T5 - Web development

T-WEB-501

Job Board

Web Project

3.2

Job Board

language: HTML, CSS, javascript and other web technologies

- The totality of your source files, except all useless files (binary, temp files, obj files,...), must be included in your delivery.

This project aims at realising:

- a database to store job advertisements,
- a webpage (*front end*) using Javascript technologies to display an online job advertisements board as well as an adminitration page for the admin user,
- an API (*back end*)
    - to allow user to apply to jobs
    - to manage the DB (only for admin user) - CRUD operations (Create, Read, Update, Delete)

There are several steps, that must be completed one after the other.

Mind out! Each step depends on the previous one…

For each step, create a directory in your repository named **Step XX** that contains all the necessary files.

When you complete a step, duplicate the files in a new folder as a base for the next step.

All your files must be tested on your local host.

STEP 01 — 2PTS

Create an SQL database to store job advertisements.
It must contain at least:

- a table to store advertisements,
- a table to store companies,
- a table to store people (whether in charge of an advertisement or applying to an advertisement),
- a table to keep information about a job application (referencing the mails sent, the people concerned, the ad concerned,…).

You can add as many relevant tables as you wish to, and add the fields you want in these tables.

STEP 02 — 1PT

Write a HTML/CSS page showing several job advertisements.
For each ad, there must be at least a place for a title, one for a short description, and a “learn more” button.
Clicking on this “learn more” button does not have any action for the moment.

The design of this page is up to you, but do not spend too much time on it now.

STEP 03 — 2PTS

The “learn more” button must now display all the available information about the ad (full description, wages, place, working time,…), without reloading the page.
No popup.

Keep the database fields coherent with the information you display on the HTML page.

Once again, the design is up to you: put this data wherever you like.

STEP 04 — 3PT

Create an API providing CRUD operations on the database tables (Create, Read, Update, Delete).
The “learn more” button must be linked to an API route to dynamically fetch the job infos.

Attention must be paid on using the appropriate HTTP verbs and the restful API routing rules

You can use the language you feel comfortable with. Here’s some candidates: Python (with Flask or FastAPI), NodeJS (with Express.js), PHP (with Laravel or Lumen), …

Fill in the database using your backend.

STEP 05 — 2PTS

On your webpage, add an “Apply” button for each ad.
When this button is clicked, it opens a form to

- enter information about you (name, email, phone,…)
- send a message to the owner of the ad (you).

This action must be saved in the database.

STEP 06 — 3PTS

Add an authentication mecanism on the webpage.
When identified, you don’t have to fill in your personal information to apply to a job.

You need a **connection** page and another one to create/modify an account.

STEP 07 — 5PTS

Create a HTML/CSS page for monitoring the database. From this page the user can list all the records of your tables, can create new records and can update or delete the existing ones.
This page can be accessed only by an admin. So a successful connection rediret to this page for an admin user or to the page created at step 02 otherwise.

Think about pagination when displaying big list of database records

The conception and page layout is up to you. Make them simple and user friendly

STEP 08 2PTS

Now you can polish your pages up by improving their design, tweaking and refining the style sheets.