# Project: Transdisciplinary project S2 – Bsc Computer Science

You now have some good skills, taken independently, in the fundamentals of computer science:

* First full stack application,
* Integrate PHP, HTML, CSS and JS in a single app.
* Use web interface to interact with database.

The goal of this project is to make you use all those different skills to realize a website generator application.

# Summary

The goal of this project is to continue the work done in the previous project (static web generator).

Now the objective is to transform the static website into a real application to add a PHP backend.

# Specifications

The global flow of the application is shown in the attached diagram. Please refer to it if something is unclear in the following specifications.

## Global logic

The EPITA International Programs Staff wants to have a simple way to extract data from the School Management System application.

As it is not directly extendable (no way to modify the School Management App), some development is required to extract data from the database and to present them appropriately.

The website should have the following pages:

* Login page,
* Welcome page,
* Populations,
* Courses,
* Grades (bonus).

The course and the login pages were not present in the previous semester project.

## Login Page

The application proposes a login page where 2 input fields are present, login and password. You will have to add the necessary information in the database to authenticate

## Welcome page

1. Presentation of the different population lists with count of student per track
2. Presentation of overall attendance (percentage)

The items presented in point a. are clickable and allow to access to the specific population page. It means that your program has to generate a population page per track with the demonstrated content

### Population page

1. Presentation of the students composing this population. For each student, there is an information of number of passing or failed courses.
2. Presentation of courses assigned to this population.

Actions:

There is a column “action” in the student table which allows you to edit students details or remove a student from the population.

* A1. By clicking on the pen, you will be able to modify the students details (only names), in a page called “Edit”
* A2. By clicking on the “-“ button, you will be able to delete a student from the population. All the attached grades should be removed as well. A confirmation modal window is shown before actually triggering the action.

There are 2 buttons above the Students table, those buttons allow you to respectively do:

* A3. (Bonus) **Add** a student to the population by providing the details for this student using a modal or a standalone page (up to you to decide), the result is a refreshed list of students in the table list. . As soon as you add a student his/her grades entries are automatically created for all the courses assigned to the population. Grades for the new student are set to NULL in that case.
* A4. **Search** a student in the existing list of students. You have to define by yourself what could be the ergonomics of this feature.

There are 2 buttons below the Courses table, those buttons allow you to:

* A5. (Bonus) **Add** and assign a new course to a population. A possible design is shown in the attached diagram showing the flow for the application. Adapt the ergonomics to your own skills and to what you think is the best. As soon as a course is created, all the grades entries are added (with a grade set to NULL) for all the students in the population.
* A6. **Search** and assign an existing course to the population.

### Course grade page (Bonus)

List of the students with their associated grade per course, it is possible to modify the course grade for a student by following the same flow as in the student edition feature.

In this case the “-“ button does not remove the line, but set the grade to NULL.

# Development Guidelines

## Setup

Identify what will be your setup. Using MySQL here is recommended. You can easily migrate from the last dump (H2) to MySQL as the syntax is very close.

## DR1 - Data retrieval - methodology

Prepare the queries from the database using select queries. Identify when you have counters and lists to fetch from the database.

For instance in the first page, we need to have a “group by count” to list all the populations student count.

## DR2 – Data retrieval – source code organization

* Make sure all your queries are stored under a folder,
* one query per file.
* Pay attention to the file names so that one can easily understand what is this query doing.

## AF1 – designing the pages and flow through your PHP application

Prepare all the application pages and make sure the flow matches what you need to achieve. This application can be developed using classical MVC or MV\* patterns (with ajax calls), choose the one that you think is the most effective for you.

## AF2 – Implementing the content of the page, page by page

Focus on one page and complete it first, because you will discover lots of things to understand better the techniques to be applied in the next pages.

# Deliverables – PAY ATTENTION

The expected deliverables is a zip containing 3 folders:

* **doc**: The technical documentation of your project : what are the different big components you use and how they connect together.
* **src** : The complete source code of your project (including php source code, sql queries, css and html if any).
* **video**: a 5 minutes video maximum with 2 parts:
  + you demonstrate the running application for maximum 2 minutes. Follow the order in the attached application screens diagram.
  + you demonstrate your environment (source code with ide) for maximum 3 minutes.

The deliverables should be zipped and submitted through Teams. The structure of your zip file should be like in the following screenshot:

If you do not enforce this layout for the submission, you will be penalized.