

PHP / SQL Practical Evaluation

Time allowed: 4h00

Scoring

At the end of the time allowed, you must deliver all your files and you will get a note on 20 points followin the scale:

- **3 points** for exercise 1
- **5 points** for exercise 2
- **4 points** for exercise 3
- **6 points** for exercise 4
- **2 points** on the following criteria:
 - Code indent
 - Commentaries
 - Relevant variable names
 - Code Lisibility

Bonus: bonus points can be awarded if facultative functionalities are done. Don't do them if you don't have enough time!

Advices

Don't worry if the exercises seem hard! Follow each step, one after one.

- First, read everything!
- Take your time to code, comment your code! (You can use the instruction as a comment)
- Keep your functions simple. They must do only one thing, it's easier to maintain.
- Concentrate on the PHP, not on the CSS!
- The more your code is readable, the easier it will be to succeed!

Each exercise is independant, separate each one in a different file or even a different folder.

Exercise 1: « Let's join! »

We have 2 SQL table:

Table "users"	Table "articles"
<ul style="list-style-type: none"> • id • firstname • lastname • email • role 	<ul style="list-style-type: none"> • id • title • content • picture • date_publish • id_user

In a separate file, write the SQL request to display an article (id = 10) and his author using a junction.

Note: Only write the SQL request, not the PHP.

Exercise 2: « The trainee »

You are responsible for a trainee. He must get an user list from a database. He also must add the ability to add new users with a form.

The main script is composed of 2 parts:

- The user list and their informations
- A form to add a new user

The trainee begins with PHP development. He doesn't indent code and do not comment code either. As his guide, you must help him and correct his code to make it fully functional. The code must also be indented.

Steps:

1. Download the files at this address:
http://wf3.axessweb.io/evaluations/php-sql/evaluation_docs.zip
2. Deploy *index.php* and *connect.php* files on your local server. Don't forget to import the SQL script *users-install.sql*. The script will create a new database with a table named "users".
3. Help your trainee to debug his code, correct his mistakes and make it functional.

Exercise 3: « Let's clean! »

In this exercise, you must create an Ajax form. You can use Ajax with JavaScript or jQuery.

The form will allow to add a new vehicle in the database with the following fields:

- Brand
- Model
- Year
- Color

The form treatment will be made in ajax. A message will display if the data entry has failed or was a success.

Exercise 4: « A classy cat... »

You work for a local animal shelter. You must redesign their website.

You have to create a class named Cat() with the following properties:

- Name (string - 3 to 20 characters)
- Age (int)
- Color (string - 3 to 10 characters)
- Sex (string - male or female)
- Breed (string - 3 to 20 characters)

Make the getters/setters to validate the above data type. You also have to create the class constructor in order to instantiate the class.

Add a getInfos() method to return all the properties as an array..

In a new file, instantiate the class to display 3 different cats and display the result with the getInfos() method previously created.