Logo

Description automatically generated

Software Engineering  
 Project

Phase IV

By: Helisa Ormëni

Saeda Haka

Epoka University, Tirana Albania 2024

Your task is to test a specific component of your software. Choose a critical class, module, or function ... Prepare comprehensive test cases, write test code, and execute tests.

Submit your assignment report by the deadline.

1. **Introduction to Testing:**
   1. Define software testing as the process of evaluating software to identify defects or bugs.
   2. Highlight the importance of testing in software development to ensure reliability and correctness.
2. **Purpose of Testing:**
   1. Explain that testing aims to identify defects early in the development process and verify that software components perform as intended.
3. **Focus on Testing a Single Component:**
   1. Choose a component **(class, module, function, etc.) from your codebase for testing.**
   2. Explain why testing this component is important, considering its role, complexity, and impact on the system.
4. **Preparing Test Cases:**
   1. To cover various scenarios, you've included test cases for normal inputs (e.g., complete form submissions), edge cases (e.g., missing required fields), and invalid inputs (e.g., incorrect data formats).
5. **Choosing Testing Frameworks:**
   1. For PHP, you're using PHPUnit for unit testing. This is a widely used framework for PHP testing.
   2. Setting up the Testing Environment:
   3. Ensure PHPUnit is installed and configured properly in your development environment.
   4. Make sure any dependencies required for testing are installed and accessible.
6. **Writing Test Code:**
   1. You've provided examples of test code for PHP components, including form validation and database connection.
   2. Test methods are created to exercise different functionalities (e.g., submitting the form with missing data).
   3. Assertions are used to validate expected outcomes (e.g., checking for error messages in the response).
7. **Running Tests:**
   1. Execute tests using PHPUnit or the appropriate testing command for your chosen framework.
   2. Interpret results to identify passing, failing, and error scenarios.
   3. Investigate failures or errors to understand the cause and fix issues as needed.
8. **Test Coverage:**
   1. High test coverage is important to ensure thorough testing of the software.
   2. Ensure that test cases cover all critical functionalities and edge cases to minimize the risk of undetected bugs.

* Form validation test

<?php

use PHPUnit\Framework\**TestCase**;

class **FormValidationTest** extends **TestCase** {

    protected $formUrl;

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

        $this->formUrl = 'http://localhost/Pet-Website/app/process\_form.php'; *// Update with your form URL*

    }

*// Test form submission with missing owner email*

    public function **testMissingOwnerEmail**() {

        $postData = [

            'appointment-type' => 'medical-appt',

            'pet-type' => 'Dog',

            'pet-name' => 'Buddy',

            'owner-name' => 'John Doe',

            'owner-number' => '1234567890',

            'appointment-date' => '2024-05-01',

            'appointment-time' => '14:00'

        ];

        $response = $this->**submitForm**($postData);

        $this->**assertStringContainsString**('Email is required', $response);

    }

*// Test form submission with missing owner phone number*

    public function **testMissingOwnerNumber**() {

        $postData = [

            'appointment-type' => 'medical-appt',

            'pet-type' => 'Dog',

            'pet-name' => 'Buddy',

            'owner-name' => 'John Doe',

            'owner-email' => 'john@example.com',

            'appointment-date' => '2024-05-01',

            'appointment-time' => '14:00'

        ];

        $response = $this->**submitForm**($postData);

        $this->**assertStringContainsString**('Phone number is required', $response);

    }

    public function **testMissingPetType**()

    {

*// Simulate form submission with missing pet type*

        $formData = [

            'appointment-type' => 'medical-appt',

            'pet-name' => 'Buddy',

            'owner-name' => 'John Doe',

            'owner-number' => '1234567890',

            'owner-email' => 'john@example.com',

            'appointment-date' => '2024-05-01',

            'appointment-time' => '12:00',

        ];

        $response = $this->**submitForm**($formData);

*// Assert that the response contains the expected error message*

        $this->**assertStringContainsString**('Pet type is required', $response);

    }

    public function **testMissingPetName**()

    {

*// Simulate form submission with missing pet name*

        $formData = [

            'appointment-type' => 'medical-appt',

            'pet-type' => 'Dog',

            'owner-name' => 'John Doe',

            'owner-number' => '1234567890',

            'owner-email' => 'john@example.com',

            'appointment-date' => '2024-05-01',

            'appointment-time' => '12:00',

        ];

        $response = $this->**submitForm**($formData);

*// Assert that the response contains the expected error message*

        $this->**assertStringContainsString**('Pet name is required', $response);

    }

    public function **testMissingOwnerName**()

    {

*// Simulate form submission with missing owner name*

        $formData = [

            'appointment-type' => 'medical-appt',

            'pet-type' => 'Dog',

            'pet-name' => 'Buddy',

            'owner-number' => '1234567890',

            'owner-email' => 'john@example.com',

            'appointment-date' => '2024-05-01',

            'appointment-time' => '12:00',

        ];

        $response = $this->**submitForm**($formData);

*// Assert that the response contains the expected error message*

        $this->**assertStringContainsString**('Owner name is required', $response);

    }

*// Function to simulate form submission and return response*

    protected function **submitForm**($postData) {

        $ch = **curl\_init**($this->formUrl);

**curl\_setopt**($ch, CURLOPT\_RETURNTRANSFER, true);

**curl\_setopt**($ch, CURLOPT\_POST, true);

**curl\_setopt**($ch, CURLOPT\_POSTFIELDS, $postData);

        $response = **curl\_exec**($ch);

**curl\_close**($ch);

        return $response;

    }

}

* Output of this test

A computer screen shot of a computer program

Description automatically generated

* Process form test

<?php

use PHPUnit\Framework\**TestCase**;

include 'app/process\_form.php';

class **ProcessFormTest** extends **TestCase** {

*// Test database connection*

    public function **testDatabaseConnection**() {

        $servername = "localhost";

        $username = "root";

        $password = "12345678";

        $dbname = "pet\_care\_vet";

        $conn = new **mysqli**($servername, $username, $password, $dbname);

        $this->**assertTrue**($conn->connect\_error === null);

    }

}

* Output of this test



A white background with black and white clouds

Description automatically generated with medium confidence

* Here are some written test case scenarios for each file in the provided HTML:

HTML File Test Case Scenarios:

* **Header and Navigation:**

1. Verify that the logo "PAW CARE" is displayed in the header.

2. Test that the navigation menu contains the following items: "Services", "Book an appointment", and "About Us".

3. Click on the "Services" dropdown menu and verify that it contains the options "Grooming" and "Medical Assistance".

4. Click on the "Book an appointment" link in the navigation menu and ensure that it scrolls to the correct section on the page.

5. Check that the "About Us" link in the navigation menu leads to the appropriate page or section.

* **Hero Section:**

1. Confirm that the hero section displays the message "You know what we like about you? Your pet!".

2. Click on the "Book a Service" button and verify that it redirects to the correct page.

* **Services Section:**

1. Test that both "Dog Grooming" and "Medical Assistance" services are displayed.

2. Click on the "Book Now" button for "Dog Grooming" and ensure it redirects to the correct page.

3. Click on the "Book Now" button for "Medical Assistance" and ensure it redirects to the correct page.

4. Verify that the prices for each service are displayed correctly.

5. Check that the images for each service are loaded and displayed properly.

* **Appointment Section:**

1. Ensure that the "Book an appointment" section displays the appropriate heading and description.

2. Click on the "Appointment Application" button and verify that it redirects to the correct page or form.

* Footer:

1. Verify that the footer displays the copyright message "© Paw care, All rights reserved".

* **General:**

1. Test the functionality of the `redirectToPage()` function to ensure it redirects to the correct pages.

2. Check that the page loads without any console errors.

3. Test the behavior of the page when accessed with different screen sizes or devices to ensure responsiveness.

* These test case scenarios cover various aspects of the HTML file, including navigation, content, functionality, and responsiveness, ensuring that the web page behaves as expected across different scenarios.