

**NATIONAL UNIVERSITY OF SCIENCES AND TECHNOLOGY**

# Project Report

**SE-314 Software Construction**

|  |  |  |  |
| --- | --- | --- | --- |
| Name | CMS Id | Class | Section |
| Ayesha Siddiqa | 407198 | SE-13 | B |
| Navaal Iqbal | 409977 | SE-13 | B |
| Amna Ahmed | 408099 | SE-13 | B |

**Introduction**

The Bonsai Pet Shop project is designed to provide an online platform for purchasing pets (cats and dogs) and DIY pet kits. The system is developed using Laravel, leveraging its MVC architecture to ensure scalability, security, and maintainability.

**SYSTEM DESIGN**

## 1. ****System Architecture****

### ****Framework****

* **Backend:** Laravel (MVC Pattern)
* **Frontend:** Blade templates with optional Vue.js/React components for interactivity
* **Database:** MySQL
* **Hosting:** local host

### ****System Components****

1. **Client Side:**
   * User interfaces for Admin, Registered Users, and Guests
   * Dynamic content using JavaScript and CSS
2. **Server Side:**
   * Laravel controllers and middleware for request handling
   * RESTful APIs for external integration
3. **Database Layer:**
   * Relational database to store and manage data

## 2. ****Core Features****

### ****Pet Management****

* Add, update, delete, and display pet details.
* Attributes: Name, Breed, Age, Gender, Price, Description, and Images.

### ****DIY Pet Kit Management****

* Add, update, delete, and display DIY pet kits.
* Attributes: Name, Description, Price, Included Items, and Images.

### ****Feedback/Reviews****

* Registered users can leave reviews for pets or kits.
* Admins can moderate reviews.

### ****Authentication and Authorization****

* User registration and login (email verification).
* Role-based access control for Admins, Registered Users, and Guests.

## 3. ****User Roles and Responsibilities****

### ****Admin****

* Manage pets and DIY kits.

### ****Guests****

* Browse pets and kits.

## 4. ****User Interface Design****

### ****Home Page****

* Overview of the shop with featured pets and kits.
* Navigation bar with links to About, Gallery, Services, Products, and Contact Us.

### ****About Page****

* Detailed information about the Bonsai Pet Shop, its mission, and values.

### ****Gallery Page****

* A visual showcase of available pets, kits, and shop highlights.
* Grid layout with high-quality images.

### ****Services Page****

* List of additional services provided by the shop (e.g., pet grooming, training, or consultation).

### ****Products Page****

* Catalog of all pets and DIY kits.
* Grid layout with images, prices, and quick actions (Add to Cart, View Details).

### ****Contact Us Page****

* Contact form for inquiries.
* Shop location, phone number, and email address

## 5. ****Database Design****

### ****Tables and Attributes****

1. **Users:**
   * ID, Name, Email, Password, Role, Created At, Updated At.
2. **Products:**
   * ID, Name, Category (Pet/DIY Kit), Description, Price, Images, Created At, Updated At.

## 6. ****Usability Considerations****

* **Responsive Design:** Mobile-friendly interface.
* **Performance Optimization:** Use caching for frequently accessed data.
* **User Feedback:** Provide clear error messages and intuitive navigation.

### Unit and Integration Test Descriptions

Below is the description of the unit and integration tests for the provided controllers:

#### **Unit Tests**

##### **1. ProductControllerTest (Unit Tests)**

* **test\_create\_product\_instance**:  
  Verifies that a product instance is created with the correct attributes. Ensures the Product model's constructor properly assigns values for category, name, description, and price.
* **test\_update\_product\_instance**:  
  Tests the ability to update attributes of a product instance. Confirms the Product model allows modifications to existing attributes such as category, name, description, and price.
* **test\_retrieve\_product\_attributes**:  
  Ensures that the attributes of a Product instance can be retrieved correctly. Validates the Product model's getters for all relevant attributes.

#### **Feature Tests**

##### **2. ProductControllerTest (Feature Tests)**

* **it\_displays\_all\_pet\_products**:  
  Checks if the index method of the ProductController returns all products to the view. Validates the correct HTTP status, the expected view (products), and the presence of products in the view data.
* **it\_stores\_a\_product**:  
  Tests the store method for valid input. Ensures that a product is successfully stored in the database and redirects the user to the products index page.
* **it\_validates\_product\_data\_when\_storing**:  
  Confirms validation rules during product creation. Ensures that incomplete or incorrect data results in validation errors for fields like category, name, and description.
* **it\_updates\_a\_product**:  
  Verifies the update functionality. Ensures that the controller updates product attributes in the database and redirects to the index page upon success.
* **it\_deletes\_a\_product**:  
  Tests the delete functionality of the ProductController. Ensures that a product is removed from the database and the user is redirected to the index page.

##### **3. HomeControllerTest (Unit Tests)**

* **test\_auth\_middleware\_is\_applied**:  
  Validates that the auth middleware is correctly applied to the HomeController. Ensures unauthenticated users are redirected to the login page.
* **test\_index\_view\_for\_authenticated\_user**:  
  Verifies that the index method returns the correct view (home) for authenticated users.
* **test\_unauthenticated\_users\_redirected\_to\_login**:  
  Confirms that unauthenticated users trying to access the home page are redirected to the login route.
* **test\_authenticated\_user\_can\_access\_home\_page**:  
  Ensures that authenticated users can successfully access the home page and the correct view (home) is returned.

##### **4. HomeControllerTest (Feature Tests)**

* **test\_authenticated\_user\_can\_access\_home\_page**:  
  Ensures that only authenticated users can access the home route. Confirms that the correct HTTP status and view (home) are returned.
* **test\_unauthenticated\_user\_is\_redirected\_to\_login**:  
  Verifies that unauthenticated users are redirected to the login page when attempting to access the home route.