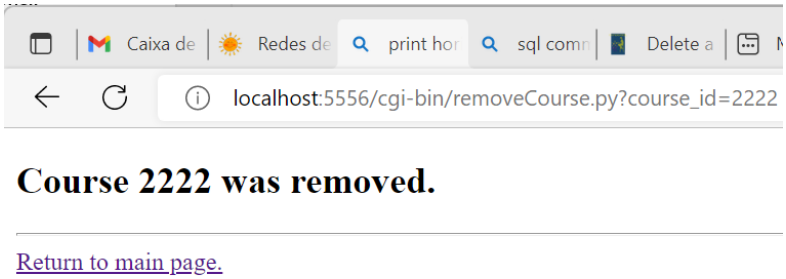


2.3 Deleting a course from the database:

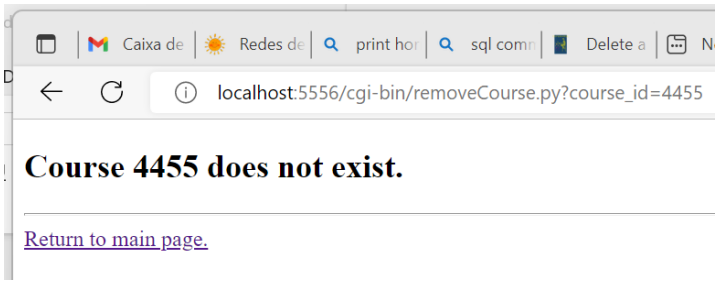
- The user should introduce the *URL BASE/removeCourse.html* in the dialog box of the browser.
- The browser should present the following page



- The user should introduce the number of the course to remove
- The browser should show the following if the removal was successful

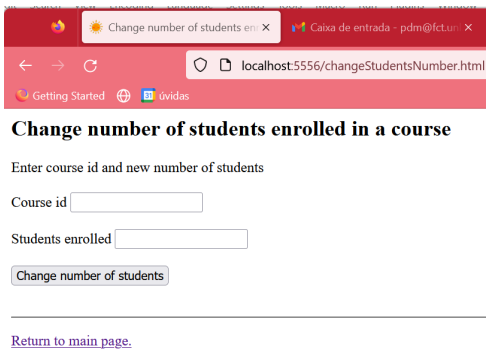


- The browser should show the following if the removal was unsuccessful because the curricular unit does not exist.

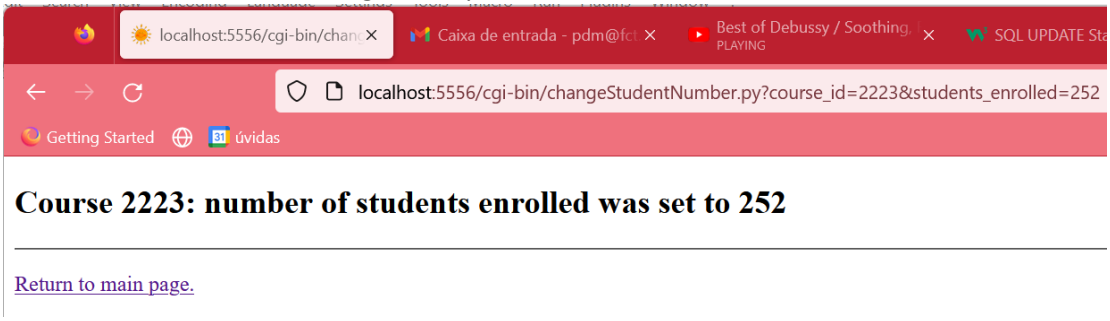


2.4 Change the number of students enrolled in a course:

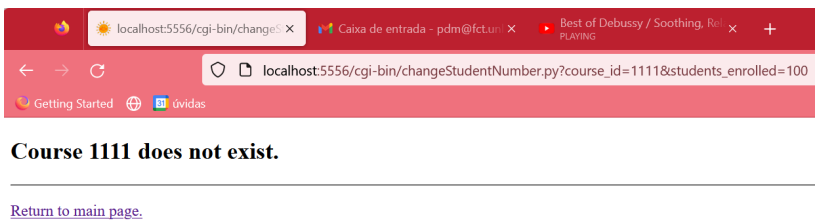
- The user should introduce the URL *BASE/changeNoOfStudentsEnrolled.html* in the dialog box of the browser.
- The browser should present the following page



- The user should introduce the course number and the new number of students
- The browser should show the following if the update was successful

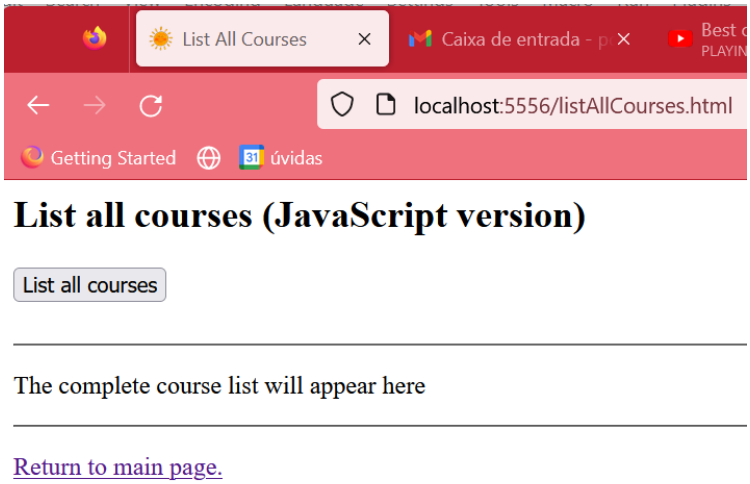


- The browser should show the following if the update was unsuccessful because the curricular unit did not exist

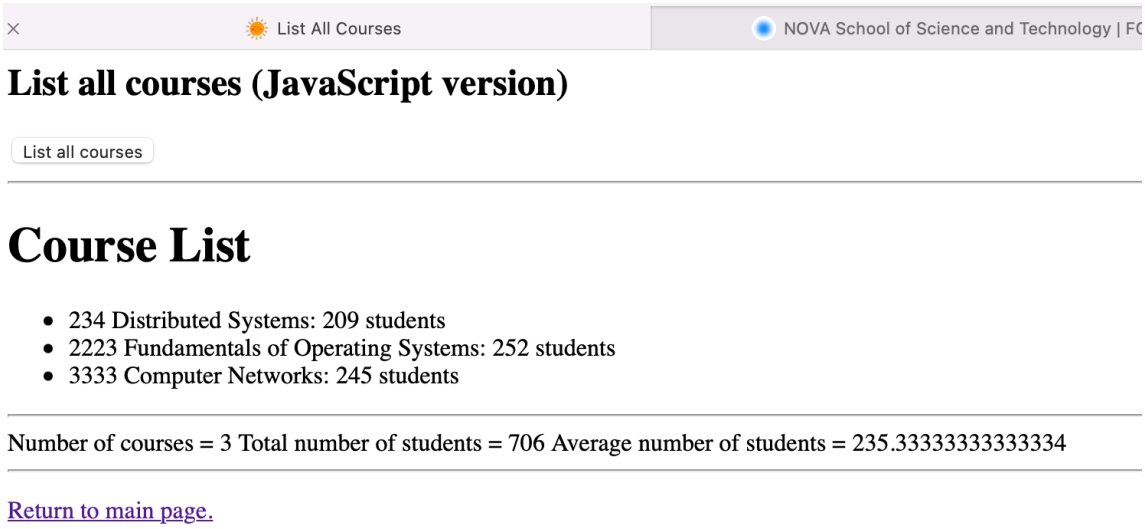


2.5 List information about all the courses (JavaScript version)

- The user should introduce the *URL BASE/listAllCourses.html* in the dialog box of the browser. This page should include JavaScript code that is executed when the button "List all courses" is clicked. This code starts a GET operation and waits for the arrival of data from the web server; the HTTP text received will replace the text "The complete course list will appear here".
- The browser should show the following page:



- After clicking, the browser should show information about all the courses, finalized with a line with the number of courses, total number of students enrolled, and the average of students enrolled in one course. The Python script code is very similar to the one produced in 2.2.



To write the JavaScript code that is included is page *URL BASE/listAllCourses.html* please see the slides of the lecture 7 of October 11,2022. You can also look at the tutorials in *https://www.w3schools.com/js/js_ajax_intro.asp*.

3. Delivery

The delivery will use a Google form, with details sent later. The deadline for delivery is Saturday 29, 2022 at 11.59 pm.