

---

# PHP / SQL Intermediate practical evaluation

## Time allowed : 4h00

---

### Marking

At the end of the allotted time, you must deliver all of your files, and a score out of 20 will be given based on to the following scale:

- **Exercice 1 : 2 points**
- **Exercice 2 : 3 points**
- **Exercice 3 : 5 points**
- **Exercice 4 : 5 points**
- **Exercice 5 : 3 points**
- **Exercice 6 : 2 points**

## Exercise 1 (2 points) : Setting up the project

---

Create a new project on git. To submit your work at the end of the evaluation, you will have to upload your sources to a GitHub repository.

## Exercise 2 (3 points) : Creating a database

---

Create a database and name it "real estate".

Inside it, you must create a table that you will call "housing" with the following fields:

Table : housing

Fields :

- id\_housing
- title
- address
- city
- pc \*
- area \*\*
- price
- photo
- type (*letting or sale*)
- description

\* stands for postcode

\*\* refers to the surface area of the housing

### Exercise 3 (5 points) : Saving data

---

Create a form to add items of housing to the table named "housing".

Several checks are to be carried out on the input data:

Required fields are: title, address, city, pc, area, price, type

The postal code format must be verified and correct.

The price and area fields must contain integers only.

The photo field must allow an image file upload, with several checks: extension and type of file, size of the file, etc.

The *type* field must be managed via a *radio* input type or a *select* option.

Save the data in the corresponding table of the database.

### Exercise 4 (5 points) : Displaying data

---

Create a page which for displaying of the data input in exercise 3. We should be able to find all the housing items with their respective information.

This display will preferably be in the form of a table, and it will be necessary to cut the text by adding "..." if the description or other textual information is too long.

Regarding the photo, the path should not be displayed, but rather the photo itself. However, this should not affect the readability for the user.

### Exercise 5 (3 points) : Complementary developments

---

Files uploaded via the photo field must be renamed.

That is, a file named "my-apartment.jpg" will automatically be renamed "housing\_38.jpg" (the number 38 depends on the housing id).

The file "housing\_38.jpg" should be kept in its original size. You have to save this file in a folder named 'uploads/'.

## Exercice 6 (2 points) : Using composer

---

In the exercise 3, you have to check if the postal code format is correct.

Use composer and download a library which helps you to validate the format of the postal code.

Use packagist to search for a good library.

For example, this one : <https://packagist.org/packages/sirprize/postal-code-validator>