# Everlytic Developer Assessment

| Name | Lucien Mendela |
| --- | --- |

| Date | 22 June 2023 |
| --- | --- |

*Please complete the answers to the questions below. The assessment should take roughly 30 minutes.*

### What is the difference between public, protected and private in a class definition?

| *A class definition contains variables and methods in a given object, in a public class it basically means we can access the property, or method from anywhere. Protected class means the property and methods can only be read and accessed from within the class itself only, also subclasses can have access to them, this extends and helps with encapsulation and abstraction. Private classes can not be accessed by outside variables and methods, they are usually hidden and only public or protected member functions can have access to them, the difference between private and protected is that with protected subclasses can have access to them but with private can only be access within the class itself.* |
| --- |
|  |
|  |

### Given this code: function doSomething(&$foo) { $bar = $foo; $foo += 1; return $foo; } $value = 3; $result = doSomething($value); echo "value: $value, result: $result"; What will be output to screen and why?

| *The result will be: value: 4, result: 4* |
| --- |
| *The function is initializing $value with the value of 3. When the function is getting called and passing the $value as a ref by using the & operator (paying attention to doSomething(&$foo)), this creates a dependency that any changes to the $foo inside that function it will affect the original variable that is getting passed in. When you look at the doSomething function the two values are assign together $bar = $foo; and it does get initiated with 3, but $foo is incremented by 1 and making it 4, and the return value of the function takes in the the initial value plus the incremented value is what getting returned. $value was passed as a reference, so the changes are what being shown and the return value is also then assigned to $result, hence the output.* |
|  |

### What is wrong with this query: "SELECT \* FROM table WHERE id = $\_POST[ 'id' ]"?

| The query does not meet the best practices of sql, it does create room for sql injection (security vulnerability), putting quotes around an array key especially when you using double quotes, in the cases whereby such a query needs to be run it needs to be in a form of a prepared query, prepared queries , allows us to sanitize user input before running query. |
| --- |
|  |
|  |

### What is the cause of this warning: 'Warning: Cannot modify header information - headers already sent', and what is a good practice to prevent it?

| The cause of this error can usually be whitespaces or hard output before modifying the header or in other cases UTF-8 is what could lead to the error. To prevent this one can always make sure that php code is sent before any form of output is sent to the browser, you can do so by avoiding to echo or even print. Lastly you can also prevent it by making sure that your php file is saved with UTF-8encoding (save using BOM) |
| --- |
|  |
|  |

### What is wrong with this code: class Foo { protected $bar; public function \_\_construct() { $this->bar = 1; } public static function doSomething() { return $this->bar; } }

| In the given code $this is a reference to the current object instance, but the general function is trying to access the non-static property of $bar using $this. This code only works when we make the doSomething() function either $bar static as well or remove the static keyword from doSomething(). |
| --- |
|  |
|  |

### Write a program that prints the numbers from 1 to 100. But for multiples of three print "Fizz" instead of the number and for the multiples of five print "Buzz". For numbers which are multiples of both three and five print "FizzBuzz".

PHP:

for ($i = 1; $i <= 100; $i++) {

if ($i % 3 === 0 && $i % 5 === 0) {

echo "FizzBuzz";

} elseif ($i % 3 === 0) {

echo "Fizz";

} elseif ($i % 5 === 0) {

echo "Buzz";

} else {

echo $i;

}

echo "\n"; // This to print a new line after every word or number

}

Php shorter:  
  
for ($i = 1; $i <= 100; $i++) {

$output = ($i % 3 === 0 ? 'Fizz' : '') . ($i % 5 === 0 ? 'Buzz' : '');

echo $output !== '' ? $output : $i;

echo "\n";

}

Javascript:  
  
for (let i = 1; i <= 100; i++) {

if (i % 3 === 0 && i % 5 === 0) {

console.log("FizzBuzz");

} else if (i % 3 === 0) {

console.log("Fizz");

} else if (i % 5 === 0) {

console.log("Buzz");

} else {

console.log(i);

}

}

Python:

for i in range(1, 101):

if i % 3 == 0 and i % 5 == 0:

print("FizzBuzz")

elif i % 3 == 0:

print("Fizz")

elif i % 5 == 0:

print("Buzz")

else:

print(i)

### What does the following code do? Explain what’s going on there.

$date = '08/26/2003';

print preg\_replace('/([0-9]+)\/([0-9]+)\/([0-9]+)/'‚ '$2/$1/$3', $date);  
The code is using the preg\_replace() function in PHP to modify the format of the given code. It takes in the code in the format of MM/DD/YYYY and rearranges it to the format DD/MM/YYYY and then prints it

|  |
| --- |
|  |

### Given a line of text $string, how would you write a regular expression to strip all the HTML tags from it? I will still use the preg\_replace() function with this `/<[^>]\*>/’ as a regular function. How it works is that the regular expression removes the HTML tags with an empty string.

|  |
| --- |
|  |
|  |

### A palindrome is a word that reads the same backward or forward. Write a function that checks is a given word is a palindrome. Characters case should be ignored. EG. Deleveled is a palindrome and should return true as character case is ignored.

### <?php class Palindrome

### {

### public static function isPalindrome($word)

{

// take the given word and convert it to lowercase

$word = strtolower($word);

// make sure we only working with numbers

$word = preg\_replace('/[^a-z0-9]/', '', $word);

// now we can reverse the word

$reversed = strrev($word);

// check if the reversed word is the same as the original word

return $reversed === $word;

}

### } echo Palindrome::isPalindrome('Deleveled');

### Considering message\_text stores a combination of html and text. What security issue is prevalent in the code below and how would you fix it? <?php $messageStmt = $db->query('select message\_text from messages where message\_id = 1'); $messageStmt->execute(); $message = $messageStmt->fetch(PDO::FETCH\_OBJ); ?> <div><?php echo $message->message\_text; ?></div> \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

### The application includes user-generated content in the html output which is the first problem this as a problem cause a security issue in this given case is XSS, because there is untrusted data which is the : $messageStmt->execute();

The problem with the given code is that it does output this unstrusted data without sanitizing or validating the the data. Attackers can inject malicious code into the page.This could be fix by using the htmlspecialchars() function to encode some of this special characters so that when they output them there is no malicious use and can’t be interpreted as html code. By using the best practices you prevent attackers from stealing sensitive information or perform other operations that they shouldn’t be allowed to perform, by simple adding the <scrip> whateverIwantToDoFunctionHere();</scrip> the attacker can damage or attack your website.

### 11.Write an inner join for the following tables

| 1. Method one:   *SELECT \**  *FROM User*  *INNER JOIN Address ON User.UserKey = Address.UsrKey;*   1. Method two:   SELECT \*  FROM User  JOIN Address  ON User.UserKey = Address.UsrKey; |
| --- |
|  |

### Complete the JS function below that validates the conditions of a password: 1. The password must be greater then 7 characters 2. The first character must be a capital letter 3. The password must contain at least one number

function isPasswordValid($password) {

var isValid = false;

// Write you logic here

if (password.length > 7) {

//first requirement checking if the first character is in caps using regular expression

if (/^[A-Z]/.test(password)) {

// Check if the password contains at least one number using regular expression

if (/\d/.test(password)) {

isValid = true;

}

}

}

return isValid;

}

Method 2:  
  
*function isPasswordValid(password) {*

*var isValid = false;*

*if (password.length > 7 && /^[A-Z]/.test(password) && /\d/.test(password)) {*

*isValid = true;*

*}*

*return isValid;*

*}*