



Proficiency Test

Thank you for taking the time to attempt this test. As a business we encourage innovative thinking and initiative. In these tests before you we are looking for those elements but we are strongly urging that you consider what has been asked for. Anything above what is asked is a bonus but due to time constraints may hinder your progress. The test should be conducted in any language you are familiar and proficient with. On completion submit to Github repository. Make sure you have given us the necessary instructions to deploy your solution, if you use a specific database ensure that we will be able to get it running.

Test 1

Create an HTML form with the following input fields to allow for the capturing of data into a database:

Name, Surname, Id No, Date of Birth, POST button, CANCEL button

Create a database with a relevant schema to store the input fields in.

REQUIREMENTS:

- Save 3 records into the database without duplicating the Id No. The ability to capture a duplicate Id No in the database table is an immediate fail.
- If a duplicate Id No is found up on capturing, the user must be informed about this and the form repopulated. People do not like to input their information in twice.
- Validate the Id No field to make sure it is a number and that it is only 13 characters long.
- Validate the Date of birth field to make sure that the input date is in the format dd/mm/YYYY.
- There must be a valid data in the name and surname fields and no characters that can cause a record not to be inputted into the database.



PASS:

- Validation works as per requirements
- Date of birth field is captured correctly and is stored properly in the database
- The Id No field is no more than 13 chars long. (Bonus – check match with Date of Birth)
- No duplicate ids are in the database and the user is made aware of this.

Test 2

This task is to test your skills in manipulating arrays and file handling.

In this test you will be making a CSV file of variable length, a form will ask for the amount of data to generate. Check the requirements on how to generate the file.

The file will have the following header fields

Id, Name, Surname, Initials, Age, DateOfBirth

The data will look like this

"1","Andre","van Zuydam", "A", "33","13/02/1979"

"2","Tyron James", "Hall", "TJ", "32", "03/06/1980";

After this you will import the file into a database and output a count of all the records imported.

REQUIREMENTS:

- 1) Create two arrays, one for names and one for surnames. There should be 20 Names and 20 Surnames in each array. Use these arrays to generate random names, ages & birthdates to populate a CSV file. The initials are the first character of the name always. Write a function to perform the task of creating the CSV file, you should pass it the number of variations you need.
- 2) The CSV file should be outputted to an output folder, the name of the file must be output.csv.
- 3) An input field will take the amount of records to be generated.
- 4) There should be NO DUPLICATE ROWS IN THE CSV. I.e. The name, surname, age, date of birth must be unique.
- 5) Output a CSV file of 1 000 000 records.



- 6) Import the file using a form variable of file type. One should browse for the file and upload it to the website.
- 7) Create a table called "csv_import" with the relevant fields and types to hold the data of the CSV file. Use code to create the table.

PASS:

- The file is outputted to the correct folder with the correct name, with the correct number of records.
- IE: A 100 record file will have 101 lines to include the headers.
- The code will have the correct arrays of Names and Surnames.
- The file is generated according to the specification of no duplicate rows.
- All the instructions are completed 1 - 7.

FAIL:

- The data imported to the database is not correct (check dates)
- There are duplicate rows in the CSV file
- The test file to generate 1000 000 fails for whatever reason.
- The import of this large file fails for whatever reason.