# Unit Test Report

**Unit**: Logging out with Session ID.

**Source File**: /hospital/hms/logout.php

**Date**: October 12, 2023

**Engineers**: Sai Mukthive, Srikanth Guttikonda, Goutham Bommu, Sai Gunturu

**Automated Test Code:**

***<?php***

***use PHPUnit\Framework\TestCase;***

***function logout\_user\_with\_session\_id($con, $session\_id) {***

***if($session\_id){***

***date\_default\_timezone\_set('USA/Phoenix');***

***$ldate = date('d-m-Y h:i:s A', time());***

***// Update the logout time in the userlog table based on the session ID***

***$con->query("UPDATE userlog SET logout = '$ldate' WHERE uid = '$session\_id' ORDER BY id DESC LIMIT 1");***

***return [***

***"message" => "You have successfully logged out.",***

***"redirect\_url" => "../index.php"***

***];***

***}else{***

***return [***

***"message" => "Unable to log out.",***

***"redirect\_url" => null***

***];***

***}***

***}***

***class LogoutWithSessionIDTest extends TestCase***

***{***

***private $db\_conn\_mock;***

***protected function setUp(): void***

***{***

***// Mock the mysqli object***

***$this->db\_conn\_mock = $this->getMockBuilder(mysqli::class)***

***->disableOriginalConstructor()***

***->getMock();***

***}***

***public function testLogoutUserWithSessionID()***

***{***

***$session\_id = "123";***

***// Mock the $mysqli->query() method for the logout update***

***$this->db\_conn\_mock->expects($this->once())***

***->method('query')***

***->willReturn(true);***

***$response = logout\_user\_with\_session\_id($this->db\_conn\_mock, $session\_id);***

***$this->assertEquals("You have successfully logged out.", $response['message']);***

***$this->assertEquals("../index.php", $response['redirect\_url']);***

***echo ("Test 1: Successfully logged out \n");***

***}***

***public function testUnsuccessfulLogoutUserWithSessionID()***

***{***

***$session\_id = "";***

***// Mock the $mysqli->query() method for the logout update***

***$this->db\_conn\_mock->expects($this->never())***

***->method('query')***

***->willReturn(false);***

***$response = logout\_user\_with\_session\_id($this->db\_conn\_mock, $session\_id);***

***$this->assertEquals("Unable to log out.", $response['message']);***

***$this->assertNull($response['redirect\_url']);***

***echo ("Test 2: Error logging out \n");***

***}***

***}***

***?>***

**Inputs:**

Database Connection: Mocked MySQLi connection.

User Session ID: A unique session identifier.

**Expected Outputs:**

**a) Successful registration:**

Message: "You have successfully logged out"

RedirectURL: “index.php”

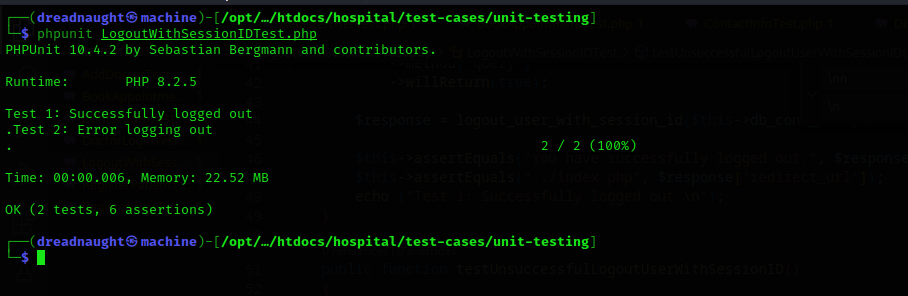
**b) Unsuccessful registration:**

Message: "Booking failed."

RedirectURL: “null”

**Actual Outputs:**

The test executed without any errors and provided the expected message and redirect URL.



**Test Methodology:**

**Method:**

Mocking and Unit Testing.

**Description:**

The logout\_user\_with\_session\_id function is structured to facilitate the logout process for a user based on their session ID. By updating the logout time in the system's userlog table, this function aims to provide a more secure and traceable user experience.

The essence of using mocking was to create a controlled environment wherein the behavior of external factors, like database interactions, could be simulated. This ensures that any discrepancies identified are purely a consequence of the logic within the logout\_user\_with\_session\_id function, rather than external influences.

The tests were divided into two scenarios: one where a valid session ID ensures a successful logout, and another where an absence of a session ID results in a failed logout attempt. This approach ensures that the function handles both anticipated and unanticipated situations gracefully.