

EECS4413Z-Test 2

How to proceed

1. Import the project from Lab 7 with the name `YourStudentLastName_YourStudentNumber_Test2`. Example: `Lee_2109800_Test2`. You will use this project for Test 2.
2. In the remainder of this document we use the short name "Test2" to denote `YourStudentLastName_YourStudentNumber_Test2` application
3. Implement new specifications, as indicated in the Questions section. Make sure the functionality of Labs 6 and 7 remains unchanged, that is
 - if your application is accessed with <http://localhost:port/Test2/Start>, it should exhibit the Labs 6 and 7 functionalities (in other words, do not change any functionality of Labs 6 and 7)
4. Do not add new Java classes or servlets. Instead, modify existing files.
5. Submit your work by
 - exporting the project to a war file and also making sure the *source files* are included in the export.
 - uploading `YourStudentLastName_YourStudentNumber_Test2.war` using Moodle Test2 page (again, make sure you exported the source).
6. In the remainder of the text, "+" means a requirement and requirements have approximately the same weight.

Notes:

1. A reminder of your database schema:

```
CREATE TABLE ENROLLMENT (cid varchar(20) not null, sid char (20) not null, credit int not null);
```

```
CREATE TABLE STUDENTS (SID varchar(20) PRIMARY KEY NOT NULL, GIVENNAME varchar(20), SURNAME varchar(20), CREDIT_TAKEN int, CREDIT_GRADUATE int);
```

The Questions

Make sure your Start servlet can be accessed with `"/Start"` and `"Start/*"` where `*` means any string. Test this before going forward.

I. A new requirement is to report **all students from the STUDENT table**, when invoked with <http://localhost:port/Test2/Start/AllStudents>.

+S1.1 Process the URL and its path; query STUDENT table to get the students names;

+S1.2 Return Form.jspx with the list of student ids, names, etc. (you can reuse the same table format as in Lab 6)

+S1.3 Do not show the form and its input field names when you display the list of students.

II. Consider that a new requirement for your application is to allow an administrator to add rows to the ENROLLMENT table. When an administrator invokes the application with an URL like this, **http://localhost:port/Test2/Start/AddEnrollment**, then

+S1.4 A new form "AddEnrollment.jspx" shows 3 input fields, with the labels "Course id" , "Student id" and "Credits" that correspond to *cid*, *sid* and *credit* columns in the Enrollment table; and a button "Enroll";

+S1.5 When the user clicks on "Enroll," the application inserts a row with the form's input in the table ENROLLMENT;

+S1.6 Report success to the user, in a new page, Results.jspx and display the content of the Enrollment table. Highlight the newly added row with red color;

+S1.7 When enroll fails, display a meaningful message, in Results.jspx; For example, the enrollment fails if *sid* does not exist in the STUDENTS table; The course id, *cid*, can be new;

+S1.8 Validate the "Credit id" field: it should be a positive integer. Validate the text fields: they cannot be empty. Server-side validation is mandatory.