

PHP Developer Technical Test

Notes

* The estimated times help you have a sense of how complex a question is. For example, if you spend 30 minutes on a question which is estimated at 10 minutes, you may not have a good approach.
* For the questions 4 and 5:
* The environment used to run the scripts has Apache 2.4, PHP 7.2, and essential libraries such as cURL, XML, JSON, etc.
* The examiner will put your submitted scripts into his public\_html directory and run them by a web browser. For example, if your script name is *question4.php*, the examiner will open the URL [*http://localhost/question4.php*](http://localhost/question4.php).
* The examiner will not download any libraries or change any configurations. If your scripts rely on some libraries, you should include them on your submission.

Question 1

*Estimated time: 15 mins*

You’re writing a book management application for a bookstore using PHP and MySQL. A book has the following properties:

* Title
* Authors
* Categories
* Published date
* Number of pages
* Quantity
* Condition: New, Used
* Price

That’s all requirements you have for now. How would you design DB tables based on the requirements? List tables, columns, data types, and indexes. Remember that requirements can be changed anytime, so your DB design should be able to deal with potential changes without spending much effort. Note: you don’t have to write SQL statements to create table.

<https://dbdesigner.page.link/2S1QgJ8ZJFRfJd2s6>

Question 2

*Estimated time: 10 mins*

What are the problems of the following code snippets, and how would you improve?

**a) Code snippet 1:**

<?php

/\*\*

\* This scripts list all students of Harvard, one per line

\*/

$students = Student::getAll(); // query DB and get all students into array

foreach ($students as $student) {

if ($student->school = 'Harvard') {

echo $student->name . '\n';

}

}

?>

**Problem**:

The conditional statement of the if statement is an assignment which would not throw an error, but can lead to confusion or bugs that are difficult to detect. In this case, it would always run the code inside of the if statement because it will always suceed.

**How to improve**:

I would change it to the equal or identical operator to produce the correct boolean value.

**b) Code snippet 2:**

<?php

/\*\*

\* From an email provided by end-users,

\* this script retrieves a user from the database

\* then prints the current address of the user.

\* It also prints a message if the city of the user has a "T" letter.

\*/

/\*\* **@var** mysqli $mysqli Given that we already have a MySQLi connection \*/

$username = $\_POST['username']; // get the input username

$result = $mysqli->query("SELECT `firstName`, `currentAddress`, `city` FROM `users` WHERE `username` = '$username' LIMIT 1");

if ($user = $result->fetch\_assoc()) {

echo $user['firstName'] . ' is living at ' . $user['currentAddress'];

$city = strtoupper($user['city']);

if (strpos($city, 'T')) {

echo "\nThis user's city has a T letter.";

}

}

?>

**Problem**:

strpos($city, 'T') will return the index of 0 if T is the first character of the name of the city which would not run the line inside of this if statement. Also the second argument for the strpos() function is case-sensitive so it would ignore all lower-case t's.

**How to improve**:

I would change the conditional expression to

"strpos(strtolower($city), 't') !== false" because it will return false if a 't' was not found. You could also use stripos(), but this would not catch accents on some particular characters. I would also call $result->free\_result() to free the memory associated with the result if $result is no longer in use.

Question 3

*Estimated time: 10 mins*

* How do you check whether a variable is empty string or not?
* I would use the built-in PHP function empty() with a variable we are checking as an argument, where the return value will return the boolean 'false' if the variable contains a falsy (or empty) value.
* You’re debugging a script with 2000 lines of code. You don’t know anything about the script because you didn’t create it. Normally the script should take only 1 second to finish but for some reason, it takes very long. How do you find the buggy lines?
* First I would check if it's only slow on my machine by checking the diagnostic tools, and my PC isn't the issue, then I would add break-points on parts of the scripts in which there may be a suspect or problematic line or code.
* When working on a task, you have questions about business logic but you can’t find answers right away -- maybe you’ll have to wait for 1 day to get answers. What would you do?
* As a junior developer, I wouldn't be very familiar with the business logic and much of the core functionality of the working product, so my first course of action would check the Jira board to study the task or go to the project manager or a senior developer to guage how I should approach the issue.

Question 4

*Estimated time: 25 mins*

A number is considered *beautiful* if the sum of the first half of its digits is equal to the sum of the other half.

For example:

* 123006 is beautiful because 1+2+3 is equal to 0+0+6
* 12347111 is beautiful because 1+2+3+4 is equal to 7+1+1+1
* 12345678 is not beautiful because 1+2+3+4 is different to 5+6+7+8
* 1234567 is not beautiful because its number of digits is odd (7)

Write a function isBeautiful($integer) that takes an integer as a parameter and returns:

* true if the integer is beautiful, or
* false if the integer is not beautiful

Question 5

*Estimated time: 2 hours*

Write a PHP simple webapp to do the following:

**1) At page load, send a POST request to** [**http://dev-trinh.besmartee.com/dev-test/api-interview.php**](http://dev-trinh.besmartee.com/dev-test/api-interview.php)**.**

a) The content of request body should be the same as the provided file **request.xml**. Important: Use SimpleXMLElement’s functions such as addChild() and addAttribute() to build the XML request. Do NOT just load request.xml as string.

b) Post field is very simple -- only 1 key-value pair:

**'xml' => $xmlString**

Where:

* 'xml' is key name
* $xmlString is the XML string you’ve just built

**2) You will receive a response in XML format:**

a) If request failed: You will receive the following response:

<?xml version="1.0"?>

<RESPONSE\_GROUP>

   <STATUS>Failed</STATUS>

</RESPONSE\_GROUP>

A request will be failed if LoginAccountIdentifier or LoginAccountPassword is invalid; or any attribute of <BORROWER> node is empty or missing. Otherwise, request will be successful.

b) If request is successful: You will receive a response just like provided file **response.xml**.

Note: If you have problem connecting to the API endpoint because unexpected reasons such as the endpoint is down or you’re getting SSL certificate error, you’re allowed to load the provided file **response.xml** into memory and go further. Don’t let it stop you!

**3) Once you receive successful response, parse it, and display the data into a table as below:**



Where to find the columns:

* Name of Creditor: /RESPONSE\_GROUP/RESPONSE/RESPONSE\_DATA/CREDIT\_RESPONSE/CREDIT\_LIABILITY/\_CREDITOR/\_Name
* Account Opening Date: /RESPONSE\_GROUP/RESPONSE/RESPONSE\_DATA/CREDIT\_RESPONSE/CREDIT\_LIABILITY/\_AccountOpenedDate
* Outstanding Balance: /RESPONSE\_GROUP/RESPONSE/RESPONSE\_DATA/CREDIT\_RESPONSE/CREDIT\_LIABILITY/\_UnpaidBalanceAmount
* Monthly Payment: /RESPONSE\_GROUP/RESPONSE/RESPONSE\_DATA/CREDIT\_RESPONSE/CREDIT\_LIABILITY/\_MonthlyPaymentAmount
* Account Type: /RESPONSE\_GROUP/RESPONSE/RESPONSE\_DATA/CREDIT\_RESPONSE/CREDIT\_LIABILITY/\_AccountType

**4) On the table, make Name of Creditor a link so that when clicking on it, some HTML will be loaded into a new div below the table using AJAX (no reloading the page).**

a) After clicking on the link, an AJAX request should be sent. The back-end PHP script that receives the AJAX request should send XML request to [**http://dev-trinh.besmartee.com/dev-test/api-interview.php**](http://dev-trinh.besmartee.com/dev-test/api-interview.php) again and receive response, just like steps 1 and 2. Then, the back-end PHP script should parse the XML response to get HTML content on node EMBEDDED\_FILE/DOCUMENT and put into AJAX response.

b) Front-end script should receive AJAX response from step 4a and load into a new div. It should look like below:



Note: HTML content will be the same for all creditors, so do not worry if you see the same thing displaying.

c) Add a *Close* button to clear the div contents dynamically. Clicking the Close button should clear the HTML contents and hide itself.

Feel free to download and use jQuery or Bootstrap if needed.