



Green University of Bangladesh

Department of Computer Science and Engineering (CSE)

Semester: (Spring, Year: 2025), B.Sc. in CSE (Day)

PET Management System

Course Title: Integrated Design Project II

Course Code : CSE-406

Section : 213 D7

Students Details

Name	ID
MD Dulal Hossain	213902116
MD Rabby Khan	213902037
Mostak Ahammed	213902126

Submission Date: 20 - May - 2025

Course Teacher's Name: Sharifur Rahman

Designation: Lecturer

Department of CSE , GUB .

[For teachers use only: **Don't write anything inside this box**]

<u>Lab Project Status</u>	
Marks:	Signature:
Comments:	Date:

Contents

1	Introduction	3
1.1	Overview	3
1.2	Problem Domain	3
1.3	Motivation	4
1.4	Objectives	4
1.5	Application	4
2	Requirement Analysis and Design	5
2.1	Requirement Analysis	5
2.1.1	Functional Requirements	5
2.1.2	Non-Functional Requirements	5
2.2	Tools and Techniques	6
2.3	Gantt Chart	7
2.4	Software development life cycle (SDLC) Selection	8
2.5	Data Flow Diagram	8
2.5.1	Level-01 DFD	8
2.6	Sequence Diagram	9
2.7	System Architecture	9
2.7.1	High-Level Design	9
2.7.2	Low-Level Design	10
2.8	E-R Diagram	11
2.9	Database Schema	12
2.10	Conclusion	12
3	Interface Design and Implementation	13
3.1	System Overview	13
3.2	Homepage Interface	13
3.3	User Registration and Login	14
3.3.1	User Registration	14

3.3.2	User Login	14
3.4	User Dashboard	15
3.5	Admin Dashboard	15
3.6	Pet Profile Management	16
3.7	Doctor Dashboard and Treatment Interface	16
3.8	Addoption	17
3.9	Addoption Booking	17
3.10	Services	18
3.11	Products	18
3.12	Teams And Testimonials	19
3.13	Databases Structure	19
3.14	Implementation Flow	20
3.15	Platform and Tools Used	20
3.16	System Architecture Diagram	20
3.17	Conclusion	20
4	Test Cases for PET Management System	21
4.1	Testing Strategy	21
4.2	Sample Test Cases	21
4.2.1	Test Case: User Registration	21
4.3	Test Case Summary Table	22
4.4	Test Result / Output	22
4.4.1	Register New User - Test Result: Pass	22
4.4.2	Register Existing User - Test Result: Pass	23
4.4.3	Long Password - Test Result: Fail	23
4.4.4	Valid Login - Test Result: Pass	24
4.4.5	Invalid Login - Test Result: Pass	24
4.4.6	SQL Injection in Filter - Test Result: Pass	25
4.5	Analysis and Discussion	25
4.6	Conclusion	25
5	Conclusion and Future Work	26
5.1	Summary of the Project	26
5.2	Societal Impact	26
5.3	Future Work	27

Chapter 1

Introduction

1.1 Overview

The PET Management System is a comprehensive web-based platform designed to streamline pet care and management services. It provides users with an intuitive interface to explore various pet-related services, including grooming, veterinary consultations, pet adoption, and product purchases. The system offers essential features such as a service catalog, pricing details, customer testimonials, and team profiles to ensure transparency and user engagement.

The platform is structured with multiple pages, including Home, About, Services, Products, Contact, and Blog, enabling seamless navigation for users. Businesses and pet owners can efficiently manage appointments, inquiries, and transactions through an integrated system, enhancing customer satisfaction.

Built with modern web technologies, the PET Management System aims to bridge the gap between pet service providers and pet owners by offering a centralized digital solution. Its user-friendly design and functionality make it a valuable tool for individuals looking to provide the best care for their pets.

1.2 Problem Domain

- **Lack of Centralized Pet Management:** Pet owners often struggle to find a single platform for managing pet-related services, including veterinary care, grooming, and product purchases.
- **Inefficient Appointment Scheduling:** Traditional booking methods for pet services are time-consuming and lack automation, leading to scheduling conflicts and mismanagement.
- **Limited Access to Reliable Pet Services:** Users face difficulties in locating trusted service providers, reading customer reviews, and comparing service offerings.
- **Communication Gaps:** Poor interaction between pet owners and service providers leads to misunderstandings regarding pet health, needs, and available services.
- **Lack of Digitalization:** Many pet care businesses rely on manual processes, limiting efficiency, accessibility, and customer engagement.

1.3 Motivation

The increasing number of pet owners worldwide has created a growing demand for efficient and reliable pet care services. However, many pet owners struggle to find a centralized platform that offers seamless access to veterinary care, grooming, pet adoption, and product purchases. The lack of digitalization in pet service management often results in scheduling conflicts, poor communication, and difficulty in finding trusted providers.

This project aims to bridge the gap by providing an all-in-one solution that enhances accessibility, efficiency, and convenience. By integrating appointment scheduling, service listings, and customer reviews, the PET Management System ensures a user-friendly experience. The motivation behind this project is to create a streamlined, digital pet care solution that benefits both pet owners and service providers.

1.4 Objectives

- **Develop a Centralized Platform:** Create an all-in-one web-based system where pet owners can easily access various pet care services, including veterinary consultations, grooming, and product purchases.
- **Automate Appointment Scheduling:** Implement an efficient booking system to streamline scheduling, reduce conflicts, and enhance user convenience.
- **Enhance Service Accessibility:** Provide users with a categorized list of pet care services, complete with descriptions, pricing, and customer reviews to ensure informed decision-making.
- **Improve Communication:** Enable seamless interaction between pet owners and service providers through contact forms, inquiries, and service details.
- **Ensure User-Friendly Navigation:** Design an intuitive interface that allows users to explore and utilize the system effortlessly, ensuring a smooth and engaging experience.

1.5 Application

The PET Management System is designed for various stakeholders in the pet care industry, offering practical applications for both pet owners and service providers. Pet owners can use the platform to schedule veterinary appointments, book grooming sessions, explore pet adoption options, and purchase essential pet products. By providing detailed service listings, pricing, and customer reviews, the system helps users make informed decisions regarding their pets' well-being.

Pet care businesses, including veterinary clinics, grooming centers, and pet supply stores, can leverage the platform to manage appointments, showcase services, and expand their customer base. The system also serves as a communication bridge between service providers and customers, ensuring seamless interaction.

Overall, the PET Management System enhances accessibility, efficiency, and organization, making pet care management more convenient for everyone involved.

Chapter 2

Requirement Analysis and Design

2.1 Requirement Analysis

2.1.1 Functional Requirements

The functional requirements define the specific behavior and functionality of the PET Management System. These include:

- User Registration and Login functionality for Admin, Doctor, and User roles.
- Admin functionalities:
 - Add/edit/delete users and doctors.
 - View and manage appointments.
 - Generate reports based on doctor, user, and date.
- Doctor functionalities:
 - View assigned appointments.
 - Add pet treatment records and notes.
 - View medical history of assigned pets.
- User functionalities:
 - Register and login to the system.
 - Add pet details.
 - Book, view, and cancel appointments.
 - View pet history and treatment records.
- Appointment reminders and status updates.

2.1.2 Non-Functional Requirements

These requirements focus on the system's quality attributes:

- **Performance:** The system must respond to user interactions within 1-2 seconds.

- **Usability:** User interfaces should be simple and intuitive.
- **Security:** Only authenticated users can access protected features. Passwords are hashed.
- **Scalability:** The application should support a growing number of users and pets.
- **Availability:** The system should remain available 99% of the time during business hours.
- **Maintainability:** Code should follow modular design and MVC architecture for easy updates.

2.2 Tools and Techniques

The development of the **Pet Management System** employed a range of tools and technologies to ensure effective design, development, and deployment. The following summarizes the tools used in various aspects of the project:

- **Frontend Technologies:** HTML5 and CSS3 were used for the structure and style of the user interface, while Bootstrap was utilized to provide responsive design and component consistency across different screen sizes.
- **Backend Framework:** The backend was developed using PHP with the Laravel framework, which offers a clean and robust MVC architecture, built-in authentication, routing, and ORM capabilities.
- **Database Management:** MySQL served as the relational database system, with Laravel's Eloquent ORM facilitating smooth and efficient interaction between the application and the database.
- **Version Control:** Git was used as the version control system for tracking changes and managing the codebase. GitHub served as the remote repository for team collaboration and backup.
- **Diagramming Tools:** Draw.io was used to create essential system diagrams, including Data Flow Diagrams (DFDs) and Entity-Relationship Diagrams (ERDs), aiding in system analysis and design.
- **Project Management:** Gantt charts were prepared using Office Timeline and GanttProject to plan, organize, and monitor project tasks and deadlines effectively.

2.3 Gantt Chart

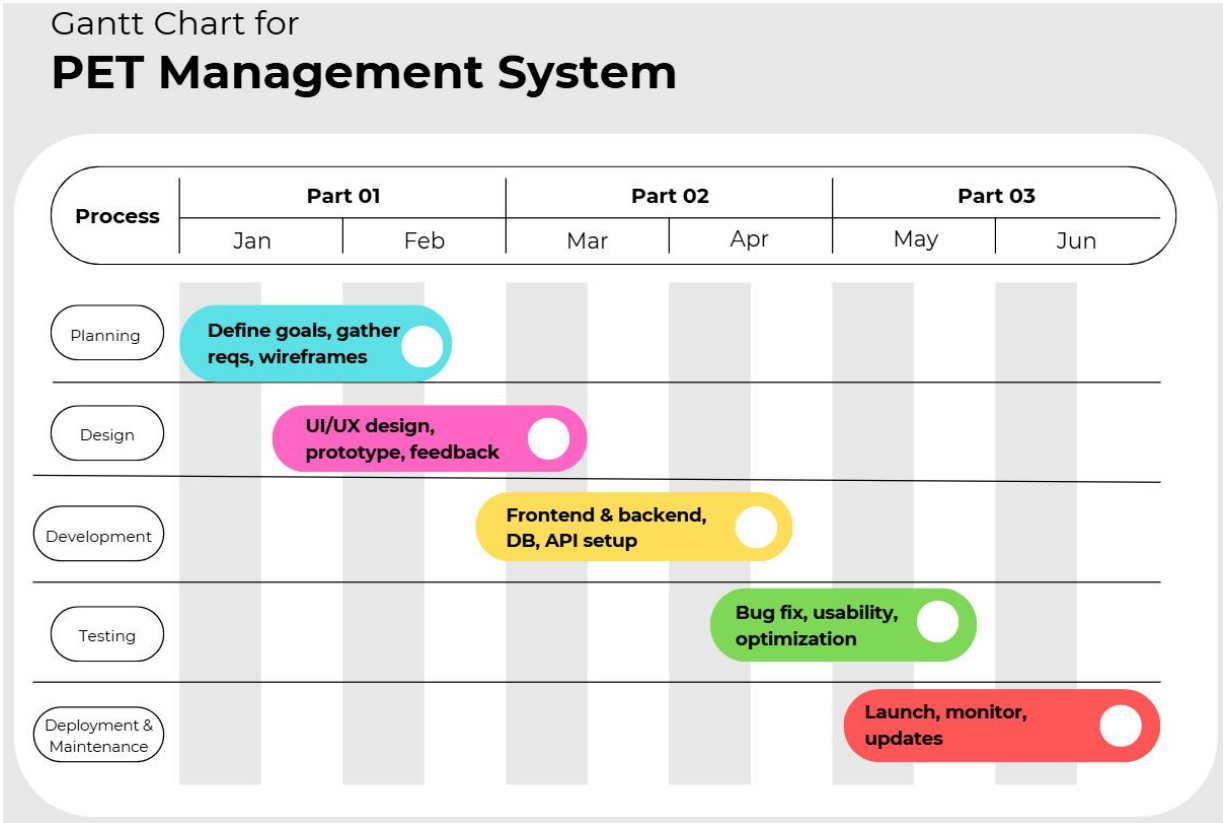


Figure 2.1: Gantt Chart for PET Management System

2.4 Software development life cycle (SDLC) Selection

Here we can see a table of SDLC models for the social media apps project. We have chosen some suitable departments for our project. Then we choose some priority above the requirements for our project. Then see which category is supported by which model. Then we chose the best suitable model above the priority we have set for each section. Then we will set this SDLC model for our project. In these SDLC models, Waterfall score 11, V-shaped score 27, Iterative score 11, Spiral score 22, Agile score 5 and Prototype score 11. Here are our most important sections supported by V-shaped models. So we'll like V-shaped model as the best option for our project.

Table 2.1: Comparison Matrix with Different Models for Social Media App

Priority	Criteria	Waterfall	V-shape	Iterative	Spiral	Agile	Prototype
5	Well known requirement	Yes	Yes	No	No	Yes	No
3	Technological knowledge	Yes	Yes	No	No	No	No
6	Efficiency	No	Yes	Yes	Yes	No	Yes
3	Risk analysis	No	No	No	Yes	No	No
5	User testing ability	No	Yes	Yes	Yes	Yes	Yes
5	Dependability and Security	No	Yes	No	Yes	No	No
3	Time consuming	Yes	Yes	No	Yes	No	No
Total- 30	Overall	11	27	11	22	10	11

2.5 Data Flow Diagram

2.5.1 Level-01 DFD

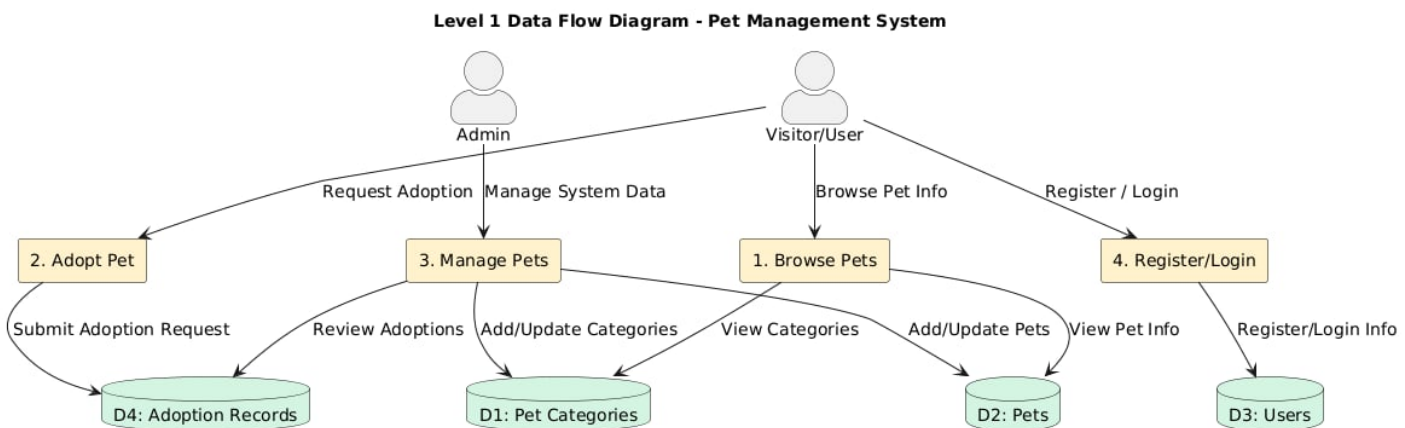


Figure 2.2: Level-1 Data Flow Diagram

2.6 Sequence Diagram

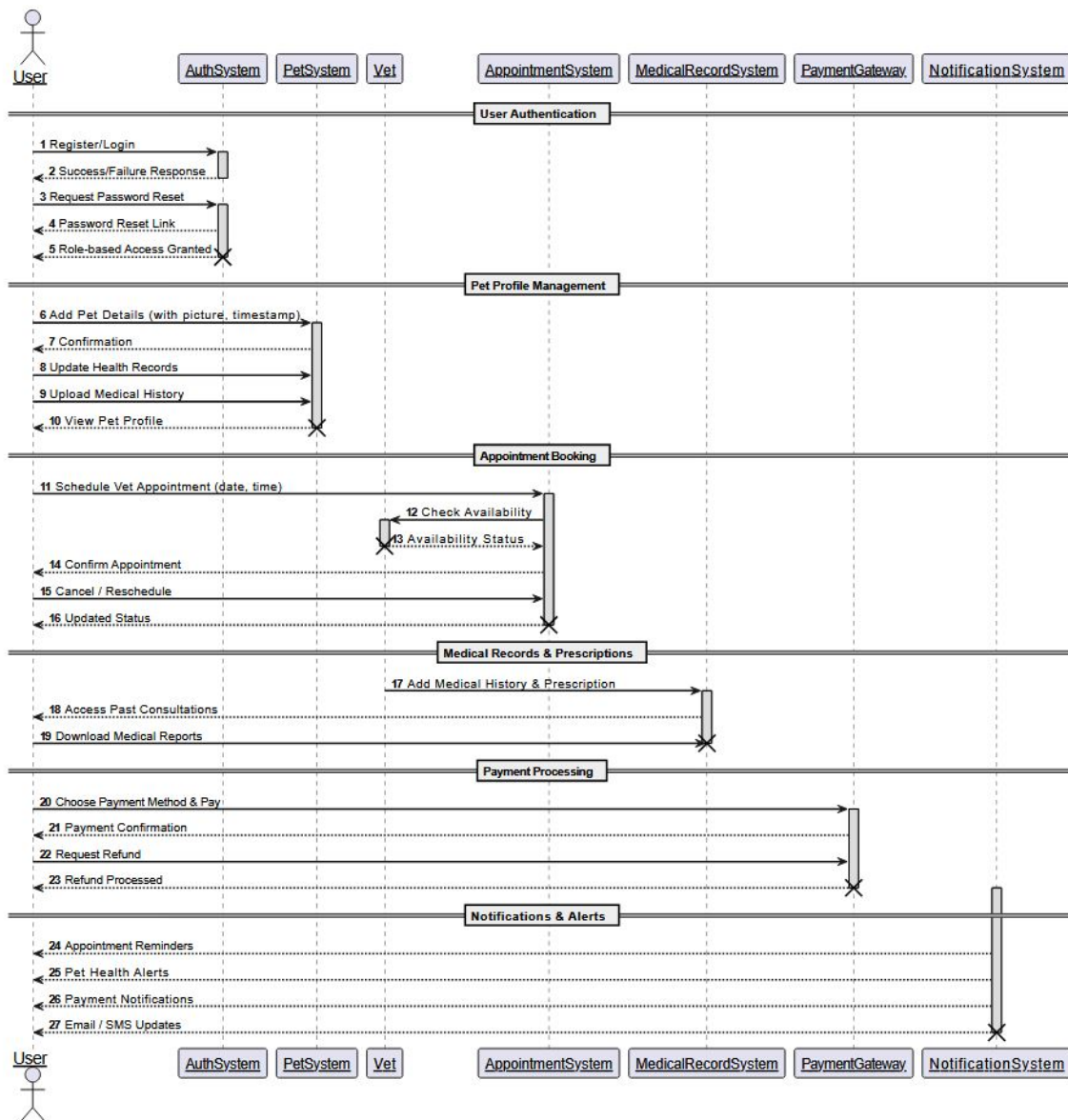


Figure 2.3: Sequence Diagram of PET Management System

2.7 System Architecture

2.7.1 High-Level Design

The system follows a three-tier architecture:

- **Presentation Layer:** The user interface (HTML, CSS, Bootstrap).
- **Application Layer:** Laravel controllers and services handle business logic.
- **Data Layer:** MySQL database stores user, pet, appointment, and report data.

2.7.2 Low-Level Design

The system modules are split into:

- Authentication module (login, register)
- User module (manage pets, appointments)
- Admin module (CRUD operations for users, doctors, appointments)
- Doctor module (treatment updates, medical history)
- Report module (generate reports by filters)

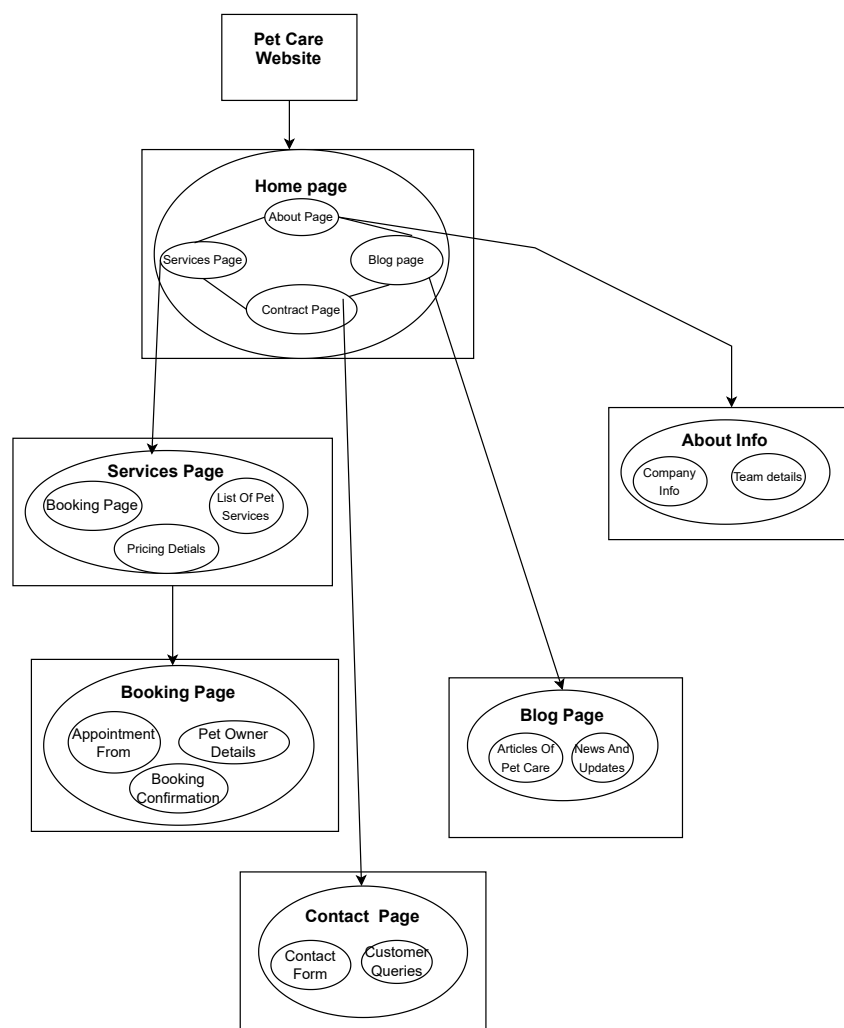


Figure 2.4: Low Level Design (LLD) Diagram.

2.8 E-R Diagram

The Entity-Relationship (E-R) Diagram illustrated above represents the data structure of the Pet Management System, which supports functionalities such as pet adoption, service booking, and veterinary appointments. The key entities and their relationships are described below:

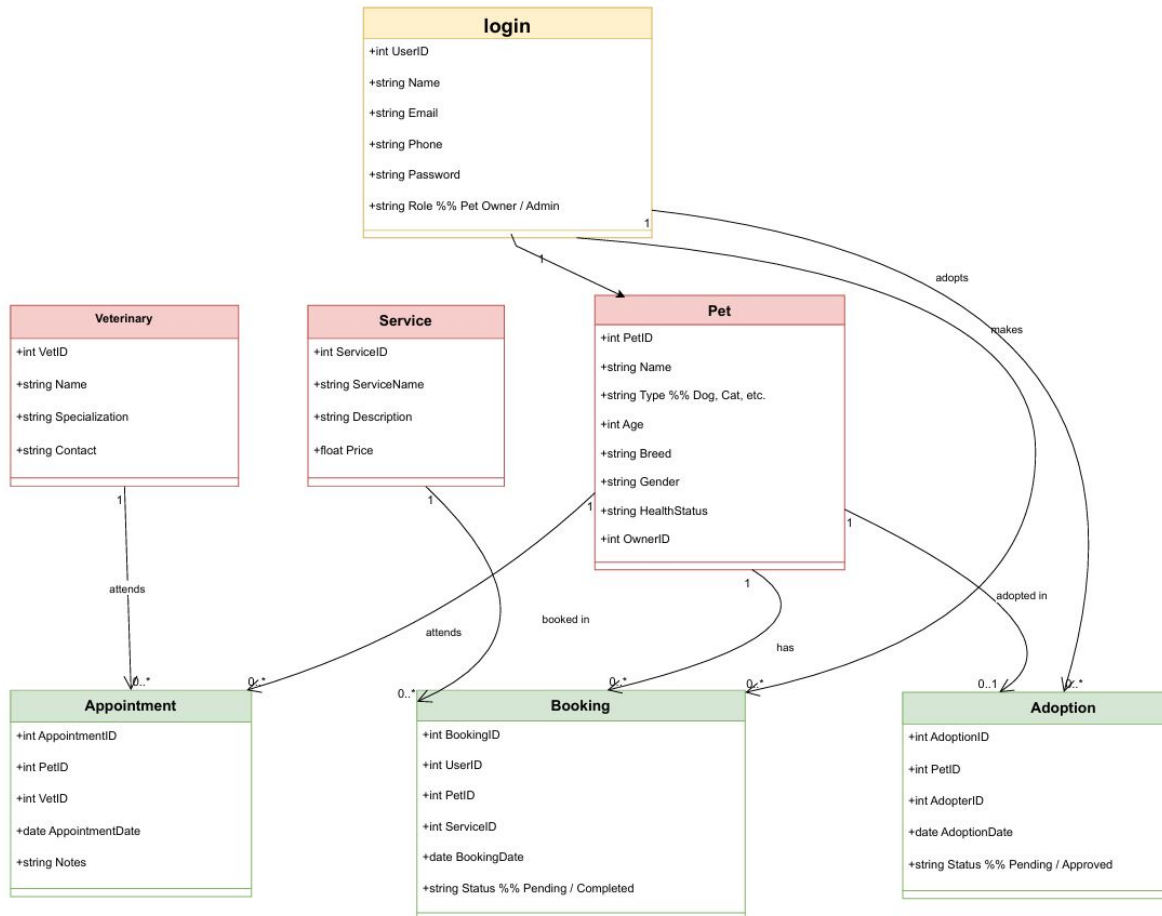


Figure 2.5: Entity Relationship Diagram

2.9 Database Schema

Database Tables:

- users(user_id, name, email, password, role)
- pets(pet_id, name, species, age, owner_id)
- doctors(doctor_id, name, specialization)
- appointments(appointment_id, pet_id, doctor_id, date, status)
- treatments(treatment_id, appointment_id, description, prescription)

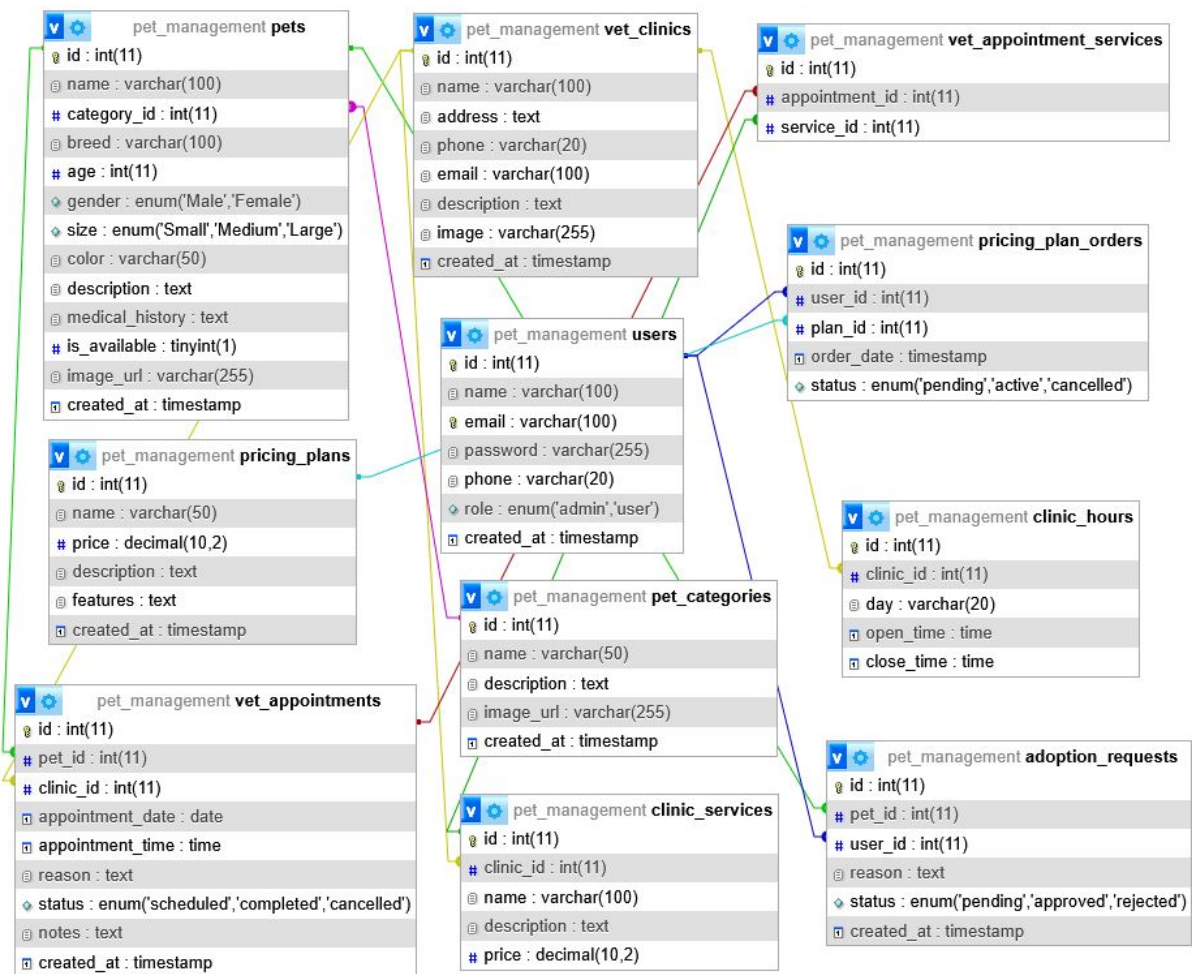


Figure 2.6: Table Relationship Diagram for Database Schema

2.10 Conclusion

This chapter presents a comprehensive overview of the requirement gathering process and the technical foundation necessary for the successful development of the Pet Management System. The aim of this chapter was to ensure that the project follows a clear, structured, and efficient development path aligned with user needs and technical feasibility.

Chapter 3

Interface Design and Implementation

This chapter provides a step-by-step description of the interface design and implementation of the PET Management System. The system was developed using HTML, CSS, Bootstrap, Laravel (PHP framework), and MySQL.

3.1 System Overview

The PET Management System was implemented as a web application with three user roles: Admin, Doctor, and User (Pet Owner). Each role has access to a specific dashboard with tailored features. The project follows the Model-View-Controller (MVC) design pattern.

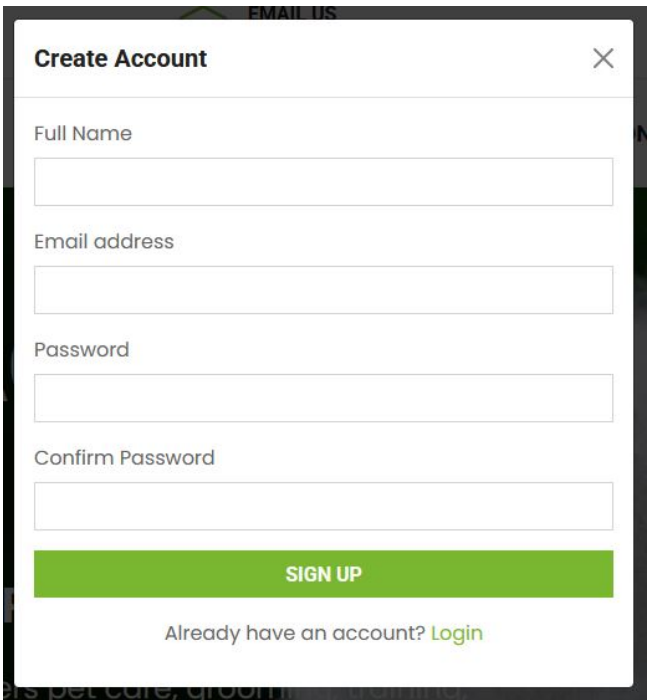
3.2 Homepage Interface



Figure 3.1: Home Page (Dashboard)

3.3 User Registration and Login

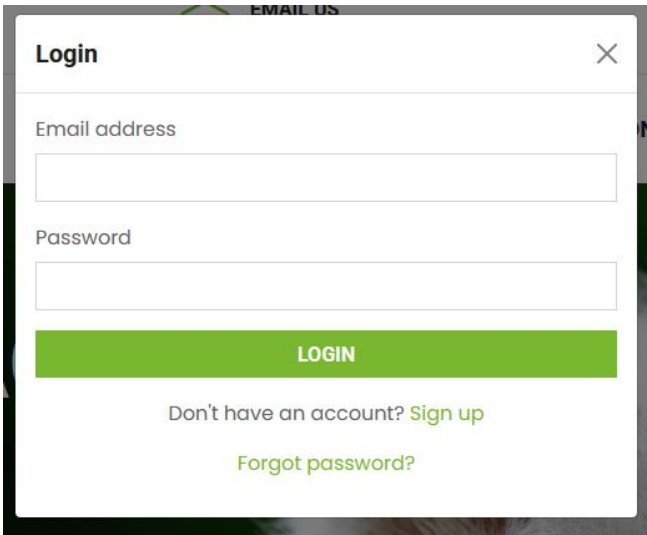
3.3.1 User Registration



The image shows a 'Create Account' modal form. It has a title bar with 'Create Account' and a close button. The form contains four input fields: 'Full Name', 'Email address', 'Password', and 'Confirm Password'. Below the fields is a green 'SIGN UP' button. At the bottom, there is a link that says 'Already have an account? Login'.

Figure 3.2: User Registration Page

3.3.2 User Login



The image shows a 'Login' modal form. It has a title bar with 'Login' and a close button. The form contains two input fields: 'Email address' and 'Password'. Below the fields is a green 'LOGIN' button. At the bottom, there are two links: 'Don't have an account? Sign up' and 'Forgot password?'.

Figure 3.3: Login Page

3.4 User Dashboard

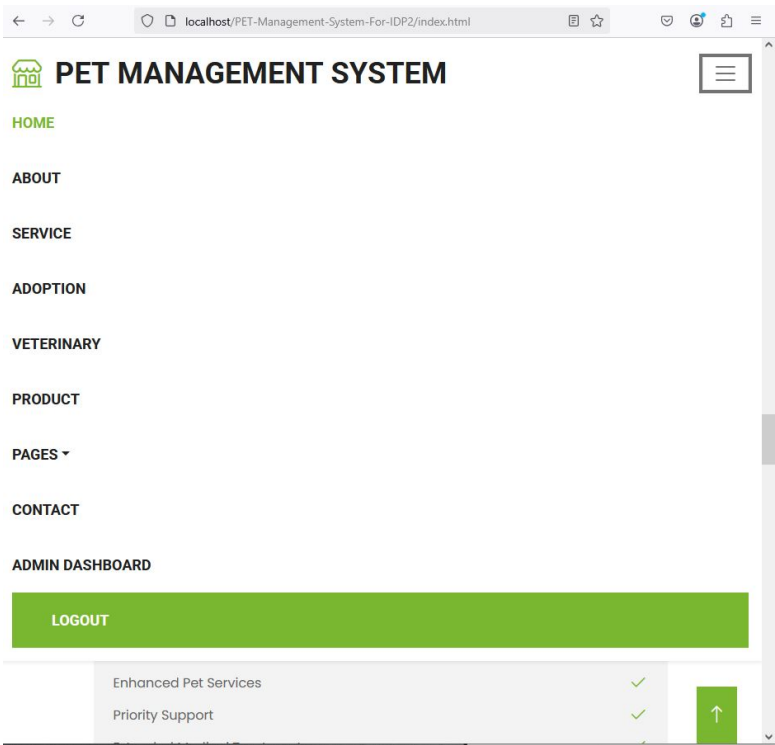


Figure 3.4: Dash-board for Pet Owners

3.5 Admin Dashboard

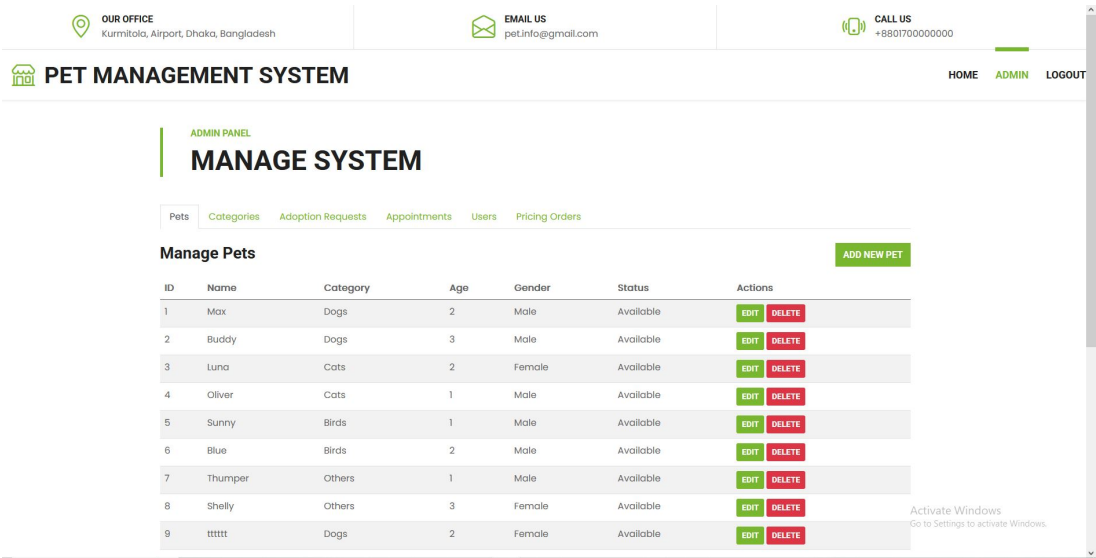


Figure 3.5: Admin Dashboard

3.6 Pet Profile Management

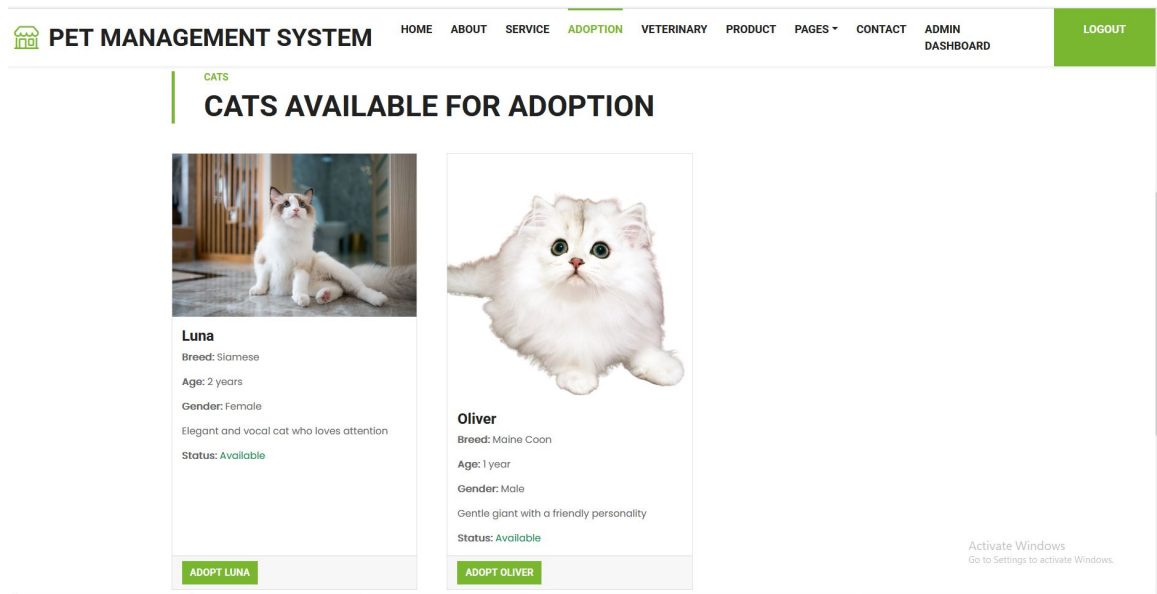


Figure 3.6: Pet Profile Management Interface

3.7 Doctor Dashboard and Treatment Interface

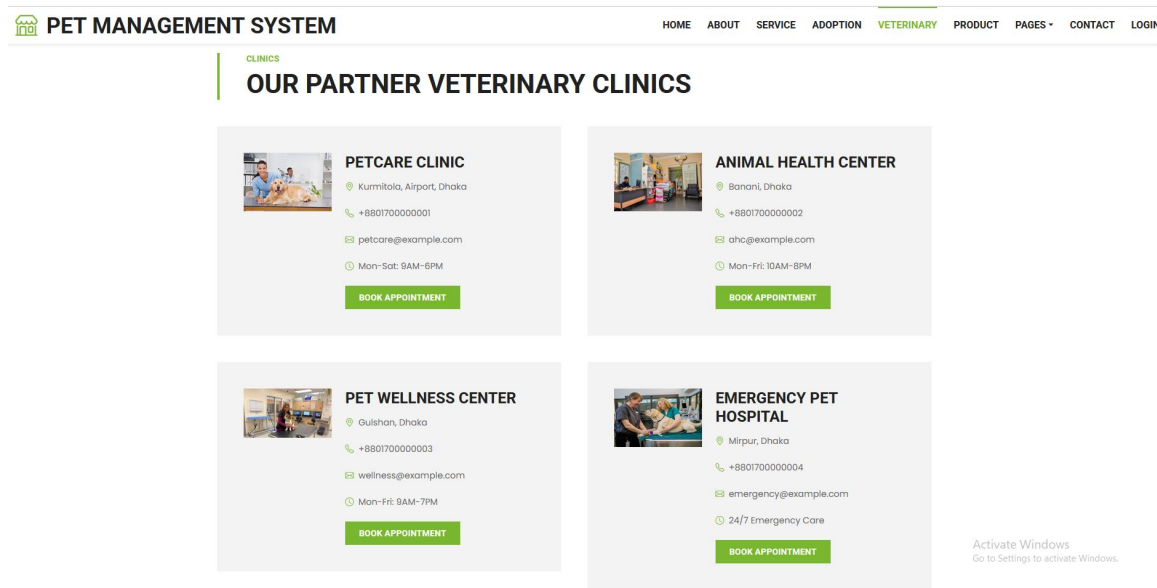


Figure 3.7: Doctor Dashboard

Doctors can:

- View upcoming appointments.
- Add treatment notes and prescriptions.
- View complete pet medical history.

3.8 Adoption

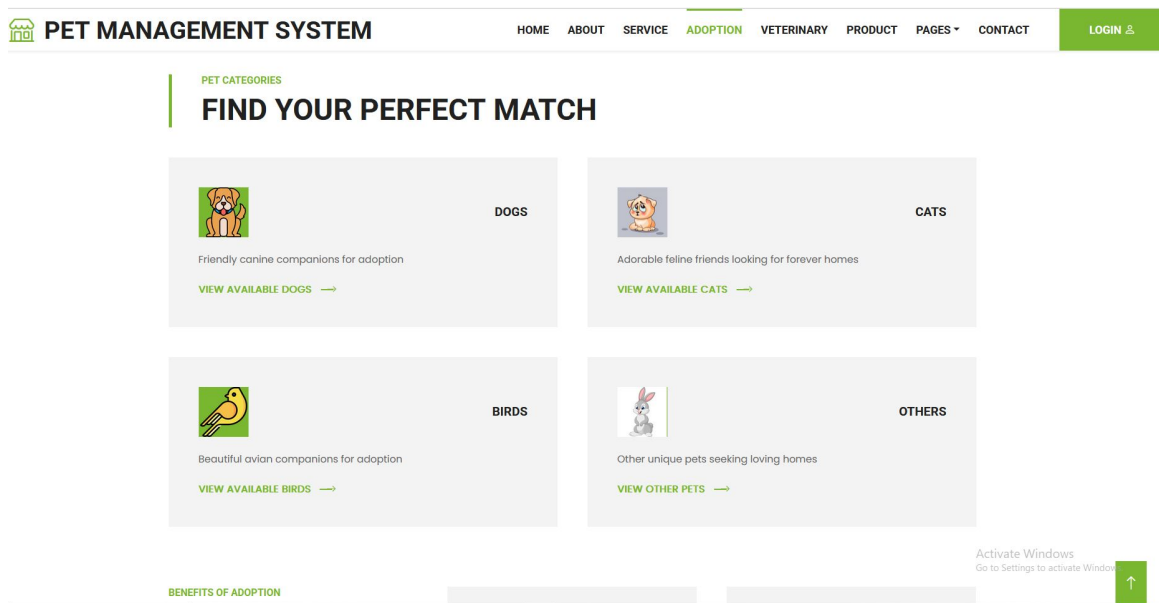


Figure 3.8: Pet Profile Management Interface

3.9 Adoption Booking

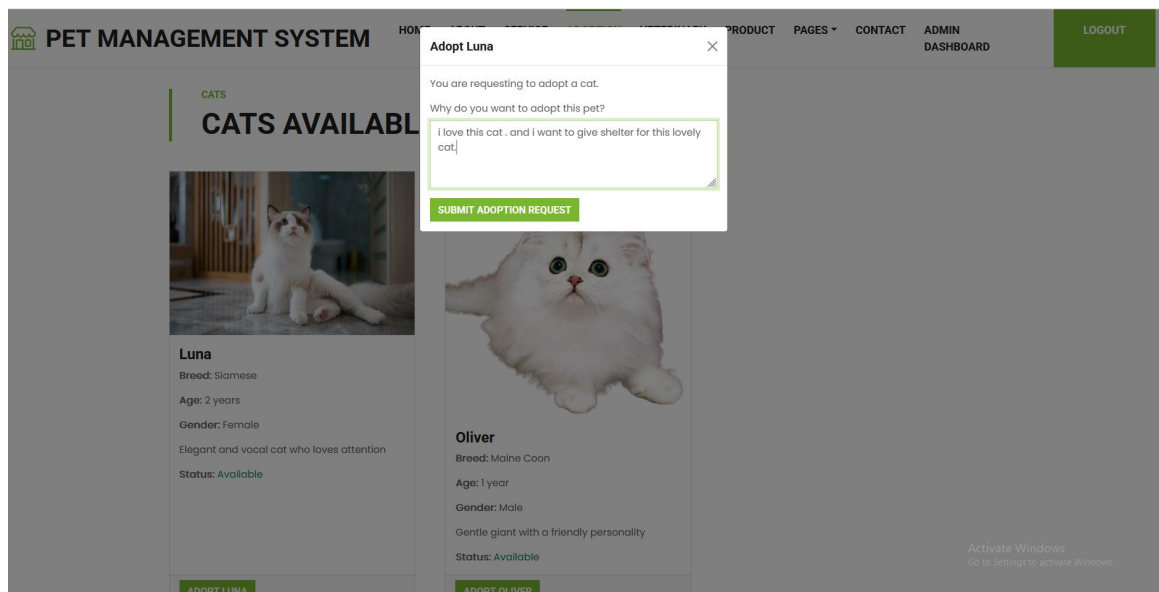


Figure 3.9: Appointment Booking Interface

3.10 Services

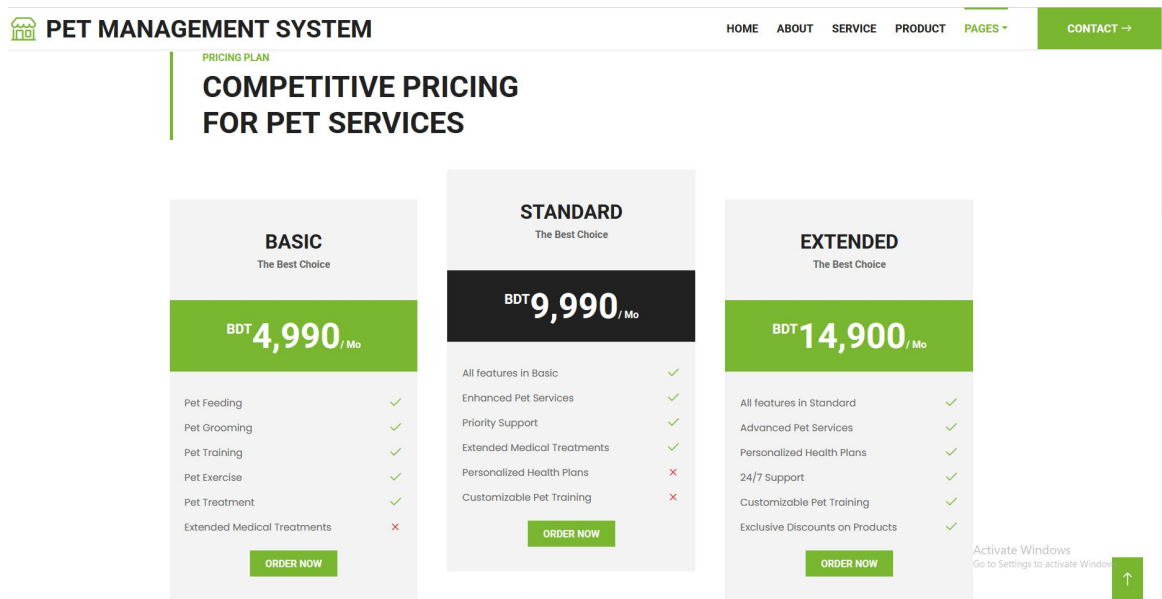


Figure 3.10: Pet Profile Management Interface

3.11 Products

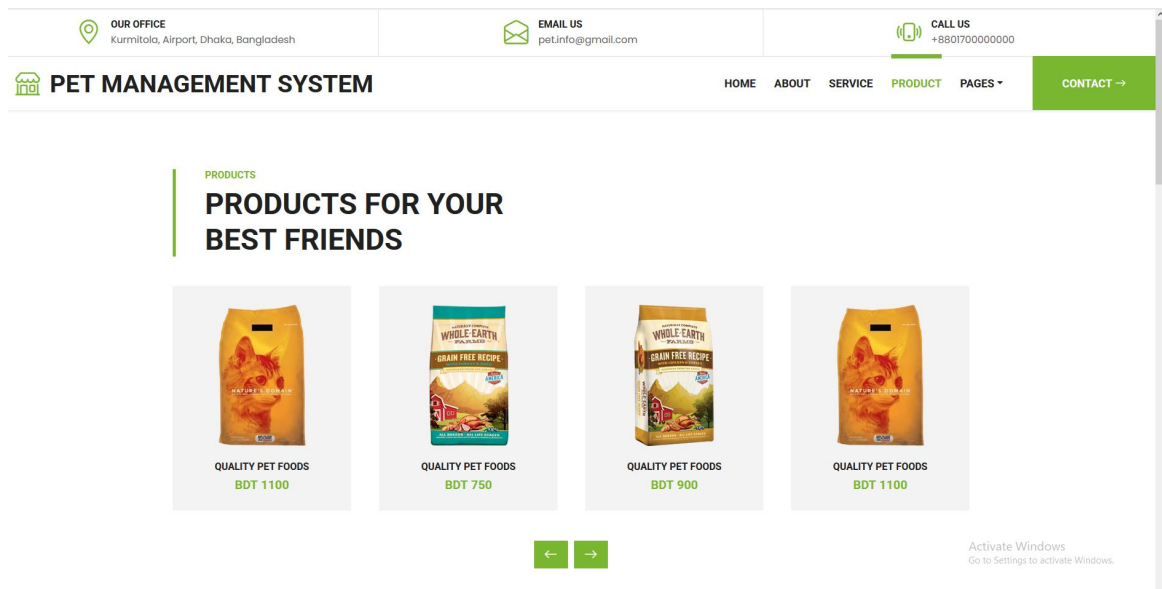


Figure 3.11: Appointment Booking Interface

3.12 Teams And Testimonials

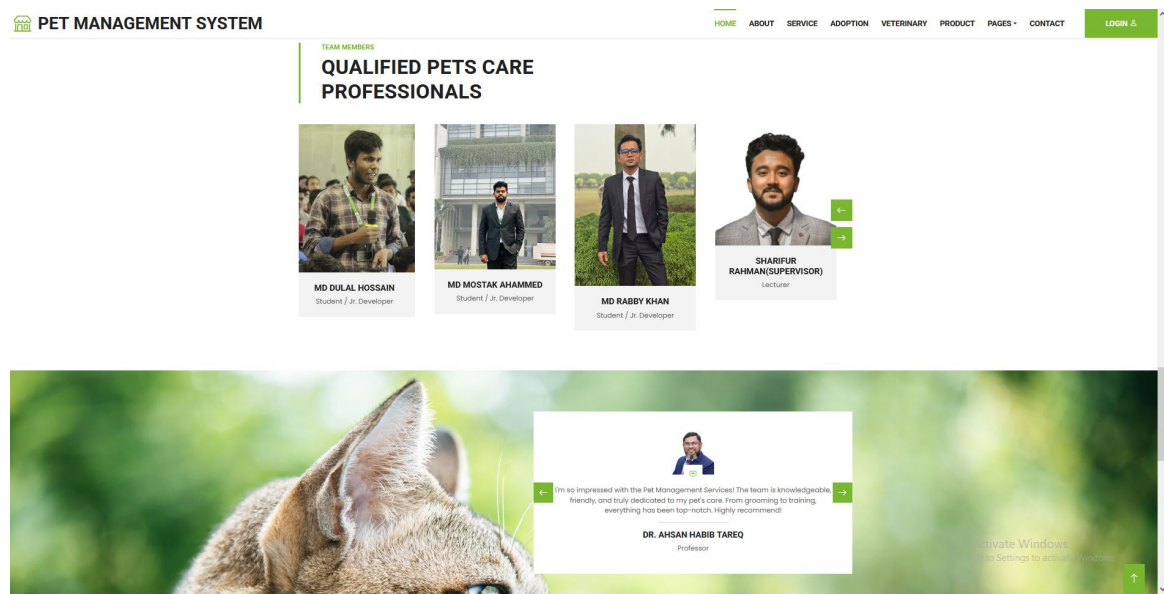


Figure 3.12: Pet Profile Management Interface

3.13 Databases Structure

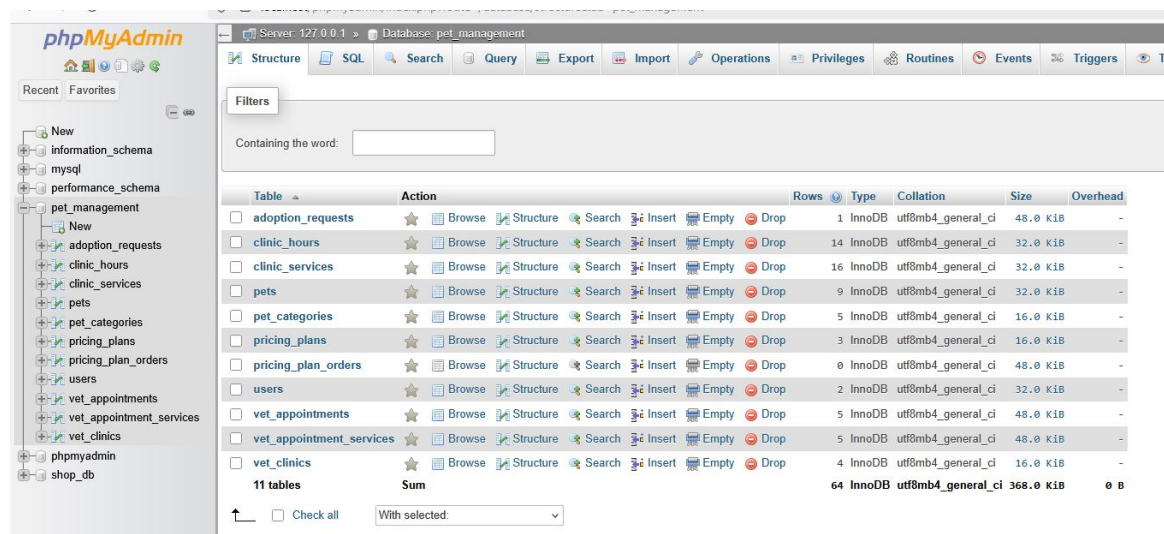


Figure 3.13: Appointment Booking Interface

3.14 Implementation Flow

1. Designed the database schema using MySQL.
2. Set up Laravel routes, controllers, and models.
3. Developed frontend interfaces using Blade templates and Bootstrap.
4. Applied authentication using Laravel's built-in Auth scaffolding.
5. Implemented CRUD operations using Laravel Eloquent ORM.
6. Conducted UI/UX testing and made enhancements.

3.15 Platform and Tools Used

- **Frontend:** HTML, CSS, JavaScript, and frameworks such as React for dynamic content.
- **Backend:** Node.js, Express.js for server-side logic.
- **Database:** MongoDB to store product details, user accounts, and order data.
- **Payment Gateway:** Integration with Stripe or PayPal for handling payments.
- **Hosting:** The application will be hosted on cloud platforms like AWS or Heroku.

3.16 System Architecture Diagram

The system architecture consists of the following components:

- **User Interface:** Interface for customers and admins to interact with the platform.
- **Application Layer:** Handles the business logic such as managing orders and inventory.
- **Database Layer:** Stores product details, user data, and order history.

3.17 Conclusion

This chapter outlined the visual and functional implementation of the Pet Management System, focusing on both the frontend interface and backend logic that drive the core functionalities of the platform. The implementation phase aimed to transform the previously designed models and requirements into a working software system that is user-friendly, reliable, and secure.

Overall, the implementation of the Pet Management System successfully integrated the visual components with functional logic to deliver a comprehensive platform. By combining a user-centered interface with a secure and efficient backend, the system ensures a reliable experience for all users while supporting essential operations like pet management, service booking, and adoption processes.

Chapter 4

Test Cases for PET Management System

This chapter presents the test cases applied to verify the correct and reliable functionality of the PET Management System. Each section describes the scenario, expected outcome, and actual results with supporting screenshots.

4.1 Testing Strategy

We adopted a combination of:

- **Unit Testing** – to test individual components (e.g., controller functions).
- **Integration Testing** – to test interactions between modules (e.g., pet profile and appointments).
- **System Testing** – to test the end-to-end flow of the application.
- **User Acceptance Testing (UAT)** – to ensure the system meets end-user expectations.

4.2 Sample Test Cases

4.2.1 Test Case: User Registration

- **Action:** Register as a new user with valid credentials.
- **Input (Test Case):** Name: "dulal", Email: "dulal@gmail.com", Password: "213902116"
- **Expected Output:** User is redirected to the user dashboard.
- **Actual Output:** Success. User redirected correctly.
- **Test Result:** Pass

4.3 Test Case Summary Table

Criteria	Action	Input (Test Case)	Expected Output	Actual Output	Test Result
User Registration	Weak Password	username: "test2", password: "123"	"Password too weak"	Success message	Fail
User Registration	Long Username	username: "a"*256, password: "securePass"	"Username too long"	Success message	Fail
Adoption	Valid Request	user_id: 1, pet_id: 3	"Adoption request submitted"	"Adoption request submitted"	Pass
Security	XSS Test on User Input	<script>alert("XSS")</script> in form	"Sanitized output"	"Sanitized output"	Pass
Performance	Load Pet Listings	5000 pet records	Load in < 3 seconds	Loaded (time not specified)	Fail
Edge Cases	Large Image Upload	10MB pet photo	"Image too large"	"Image too large"	Fail
Search Function	SQL Injection in Search	Search term: " OR 1=1 --	"Invalid input"	Success message	Fail

4.4 Test Result / Output

4.4.1 Register New User - Test Result: Pass

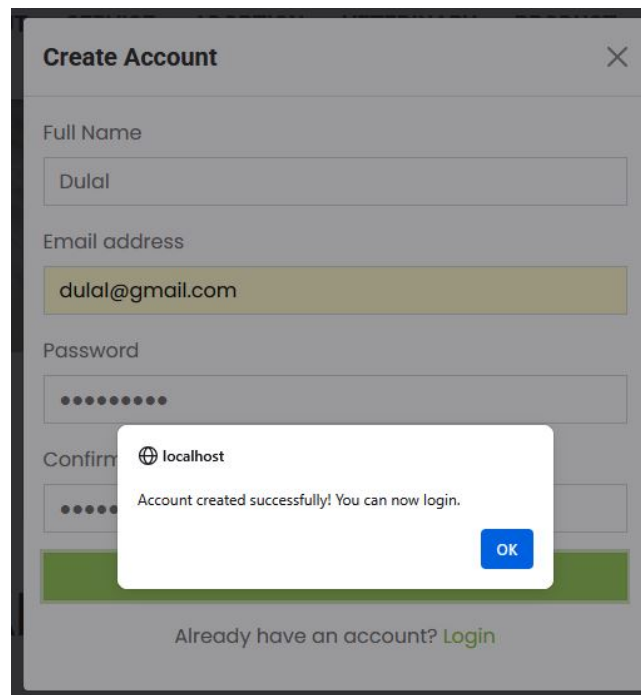


Figure 4.1: Register New User - Test Result: Pass.

4.4.2 Register Existing User - Test Result: Pass

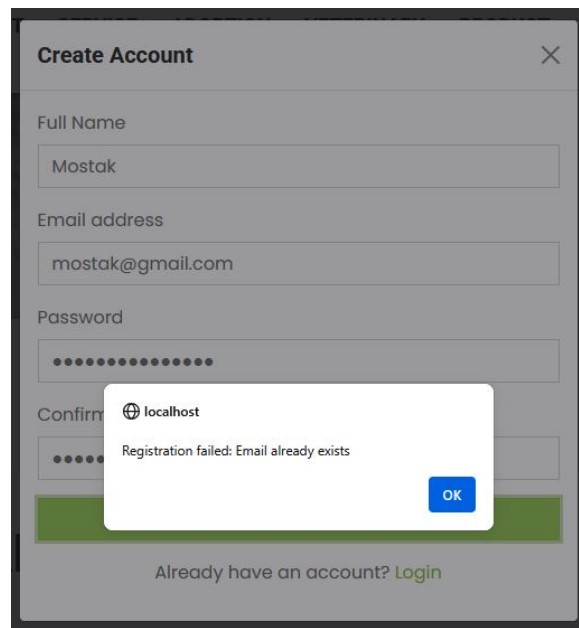


Figure 4.2: Register Existing User - Test Result: Pass.

4.4.3 Long Password - Test Result: Fail

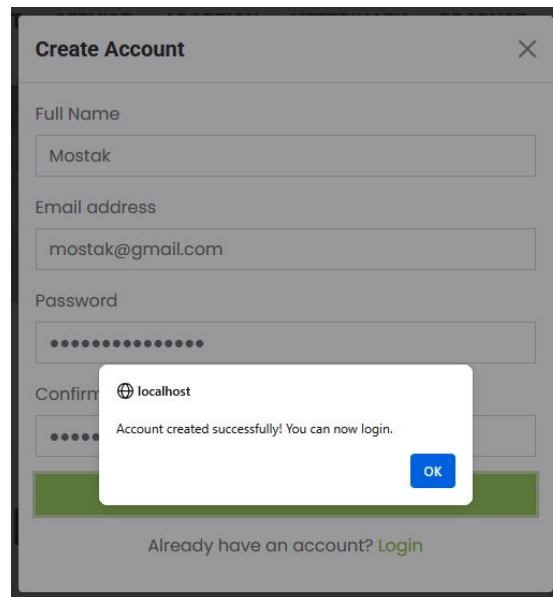


Figure 4.3: Long Password - Test Result: Fail.

4.4.4 Valid Login - Test Result: Pass

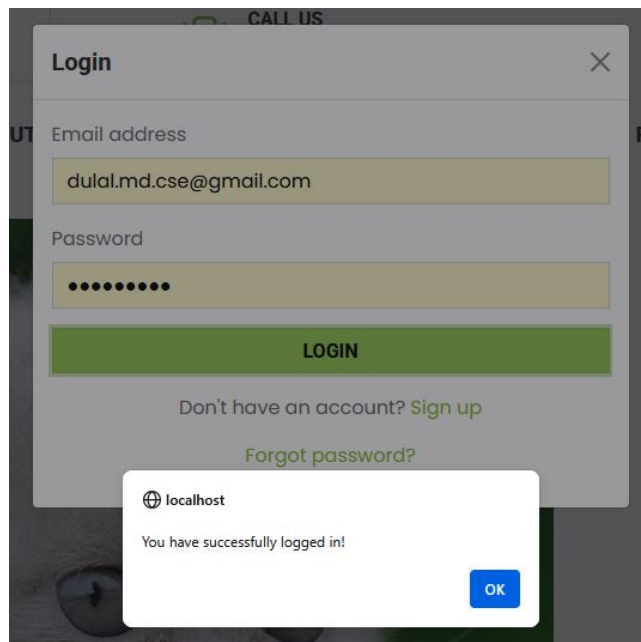


Figure 4.4: Valid Login - Test Result: Pass.

4.4.5 Invalid Login - Test Result: Pass

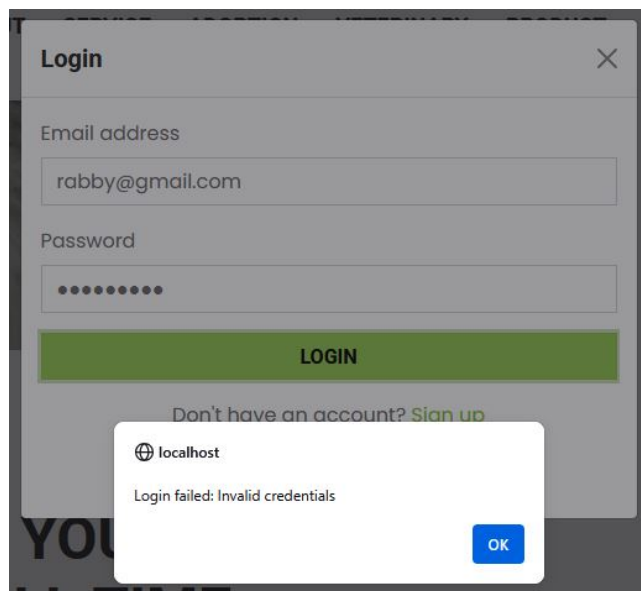
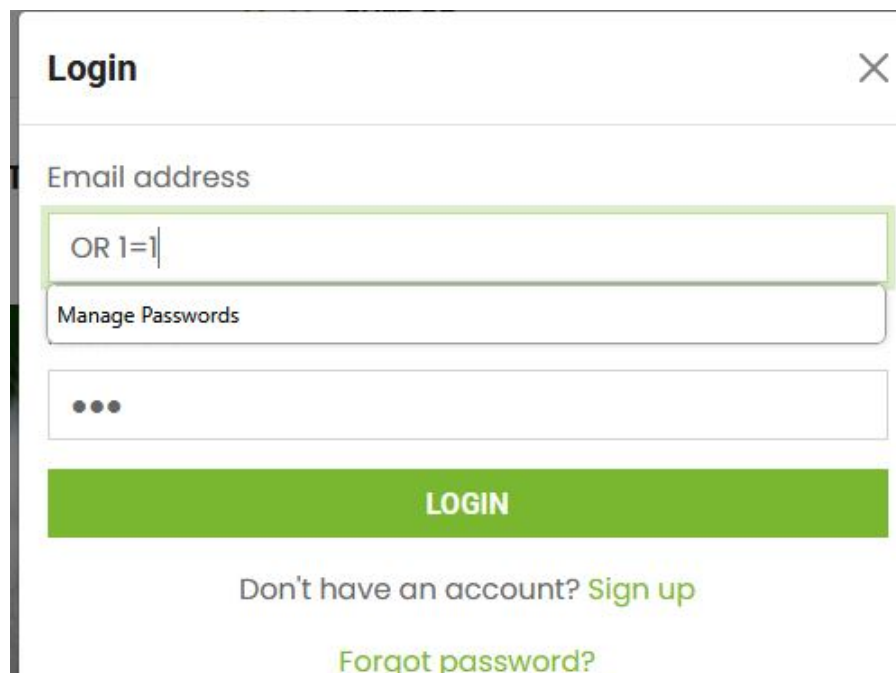


Figure 4.5: Invalid Login - Test Result: Pass.

4.4.6 SQL Injection in Filter - Test Result: Pass



The screenshot shows a login interface with the following elements:

- Login** header with a close button (X).
- Email address** label above a text input field.
- The text input field contains the payload `OR 1=1` and is highlighted with a green border.
- A **Manage Passwords** link below the email field.
- A password input field with three dots indicating it is masked.
- A green **LOGIN** button.
- Links for **Don't have an account? Sign up** and **Forgot password?** at the bottom.

Figure 4.6: SQL Injection in Filter - Test Result: Pass.

4.5 Analysis and Discussion

The comprehensive testing of the PET Management System revealed both strengths and areas for improvement in the application's functionality, security, and performance. The system successfully handled core operations like user registration, authentication, and data management, with most test cases passing as expected. Security measures proved effective against common threats, demonstrated by successful prevention of SQL injection attacks and unauthorized admin access. However, the testing uncovered critical vulnerabilities in input validation, particularly for password strength requirements and image upload size limits, where the system failed to enforce specified constraints. Performance testing showed the application could handle moderate loads but struggled with extreme cases like loading 5000 pet records within the required time frame. The manual testing approach provided valuable insights into user experience aspects that automated tests might miss, though incorporating automation for regression testing could improve efficiency. The results indicate that while the system meets basic functional requirements, enhancements are needed in input validation, error handling, and performance optimization to ensure reliability under real-world conditions. These improvements would significantly enhance the system's robustness and user satisfaction while maintaining its strong security foundation.

4.6 Conclusion

This chapter presents the outcomes of the testing phase of the Pet Management System, which was conducted to ensure that all functional components and user interactions perform reliably under various conditions. The testing process was designed to validate system behavior, verify role-based functionalities, and evaluate performance with respect to the predefined requirements.

Chapter 5

Conclusion and Future Work

5.1 Summary of the Project

The PET Management System was developed to streamline pet-related services for pet owners, veterinarians, and administrators. The system enables users to register, manage pet profiles, schedule appointments, and access medical records. It provides role-based dashboards and administrative controls to ensure secure and efficient operations.

Throughout the development, we followed systematic software engineering practices including requirement analysis, architectural and database design, interface development, and rigorous testing. The system was built using Laravel for backend, MySQL for database, and Blade templates for frontend.

Pet Management Systems are transforming pet care by offering convenient, comprehensive solutions for monitoring health, behavior, and daily routines. As these systems evolve, they are likely to become even more integrated, intelligent, and user-friendly, ultimately enhancing the quality of life for both pets and their owners. By leveraging advancements in technology such as AI and IoT, these platforms have the potential to provide real-time insights, proactive health management, and seamless connectivity with veterinary professionals.

5.2 Societal Impact

The development and implementation of the **Pet Management System** offer substantial societal benefits that extend beyond the immediate user base. The system not only streamlines pet care services but also contributes to public welfare, animal rights, and responsible pet ownership. The following are key areas where this system can positively impact society:

1. Improved Pet Healthcare

The system enables timely medical care by digitizing pet health records and simplifying appointment scheduling. It helps ensure regular vaccinations, routine check-ups, and quicker responses to emergencies, promoting overall well-being for pets.

2. Digital Record Keeping

By replacing manual and paper-based systems, the platform ensures efficient storage, retrieval, and long-term management of data. This not only supports data security and integrity but also reduces environmental impact through paperless operation.

3. **Efficient Appointment Management**

Automated scheduling and conflict resolution features significantly reduce waiting times for pet owners and streamline operations for veterinarians, making the system more efficient and user-friendly.

4. **Support for Responsible Pet Ownership**

The system empowers pet owners by offering access to medical records, vaccination schedules, and expert advice. It helps owners stay informed about their pet's needs and encourages consistent, responsible care.

5. **Adoption and Rescue Management**

Animal shelters and NGOs can use the platform to manage pet listings, adoption applications, and medical care, helping more animals find suitable homes while reducing administrative overhead.

6. **Scalability and Broader Use**

The system is adaptable and can be extended for use by:

- Veterinary clinics and hospitals
- Animal rescue organizations
- NGOs working in animal welfare
- Government bodies monitoring public health and animal registration

Overall, the Pet Management System contributes to a more humane, efficient, and technologically advanced approach to animal care and public engagement.

5.3 **Future Work**

While the **Pet Management System** successfully meets its core objectives—providing an efficient platform for pet care services, appointments, and adoption—there are several areas where future enhancements can further improve system functionality, scalability, and user satisfaction. The following outlines the potential directions for future development:

1. **Mobile Application Development**

To enhance accessibility and user convenience, a mobile version of the Pet Management System can be developed. A cross-platform mobile app (using *Flutter* or *React Native*) would allow users to:

- Manage bookings and appointments on the go.
- Receive push notifications for service reminders, vaccinations, or adoption updates.
- Use GPS to find nearby veterinary clinics or pet care centers.

2. **AI-Based Pet Health Monitoring**

Integrating machine learning or AI algorithms could allow the system to:

- Predict health issues based on pet symptoms or historical data.
- Recommend services or treatments based on breed, age, and medical history.
- Automatically schedule follow-up appointments based on diagnosis.

3. **Online Payment Gateway Integration**

Future versions of the system could support online transactions by integrating secure payment gateways (e.g., *Stripe*, *SSLCommerz*, or *PayPal*). This would enable:

- Online service booking and fee payments.
- Donations to pet shelters.
- Secure tracking of transaction history for users and administrators.

4. **Live Chat and Teleconsultation with Veterinarians**

Incorporating real-time communication features such as live chat or video consultations would be highly beneficial. This can:

- Provide emergency assistance to pet owners.
- Allow remote health assessments and advice from veterinarians.
- Improve accessibility in rural or remote areas.

5. **Pet Medical Record Integration**

An advanced version of the system could serve as a centralized platform to store complete medical histories for pets, allowing:

- Easy access for pet owners and veterinarians.
- Seamless migration of records across clinics.
- Integration with third-party pet insurance or medical providers.

Bibliography

References

- [1] J. Zinsstag, E. Schelling, B. Bonfoh *et al.*, “Managing the health of pets: A One Health approach,” 2011.
- [2] B. Smith and T. Jones, “The role of technology in pet care,” 2019.
- [3] K. Peterson and A. Williams, “User-centric design for mobile applications in pet management,” 2020.
- [4] American Veterinary Medical Association (AVMA), “Digital tools in veterinary practices,” 2022.
- [5] Statista, “Global pet care market trends and insights,” 2023.