* Unit test Đăng nhập

use PHPUnit\Framework\TestCase;

class LoginTest extends TestCase

{

public function testLoginWithValidCredentials()

{

$auth = new Auth();

$result = $auth->login('user@example.com', 'password');

$this->assertEquals('Login successful!', $result);

}

public function testLoginWithInvalidCredentials()

{

$auth = new Auth();

$result = $auth->login('user@example.com', 'wrongpassword');

$this->assertEquals('Incorrect email or password.', $result);

}

public function testLoginWithoutEmail()

{

$auth = new Auth();

$result = $auth->login('', 'password');

$this->assertEquals('Email is required.', $result);

}

public function testLoginWithoutPassword()

{

$auth = new Auth();

$result = $auth->login('user@example.com', '');

$this->assertEquals('Password is required.', $result);

}

}

* Unit test Thêm sp

public function testCreateProduct()

{

// Tạo mock data cho sản phẩm mới

$productData = [

'name' => 'Test Product',

'price' => 100,

'description' => 'This is a test product.',

];

// Gọi hàm thêm sản phẩm mới

$response = $this->post('/products', $productData);

// Kiểm tra xem response status code có phải là 302 không

$response->assertStatus(302);

// Kiểm tra xem sản phẩm đã được thêm vào database chưa

$this->assertDatabaseHas('products', $productData);

}

* Xoá sp

public function testDeleteProduct()

{

// Tạo mock data cho sản phẩm cần xóa

$product = new Product([

'name' => 'Test Product',

'price' => 100,

'description' => 'This is a test product.',

]);

$product->save();

// Gọi hàm xóa sản phẩm

$response = $this->delete('/products/'.$product->id);

// Kiểm tra xem response status code có phải là 302 không

$response->assertStatus(302);

// Kiểm tra xem sản phẩm đã được xóa khỏi database chưa

$this->assertDatabaseMissing('products', ['id' => $product->id]);

}

* Sửa sp

public function testUpdateProduct()

{

// Tạo mock data cho sản phẩm cần sửa

$product = new Product([

'name' => 'Test Product',

'price' => 100,

'description' => 'This is a test product.',

]);

$product->save();

// Tạo mock data mới cho sản phẩm

$newProductData = [

'name' => 'New Test Product',

'price' => 200,

'description' => 'This is a new test product.',

];

// Gọi hàm sửa sản phẩm

$response = $this->put('/products/'.$product->id, $newProductData);

// Kiểm tra xem response status code có phải là 302 không

$response->assertStatus(302);

// Kiểm tra xem sản phẩm đã được sửa trong database chưa

$this->assertDatabaseHas('products', $newProductData);

}