Project Proposal

Database (MII212501)

PawPoint: Veterinary Clinic Appointment Scheduling and Management Application

Written By:

Deira Aisya Refani (24/532821/PA/22539)

Ratu Faiha Salsabilla (24/532756/PA/22533)

Aufaa Azzahra Aryawan (24/546854/PA/23215)

Universitas Gadjah Mada

Faculty Of Mathematics and Natural Science

Department of Computer Science and Electronics

2025

1. Introduction

1.1. Problem Background

Veterinary clinics provide medical care for all kinds of animals, from large livestock such as horses and cows to exotic pets like birds and reptiles, as well as aquatic animals including fish and amphibians. Due to this wide diversity of species and medical needs, an organised veterinary clinic appointment system is crucial for managing schedules, treatments, and medical histories efficiently. Moreover, a single client (pet owner) may own multiple animals with different health records, which makes manual record-keeping inefficient, time-consuming, and highly prone to human error.

Therefore, an organised computerised system that integrates pet, owner, veterinarian, and appointment data into a single database is vital for modern veterinary clinics. Such a system can improve data accuracy, consistency, and accessibility while reducing redundant data entry. In addition, it enables veterinarians to quickly access complete medical histories, leading to better diagnosis and treatment decisions.

Furthermore, a well-designed appointment management system can optimise scheduling, reduce waiting times, and prevent appointment conflicts. It also enhances communication between clinic staff, veterinarians, and pet owners through timely updates and reminders. From an administrative perspective, the system supports reporting, billing, and long-term data analysis, which can help clinics improve operational efficiency and service quality. Overall, a computerised veterinary clinic management system plays a critical role in delivering reliable, efficient, and high-quality animal healthcare services.

1.2. Objectives

The main objective of PawPoint is to develop a structured and efficient database solution that automates appointment scheduling and record management in a veterinary clinic. Our system aims to:

1. Store and manage information about pet owners, pets, veterinarians, and appointments.
2. Simplify the scheduling process and prevent overlapping appointments.
3. Record treatment or vaccination details after each appointment.
4. Allow easy retrieval of pet medical history and appointment records.
5. Provide search and reporting function to support administrative tasks and decision making.

1.3. Users

The PawPoint Veterinary Clinic Appointment System is made for different types of users who help run the clinic's daily activities. Each user has their own role and level of access to make the system organized and efficient. There are four main types of users: Admin – schedules appointments, updates records, and oversees clinic operations Veterinarian – views assigned appointments and records diagnoses, treatments, and

prescriptions
Pet Owner – views appointment history and manages their pets' information

1.4. Use Cases

The main use cases supported by the system include:

1. Registering new pet owners and their animals.
2. Editing and deleting owner or pet profiles
3. Scheduling and updating appointments between client and veterinarians.
4. Updating appointment status
5. Preventing overlapping appointments
6. Recording medical treatments and diagnoses for each pet visit.
7. Retrieving pet medical histories
8. Generating clinic reports (by date, vet, or pet)

2. Database Design

2.1. ERD

The Entity Relationship Diagram (ERD) illustrates the overall structure of the database and how the entities are related to one another. It represents the main components of the Veterinary Appointment System, including users, pet owners, pets, veterinarians, appointments, clinics, and treatment records. The ERD helps visualize the logical connections and cardinalities between entities, ensuring that the system supports efficient data management and integrity. The ERD is shown in Diagram 1.

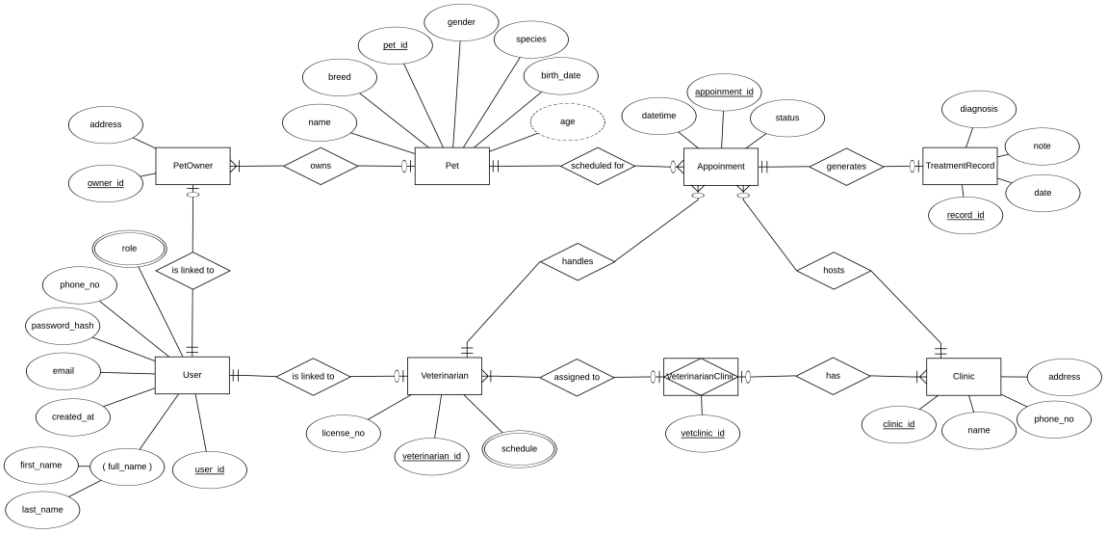


Diagram 1. Entity Relational Diagram (ERD) of PawPoint

2.2. Entity Explanation

Each entity represents a key component of the PawPoint system, and the attributes describe the data stored for each one. These entities work together to manage information about users, pets, veterinarians, and appointments within the database. The main entities and their attributes are:

1. User

   This entity holds the main account details of people who use the system. It helps identify each person and define their role.

   a. user_id (Primary Key)
   b. email
   c. password_hash
   d. role
   e. full_name (first_name, last_name)
   f. created_at
   g. phone_no

2. PetOwner

   This entity contains information about pet owners who use the system to manage their pets. It connects each owner to their user account for easier access.

   a. owner_id (Primary Key)
   b. address

3. Pet

   Holds data about each pet, including its name, type, and other information. Every pet is linked to one pet owner.

   a. pet_id (Primary Key)
   b. name
   c. species
   d. breed
   e. birth_date
   f. age (Derived)
   g. gender
   h. birth_date

4. Veterinarian

   Includes information about veterinarians such as their name, specialization, contact details, and schedule. Each one is linked to a user account.

   a. veterinarian_id (Primary Key)
   b. license_no
   c. schedule (Multi-valued)

5. Appointment

   Records all appointment details including the pet, veterinarian, and clinic involved, along with date, notes, and status.

   a. appointment_id (Primary Key)
   b. datetime
   c. status

6. TreatmentRecord

   Stores medical details from each appointment such as diagnosis, treatment, and prescription information.

   a. record_id (Primary Key)
   b. diagnosis
   c. note

d. date

7. Clinic

   Keeps information about the clinic like its name, address, and phone number. It also links to all related appointments.

   a. clinic_id (Primary Key)
   b. name
   c. address
   d. phone_no

8. VeterinarianClinic

   An associative entity that manages the relationship between veterinarias and the clinics where rhey are assigned to work.

   a. vetclinic_id (Primary Key)

2.3. Relationship Explanation

Each entity in the PawPoint system is interconnected through well-defined relationships that enable the database to represent real-world interactions within a veterinary clinic. These relationships ensure that data about users, pets, veterinarians, and appointments is consistently linked and accurately maintained. The relationships are:

1. User is linked to PetOwner (1:1) = Not all users are pet owners. However, every pet owner must be associated with exactly one user account.
2. User is linked to Veterinarian (1:1) = A user may or may not be a veterinarian, but every veterinarian must be registered as a user.
3. PetOwner owns Pet (1:N) = A pet owner may own zero or multiple pets, while every pet must be owned by exactly one pet owner.
4. Pet scheduled for Appointment (1:N) = A pet may have no appointments or multiple appointments, but every appointment must be associated with exactly one pet.
5. Veterinarian handles Appointment (1:N) = A veterinarian may handle zero or many appointments, while each appointment must be handled by exactly one veterinarian.
6. Veterinarian assigned to VeterinarianClinic (1:N) = A veterinarian may be assigned to zero or multiple clinics, and a clinic may have zero or multiple veterinarians.
7. VeterinarianClinic has Clinic (N:1) = A veterinarian may be assigned to zero or multiple clinics, and a clinic may have zero or multiple veterinarians.
8. Clinic hosts Appointment (1:N) = A veterinarian may handle zero or many appointments, while each appointment must be handled by exactly one veterinarian.
9. Appoinment generates TreatmentRecord (1:1) = An appointment may result in at most one treatment record (for example, cancelled appointments), while every treatment record must be generated from exactly one appointment.

2.4. Normalization Steps

2.4.1. User

The initial user table contained repeating role information for users with multiple roles, which violated the normalisation rules by introducing redundancy.

1NF: All attributes were atomic, but the role attribute contained repeating values for a single user, Finanazwa Ayesha, who has the role of both a pet_owner and being a veterinarian.

2NF: Since the primary key in this table is user_id, all non-key attributes were fully functionally dependent on it.

3NF: To remove redundancy and transitive dependency, the 'role' attribute was separated into its own table, role, and a junction table 'user_role' was created to handle the many-to-many relationship between users and roles.

2.4.2. PetOwner

The PetOwner table stores address information for users who own pets.

1NF: All attributes are atomic

2NF: The primary key 'owner_id' uniquely identifies each record, and all attributes are fully dependent on it.

3NF: There are no transitive dependencies, and the table is already in 3rd normal form.

2.4.3. Veterinarian

At first, the veterinarian's schedules were stored as multivalued attributes, which violated the 1st normal form.

1NF: The multivalued attribute, schedule, was removed and decomposed into individual schedule entries.

2NF: All non-key attributes depend fully on the primary key, veterinarian_id.

3NF: The veterinarian schedule was separated into a new table to remove any transitive dependencies.

2.4.4. Pet

The pet table contains descriptive attributes related to animals owned by the pet owners.

1NF: All attributes are atomic

2NF: The primary key, pet_id, uniquely determines all the other attributes in this table

3NF: The attribute 'age' is a derived attribute calculated from 'birth_date' and was removed to prevent redundancy.

2.4.5. Appointment

The appointment table represents the scheduled visits between pets and veterinarians at the clinics

1NF: All attributes contain atomic values

2NF: The primary key, appointment_id, uniquely determines all the other attributes in this table

3NF: There are no transitive dependencies, and the foreign keys are used to link related entities without any duplication.

2.4.6. VeterinarianClinic

The relationship between veterinarians and clinics is many-to-many.

1NF: Each record represents a single veterinarian-clinic association; it is atomic

2NF: The composite primary key (veterinarian_id, clinic_id) ensures full functional dependency.

3NF: No non-key attributes exist; therefore, the table is already in 3rd normal form

2.4.7. TreatmentRecord

The TreatmentRecord table stores medical records that are generated after the pets' appointments.

1NF: All attributes are atomic

2NF: The primary key, record_id, uniquely identifies each treatment record

3NF: All attributes depend only on the primary key, and no transitive dependencies exist

2.4.8. Clinic

The clinic table stores all the information about the veterinary clinics within the system

1NF: All attributes are atomic

2NF: The primary key, clinic_id, uniquely determines all the other attributes in this table

3NF: There are no transitive dependencies, and the table satisfies 3rd normal form

User Table

1NF, 2NF

user_id	first_name	last_name	created_at	email	phone_no	password_hash	role
100	Finanazwa	Ayesha	12/24/2025 0:00:00	finanazwak123@gmail.com	6212345678	v3uewjb2eidk	pet_owner
100	Finanazwa	Ayesha	12/24/2025 0:00:00	finanazwak123@gmail.com	6212345678	v3uewjb2eidk	veterinarian
102	Regina	Joan	12/25/2025 0:00:00	joanisme@gmail.com	6234518290	wgjbedjqmwd	pet_owner
103	Yohana	Amelia	12/26/2025 0:00:00	admin123@gmail.com	6235640831	ysuwnwmis2	admin
102	Regina	Joan	12/25/2025 0:00:00	joanisme@gmail.com	6234518290	wgjbedjqmwd	veterinarian

Functional dependencies: user_id -> {first_name, last_name, created_at, email, phone_no, password_hash}

3NF

user_id	first_name	last_name	created_at	email	phone_no	password_hash
100	Finanazwa	Ayesha	12/24/2025 0:00:00	finanazwak123@gmail.com	6212345678	v3uewjb2eidk
102	Regina	Joan	12/25/2025 0:00:00	joanisme@gmail.com	6234518290	wgjbedjqmwd
103	Yohana	Amelia	12/26/2025 0:00:00	admin123@gmail.com	6235640831	ysuwnwmis2

role_id	role
1	pet_owner
2	veterinarian
3	admin

user_id	role_id
100	1
100	2
102	1
103	3
102	2

Veterinarian table

UNF

veterinarian_id	license_no	schedule
1	AD3829810U	Sunday 18:00-20:00, Friday 08:00-10:00, Monday 19:00-21:00
2	AH192873198	Wednesday 10:00-12:00, Thursday 08:00-12:00
3	AB193802989	Monday 12:00-15:00, Tuesday 19:00-22:00
4	AS192733193	Saturday 16:00-20:00, Sunday 08:00-12:00

1NF

veterinarian_id	license_no	day	time_start	time_end
1	AD3829810U	Sunday	18:00	20:00
1	AD3829810U	Friday	8:00	10:00
1	AD3829810U	Monday	19:00	21:00
2	AH192873198	Wednesday	10:00	12:00

2	AH192873199	Thursday	8:00	13:00
3	AB193802989	Monday	12:00	15:00
3	AB193802989	Tuesday	19:00	22:00
4	AS192733193	Saturday	16:00	20:00
4	AS192733193	Sunday	8:00	12:00

Functional dependencies: veterinarian_id -> {license_no}, veterinarian_id, schedule_day -> {schedule_time}

2NF, 3NF

veterinarian_id	license_no
1	AD3829810U
2	AH192873198
3	AB193802989
4	AS192733193

VeterinarianClinic table

1NF, 2NF, 3NF

veterinarian_id	clinic_id
1	1
1	2
2	1
3	3
4	2

Functional dependencies: (veterinarian_id, clinic_id) -> {}

PetOwner table

1NF, 2NF, 3NF

owner_id	address
1	Jl. ajsgkahs A
2	Jl. ajsgkahs B
3	Jl. ajsgkahs C
4	Jl. ajsgkahs D
5	Jl. ajsgkahs E

Functional dependencies: owner_id -> {address}

Pet table

1NF, 2NF, 3NF

pet_id	name	breed	gender	species	birth_date
1	Alam	Persian	Unknown	Cat	6/15/2023
2	Bari	Bulldog	Male	Dog	6/27/2024
3	Cika	Parrot	Female	Bird	6/17/2025

Functional dependencies: pet_id -> {name, breed, gender, species, birth_date}

TreatmentRecord table

1NF, 2NF, 3NF

record_id	date	diagnosis	note	appointment_id
1	1/10/2025	Skin infection	Prescribed ointment	1
2	1/11/2025	fever	Antibiotics given	2

Functional dependencies: record_id -> {date, diagnosis, note, appointment_id}

Appointment table

1NF, 2NF, 3NF

appointment_id	datetime	status	pet_id	clinic_id	veterinarian_id
1	1/10/2025 10:00:00	Scheduled	1	2	1
2	1/11/2025 14:00:00	Completed	2	1	3
3	1/12/2025 09:00:00	cancelled	3	3	2

Functional dependencies: appointment_id -> {datetime, status, pet_id, clinic_id, veterinarian_id}

Clinic table

1NF, 2NF, 3NF

clinic_id	name	phone_no	address
1	Pet Clinic A	634262892	Jl. Blabla A
2	Pet Clinic B	634262893	Jl. Blabla B
3	Pet Clinic C	634262894	Jl. Blabla C

4	Pet Clinic D	634262895	Jl. Blabla D
5	Pet Clinic E	634262896	Jl. Blabla E

Functional dependencies: clinic_id -> {name, phone_no, address}

3. Database Implementation

3.1. Relational Schema

The database is implemented using a relational model derived from the ERD above in diagram 1 (chapter 2.1). Each entity and relationship has been mapped into tables that satisfy 3rd Normal Form (3NF) to reduce redundancy and improve data integrity. The main tables in our database are: user, role, user_role, pet_owner, pet, veterinarian, veterinarian_schedule, veterinarian_clinic, clinic, appointment, and treatment_record.

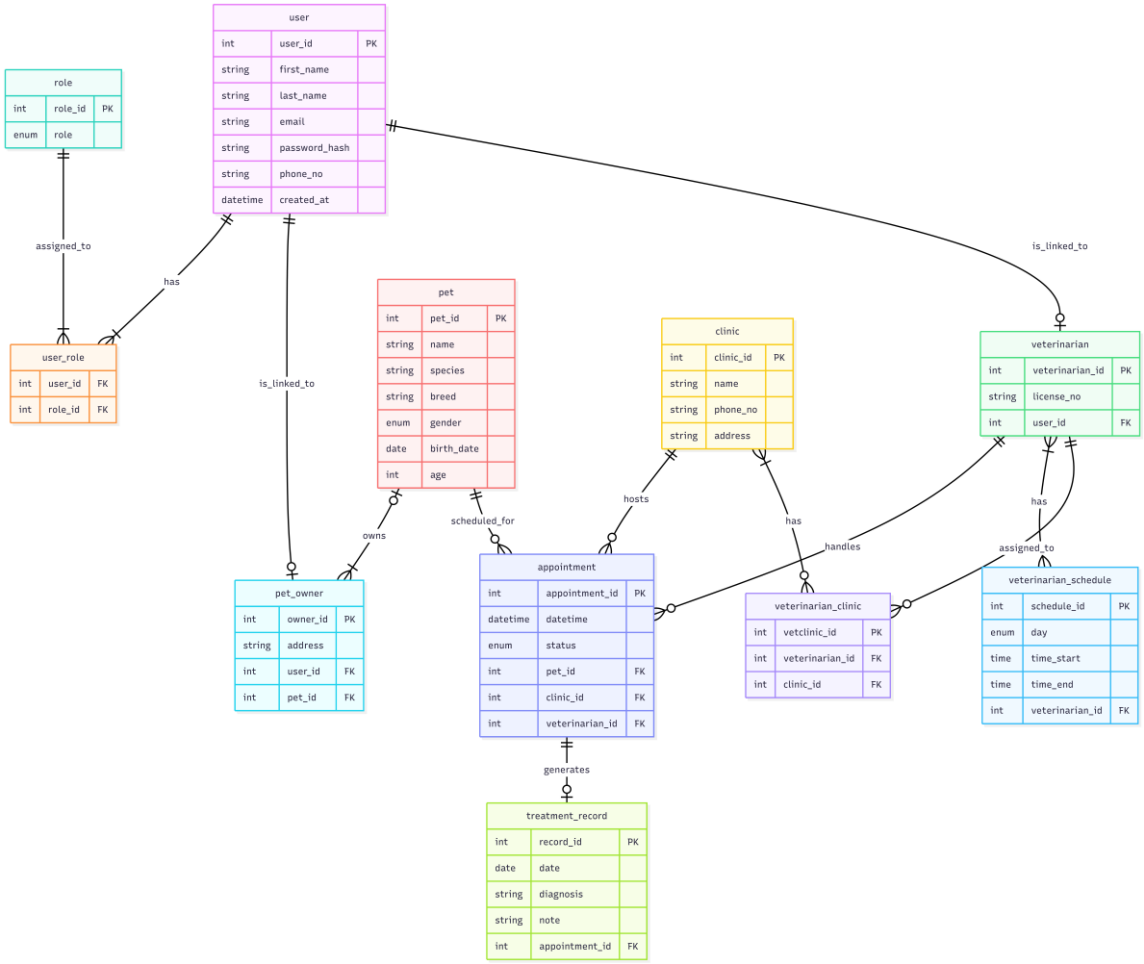


Diagram 2. Database Relational Scheme

3.2. Key Constraints

3.2.1. Primary Keys (PK)

Table	Primary Key
user	user_id
role	role_id
pet_owner	owner_id
pet	pet_id
veterinarian	veterinarian_id
veterinarian_schedule	schedule_id

clinic	clinic_id
appointment	appointment_id
treatment_record	record_id
veterinarian_clinic	vetclinic_id

3.2.2. Foreign Keys (FK)

Table	Foreign Key	References
user_role	user_id	user(user_id)
user_role	role_id	role(role_id)
pet_owner	user_id	user(user_id)
pet_owner	pet_id	pet(pet_id)
veterinarian	user_id	user(user_id)
veterinarian_schedule	veterinarian_id	veterinarian(veterinarian_id)
appointment	pet_id	pet(pet_id)
appointment	clinic_id	clinic(clinic_id)
appointment	veterinarian_id	veterinarian(veterinarian_id)
treatment_record	appointment_id	appointment(appointment_id)
veterinarian_clinic	veterinarian_id	veterinarian(veterinarian_id)
veterinarian_clinic	clinic_id	clinic(clinic_id)

3.3. Example SQL Statements

3.3.1. Create user Table

```
CREATE TABLE user (
    user_id INT AUTO_INCREMENT PRIMARY KEY,
    first_name VARCHAR(100) NOT NULL,
    last_name VARCHAR(100) NOT NULL,
    email VARCHAR(150) NOT NULL UNIQUE,
    password_hash VARCHAR(255) NOT NULL,
    phone_no VARCHAR(20),
    created_at DATETIME DEFAULT CURRENT_TIMESTAMP
);
```

3.3.2. Create role Table

```
CREATE TABLE role (
    role_id INT AUTO_INCREMENT PRIMARY KEY,
    role ENUM('pet_owner', 'veterinarian', 'admin') NOT NULL
);
```

3.3.3. Create user_role Table

```
CREATE TABLE user_role (
    user_id INT NOT NULL,
    role_id INT NOT NULL,
    PRIMARY KEY (user_id, role_id),
    FOREIGN KEY (user_id) REFERENCES user(user_id) ON DELETE CASCADE,
```

```
    FOREIGN KEY (role_id) REFERENCES role(role_id) ON DELETE CASCADE
);
```

3.3.4. Create clinic Table

```
CREATE TABLE clinic (
    clinic_id INT AUTO_INCREMENT PRIMARY KEY,
    name VARCHAR(150) NOT NULL,
    phone_no VARCHAR(20),
    address TEXT NOT NULL
);
```

3.3.5. Create veterinarian Table

```
CREATE TABLE veterinarian (
    veterinarian_id INT AUTO_INCREMENT PRIMARY KEY,
    license_no VARCHAR(100) NOT NULL UNIQUE,
    user_id INT UNIQUE,
    FOREIGN KEY (user_id) REFERENCES user(user_id) ON DELETE SET NULL
);
```

3.3.6. Create veterinarian_clinic Table

```
CREATE TABLE veterinarian_clinic (
    vetclinic_id INT AUTO_INCREMENT PRIMARY KEY,
    veterinarian_id INT NOT NULL,
    clinic_id INT NOT NULL,
    UNIQUE KEY uq_vet_clinic (veterinarian_id, clinic_id),
    FOREIGN KEY (veterinarian_id) REFERENCES veterinarian(veterinarian_id) ON
DELETE CASCADE,
    FOREIGN KEY (clinic_id) REFERENCES clinic(clinic_id) ON DELETE CASCADE
);
```

3.3.7. Create pet Table

```
CREATE TABLE pet (
    pet_id INT AUTO_INCREMENT PRIMARY KEY,
    name VARCHAR(100) NOT NULL,
    species VARCHAR(50) NOT NULL,
    breed VARCHAR(50),
    gender ENUM('male', 'female', 'unknown') DEFAULT 'unknown',
    birth_date DATE,
    age INT
);
```

3.3.8. Create pet_owner Table

```
CREATE TABLE pet_owner (
    owner_id INT AUTO_INCREMENT PRIMARY KEY,
    address TEXT,
    user_id INT NOT NULL,
    pet_id INT NOT NULL,
    FOREIGN KEY (user_id) REFERENCES user(user_id) ON DELETE CASCADE,
```

```
    FOREIGN KEY (pet_id) REFERENCES pet(pet_id) ON DELETE CASCADE
);
```

3.3.9. Create appoinment Table

```
CREATE TABLE appointment (
    appointment_id INT AUTO_INCREMENT PRIMARY KEY,
    datetime DATETIME NOT NULL,
    status ENUM('scheduled', 'completed', 'cancelled') DEFAULT 'scheduled',
    pet_id INT NOT NULL,
    clinic_id INT NOT NULL,
    veterinarian_id INT NOT NULL,
    FOREIGN KEY (pet_id) REFERENCES pet(pet_id) ON DELETE CASCADE,
    FOREIGN KEY (clinic_id) REFERENCES clinic(clinic_id) ON DELETE CASCADE,
    FOREIGN KEY (veterinarian_id) REFERENCES veterinarian(veterinarian_id) ON
DELETE CASCADE
);
```

3.3.10. Create treatment_record Table

```
CREATE TABLE treatment_record (
    record_id INT AUTO_INCREMENT PRIMARY KEY,
    date DATE,
    diagnosis TEXT,
    note TEXT,
    appointment_id INT UNIQUE,
    FOREIGN KEY (appointment_id) REFERENCES appointment(appointment_id) ON
DELETE SET NULL
);
```

3.3.11. Create veterinarian_schedule Table

```
CREATE TABLE veterinarian_schedule (
    schedule_id INT AUTO_INCREMENT PRIMARY KEY,
    day ENUM('monday','tuesday','wednesday','thursday','friday','saturday','sunday') NOT
NULL,
    time_start TIME NOT NULL,
    time_end TIME NOT NULL,
    veterinarian_id INT NOT NULL,
    UNIQUE KEY uq_vet_day (veterinarian_id, day),
    FOREIGN KEY (veterinarian_id) REFERENCES veterinarian(veterinarian_id) ON
DELETE CASCADE
);
```

3.3.12. Create trg_appointment_clinic_match_insert Trigger

```
DELIMITER //
CREATE TRIGGER trg_appointment_clinic_match_insert
BEFORE INSERT ON appointment
FOR EACH ROW
BEGIN
IF NOT EXISTS (
SELECT 1 FROM veterinarian_clinic vc
```

```
WHERE vc.veterinarian_id = NEW.veterinarian_id
AND vc.clinic_id = NEW.clinic_id
) THEN
SIGNAL SQLSTATE '45000' SET MESSAGE_TEXT = 'Veterinarian must
be assigned to the selected clinic';
END IF;
END //

CREATE TRIGGER trg_appointment_clinic_match_update
BEFORE UPDATE ON appointment
FOR EACH ROW
BEGIN
IF NOT EXISTS (
SELECT 1 FROM veterinarian_clinic vc
WHERE vc.veterinarian_id = NEW.veterinarian_id
AND vc.clinic_id = NEW.clinic_id
) THEN
SIGNAL SQLSTATE '45000' SET MESSAGE_TEXT = 'Veterinarian must
be assigned to the selected clinic';
END IF;
END //
DELIMITER ;
```

4. Application Implementation

4.1. Technology Used

This system is developed using a modern web technology stack that supports scalability, maintainability, and secure data processing. The technologies are categorized into frontend, backend, database, and development tools. The website has been deployed and can be accessed through https://db-project-group10-paw-point.vercel.app/.

4.1.1. Frontend

The frontend of the application is built using React 19.2.0, which enables the development of dynamic and component-based user interfaces. Vite 7.2.4 is used as the build tool to provide fast development startup and optimized production builds. For styling, Tailwind CSS 4.1.18 is utilized to enable utility-first and responsive design.

Client-side routing is handled by React Router DOM 7.11.0, allowing seamless navigation without full page reloads. Communication with the backend API is performed using Axios 1.13.2, which simplifies HTTP request handling. Authentication state is managed using the React Context API, ensuring centralized and consistent user authentication across the application. The frontend application is deployed using Vercel.

4.1.2. Backend

The frontend of the application is built using React 19.2.0, which enables the development of dynamic and component-based user interfaces. Vite

7.2.4 is used as the build tool to provide fast development startup and optimized production builds. For styling, Tailwind CSS 4.1.18 is utilized to enable utility-first and responsive design.

Client-side routing is handled by React Router DOM 7.11.0, allowing seamless navigation without full page reloads. Communication with the backend API is performed using Axios 1.13.2, which simplifies HTTP request handling. Authentication state is managed using the React Context API, ensuring centralized and consistent user authentication across the application. The frontend application is deployed using Vercel.

4.1.3. Database

The system uses PostgreSQL version 15 or later as the primary relational database management system. The database is hosted on Supabase, which provides a managed cloud-based PostgreSQL service. Database connections are managed using psycopg2 SimpleConnectionPool with a maximum of two connections, in accordance with the free-tier limitations.

The database implementation includes secure SSL/TLS encryption for all connections, efficient connection pooling to prevent connection exhaustion, and automatic cleanup and reuse of database connections to ensure optimal performance.

4.2. Features

4.2.1. Authentication and Authorization

The system implements a secure authentication mechanism using JSON Web Tokens (JWT). Users are required to register and log in before accessing protected resources. Passwords are securely stored using hashing techniques, and role-based access control ensures that each user can only perform actions permitted by their role. This approach protects sensitive data and prevents unauthorized access.

4.2.2. Pet Management

Pet owners are able to register and manage detailed pet profiles, including species, breed, age, and medical background. Each pet is linked to its owner through a relational mapping, ensuring accurate ownership tracking. Administrators have full visibility of all registered pets, while pet owners can only access data related to their own pets.

4.2.3. Clinic Management

The system provides centralized management of veterinary clinics, including clinic names, contact information, and addresses. Clinic data is publicly accessible to support appointment booking. Administrative users can create, update, and manage clinic records, enabling structured organization of veterinary services across multiple locations.

4.2.4. Veterinarian Management and Scheduling

Veterinarians are registered with verified license numbers and can be assigned to one or more clinics. The system supports schedule management by allowing veterinarians to define their availability on specific days and times. This scheduling feature helps prevent appointment conflicts and ensures that bookings align with veterinarian availability.

4.2.5. Appointment Booking and Management

Pet owners can book appointments by selecting a pet, clinic, veterinarian, and preferred date and time. The system validates pet ownership and veterinarian–clinic assignments before confirming a booking. Appointments progress through defined statuses such as scheduled, completed, or cancelled. Veterinarians and administrators can manage appointment details according to their access rights.

4.2.6. Treatment Records

After an appointment is completed, veterinarians can create treatment records that document diagnoses and medical notes. These records are directly linked to specific appointments, forming a structured medical history for each pet. This feature supports continuity of care and provides reliable documentation for future reference.

4.2.7. Owner and User Management

The system maintains ownership records that associate users with their pets, allowing support for multiple pets per owner. Administrative users can access user data for management and monitoring purposes, while sensitive information such as password hashes is strictly protected.

4.2.8. Reporting and Administration

PawPoint includes reporting features that allow administrators to analyze appointment statistics, clinic activity, and treatment data. These reports support operational decision-making, workload monitoring, and service quality evaluation.

4.3. API Functionality

4.3.1. Authentication

- POST `/register` — Register a new user

- POST `/login` — Log in and receive a JWT token

- GET `/profile` — View user profile (requires auth)

4.3.2. Pets

- POST `/pets` — Create a new pet (owner/admin)

- GET `/pets` — List pets (owner/admin)

- GET `/pets/<id>` — View pet details

- PUT `/pets/<id>` — Update a pet

- DELETE `/pets/<id>` — Delete a pet

4.3.3. Appointments

- POST `/appointments` — Create an appointment (owner/admin)

- GET `/appointments` — List appointments

- GET `/appointments/<id>` — View appointment details

- PUT `/appointments/<id>` — Update an appointment (vet/admin)

- PUT `/appointments/<id>/status` — Update appointment status

4.3.4. Clinics

- GET `/clinics` — List clinics

- GET `/clinics/<id>` — View clinic details

- POST `/clinics` — Create a new clinic (admin)

- PUT `/clinics/<id>` — Update a clinic (admin)

4.3.5. Veterinarians

- GET `/veterinarians` — List veterinarians

- GET `/veterinarians/<id>` — View veterinarian details

- GET `/veterinarians/clinic/<clinic_id>` — View veterinarians in a specific clinic

- POST `/veterinarians` — Create a new veterinarian (admin)

- GET `/veterinarians/<id>/schedules` — View veterinarian schedules

- POST `/veterinarian-schedules` — Create veterinarian schedules

4.3.6. Treatments

- GET `/treatments` — List treatments (vet/admin)

- GET `/treatments/<id>` — View treatment details (vet/admin)

- POST `/treatments` — Create a treatment record (vet/admin)

- PUT `/treatments/<id>` — Update a treatment record (vet/admin)

4.3.7. Reports (Admin Only)

- GET `/reports/appointments/status` — Appointment report by status

- GET `/reports/appointments/clinic` — Appointment report by clinic

- GET `/reports/treatments` — Treatments report

5. Testing and Results

5.1. Flow

5.1.1. Landing Page

The Landing Page serves as the initial entry point for the PawPoint application, designed to welcome users and guide them through the system's core features. When users first visit the application, they encounter a visually engaging interface with a prominent gradient header featuring the PawPoint branding and logo. The page displays three feature cards highlighting the key benefits of the system: Easy Scheduling for appointment management, Pet Management for storing pet information, and Health Tracking for accessing treatment records. If users are already authenticated, the landing page redirects them to their role-specific dashboard with a personalized welcome message showing their first name.

For unauthenticated users, the Landing Page functions as a marketing and navigation hub. It provides two primary call-to-action buttons—"Login" and "Get Started" (Register)—allowing new users to create accounts or existing users to sign in. The page emphasizes the platform's benefits through a comprehensive "Why Choose PawPoint?" section that outlines security, reliability, and ease of use. The gradient design and smooth animations create a modern, professional aesthetic that builds confidence in the application's quality and reliability.

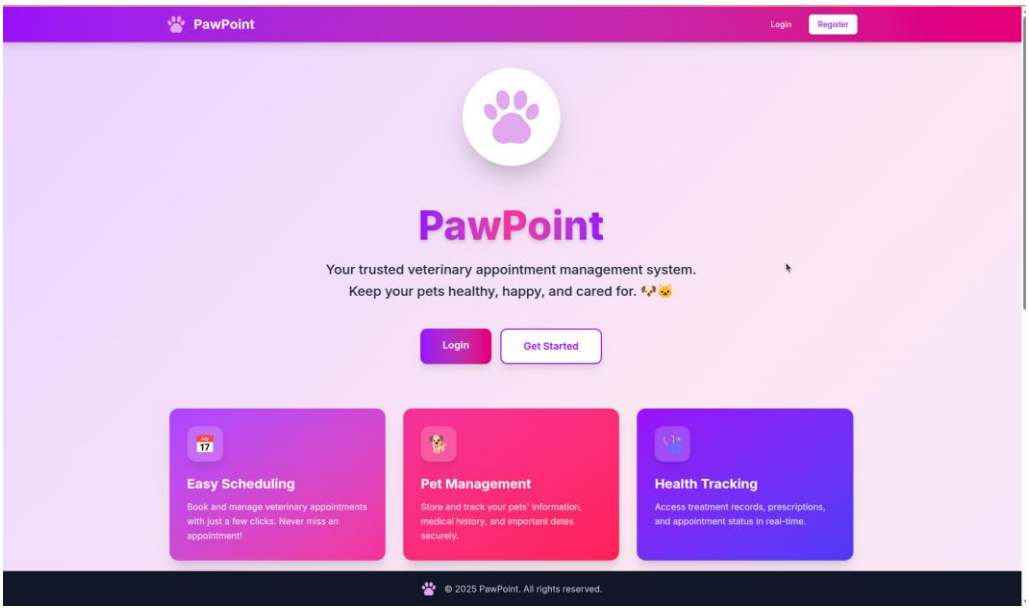


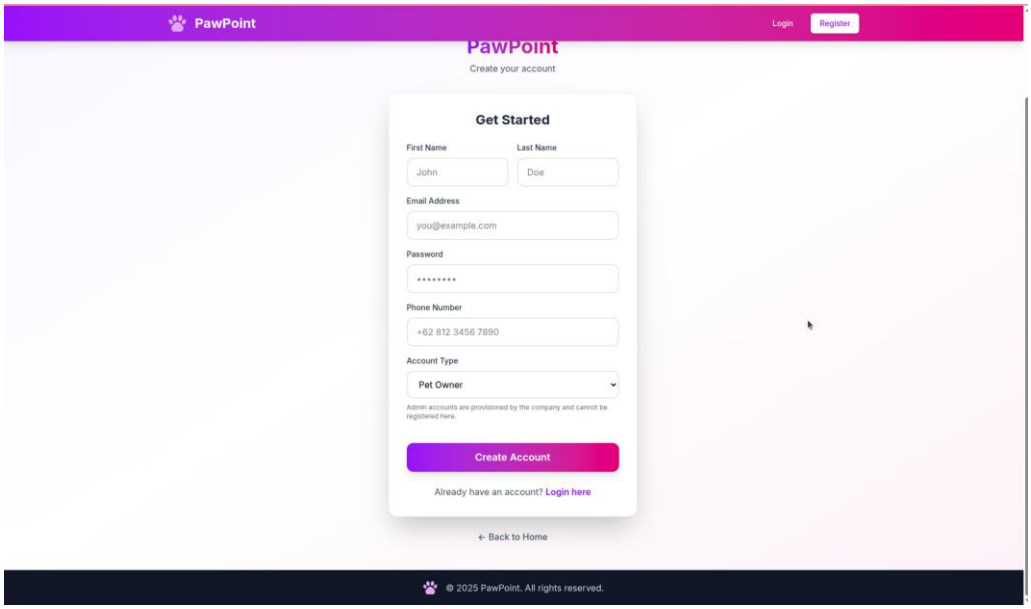
Image 1. The main page when opening the PawPoint website

5.1.2. Register page

The Register Page is the primary onboarding interface where new users create their accounts in the PawPoint system. Users begin by entering basic personal information including first name, last name, email, password, and phone

number. The page includes a critical dropdown menu for selecting their account type: Pet Owner or Veterinarian. Admin accounts cannot be self-registered and are provisioned by company administrators only. This role selection is fundamental as it determines which features and functionalities are available to the user post-registration.

When users select "Veterinarian" as their role, two additional required fields appear dynamically: a Veterinarian License Number and a Clinic/Hospital Affiliation dropdown. The License Number field displays validation text explaining that the license must be pre-registered in the system to create a veterinarian account. The clinic dropdown is dynamically populated by fetching available clinics from the backend API, allowing veterinarians to immediately associate themselves with their workplace. Client-side validation ensures all required fields are completed before submission, and the registration endpoint creates a new user record with the appropriate role-based relationships in the database.



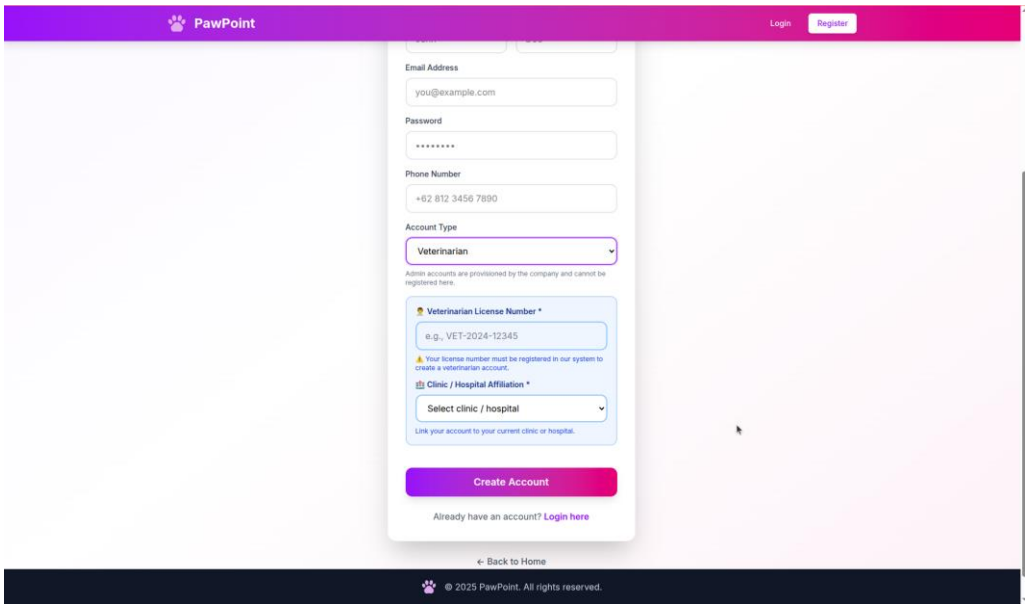


Image 2. The Registering page, when a new user wants to create a new account

5.1.3. Login Page

The Login Page provides a straightforward authentication interface where existing users enter their email and password credentials to access the system. The page features the PawPoint logo and branding at the top, creating visual consistency with the rest of the application. Users input their email address and password into clearly labeled form fields with placeholder text for guidance. The page includes error handling that displays detailed error messages if authentication fails, helping users troubleshoot issues such as incorrect credentials or account lockouts.

Upon successful authentication, the backend validates credentials against the stored password hash and generates a JWT token. The Login Page then retrieves user data including user_id, role, and name, storing this information in the browser's localStorage through the AuthContext. The role-based routing logic automatically redirects authenticated users to their appropriate dashboard: admin users to `/admin/dashboard`, veterinarians to `/vet/dashboard`, and pet owners to `/owner/dashboard`. This seamless redirection creates a personalized experience tailored to each user's permissions and responsibilities.

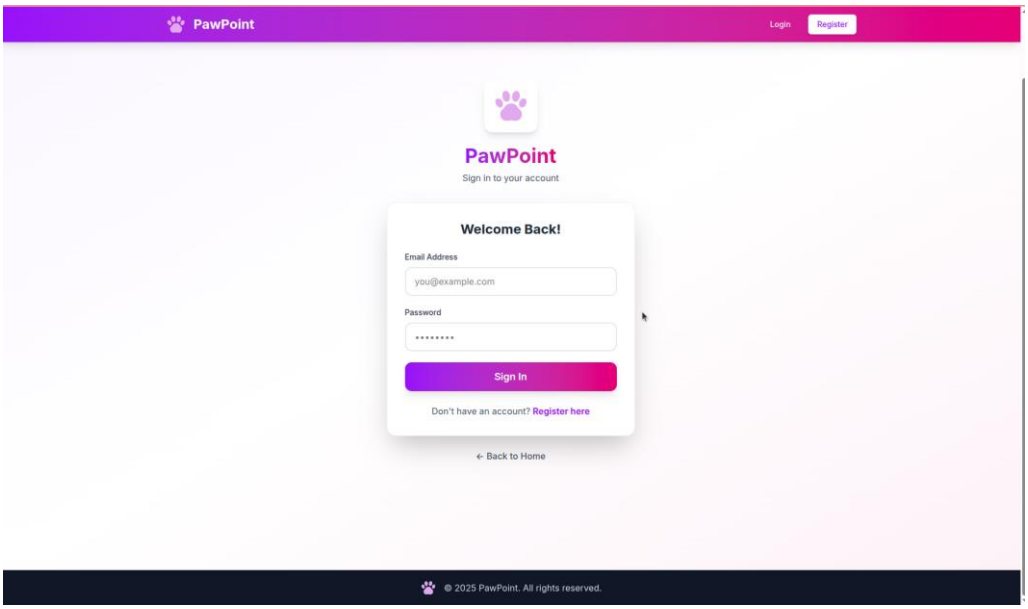


Image 3. The Login page when a user already has an account

5.1.4. Dashboard Page (Admin)

The Admin Dashboard serves as the command center for system administrators, providing a comprehensive overview of all platform activity and key metrics. The dashboard displays four critical statistics in visually distinct gradient cards: Total Appointments (blue), Pending Appointments requiring review (yellow/orange), Total Users across all roles (green), and Active Clinics available in the system (pink/purple). These metrics update in real-time,

providing administrators with immediate insight into system usage and pending workload.

The dashboard's primary feature is a "Pending Appointments" section that displays appointments awaiting administrative review or approval. Each appointment card shows the pet name, clinic, appointment date/time, pet owner's name, and veterinarian assigned. Administrators can take action on each appointment using "Approve" or "Reject" buttons, which update the appointment status to "completed" or "cancelled" respectively. This workflow ensures administrative oversight of appointment scheduling and helps maintain data quality. The page also provides quick links to access detailed management pages for users, clinics, pets, and reports, giving administrators centralized control over all system resources.

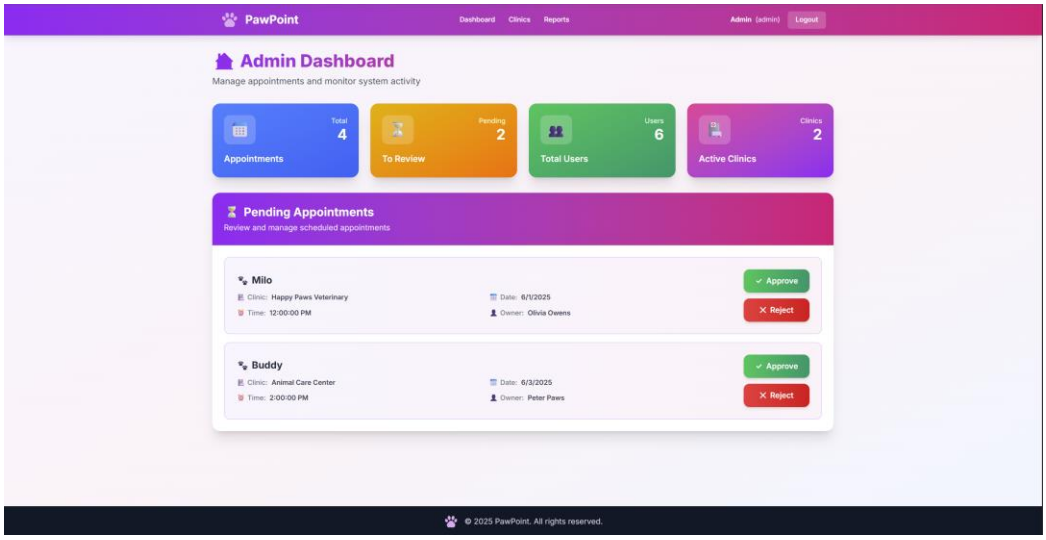


Image 4. The Dashboard of the admin of the clinic. They can see what is on schedule that day and can accept or reject an appointment

5.1.5. Clinics Page (Admin)

The Admin Clinics Management page enables administrators to manage all veterinary clinics registered in the PawPoint system. The page displays clinics in a searchable list or table format, showing each clinic's name, contact phone number, address, and associated veterinarians. Administrators can create new clinic records by filling out a form with the clinic name, phone number, and address information. This page ensures that clinic data remains current and accurate, which is essential for the appointment booking workflow where users select clinics during appointment scheduling.

Administrators can also view which veterinarians are affiliated with each clinic, helping them manage the veterinarian-clinic relationship mapping. This is particularly important for the appointment system's clinic filtering logic, which shows only veterinarians affiliated with the selected clinic when booking appointments. Maintaining accurate clinic information and veterinarian affiliations ensures that the appointment booking workflow functions smoothly and that users always see accurate available options.

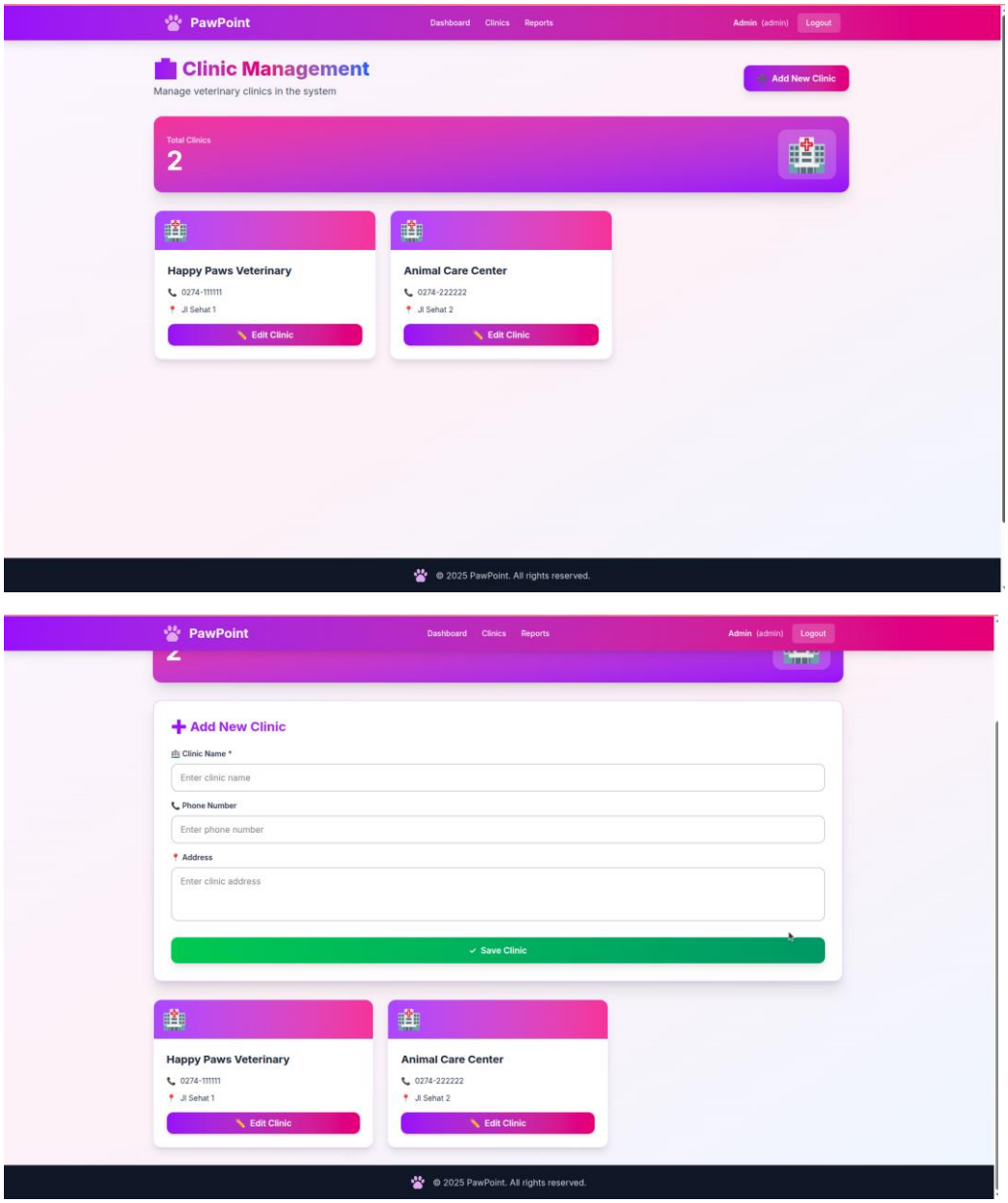


Image 5. The admin can manage and add clinics to the system

5.1.6. Reports Page (Admin)

The Admin Reports page provides administrators with analytics and insights into system usage and performance. The page displays reports for appointments grouped by status (scheduled, completed, cancelled), appointments grouped by clinic location, and overall treatment record statistics. These reports help administrators track key performance indicators such as appointment completion rates, clinic workload distribution, and treatment patterns across the platform. The data is typically presented in table format with summary statistics, allowing administrators to identify trends, troubleshoot issues, and make data-driven decisions about system management.

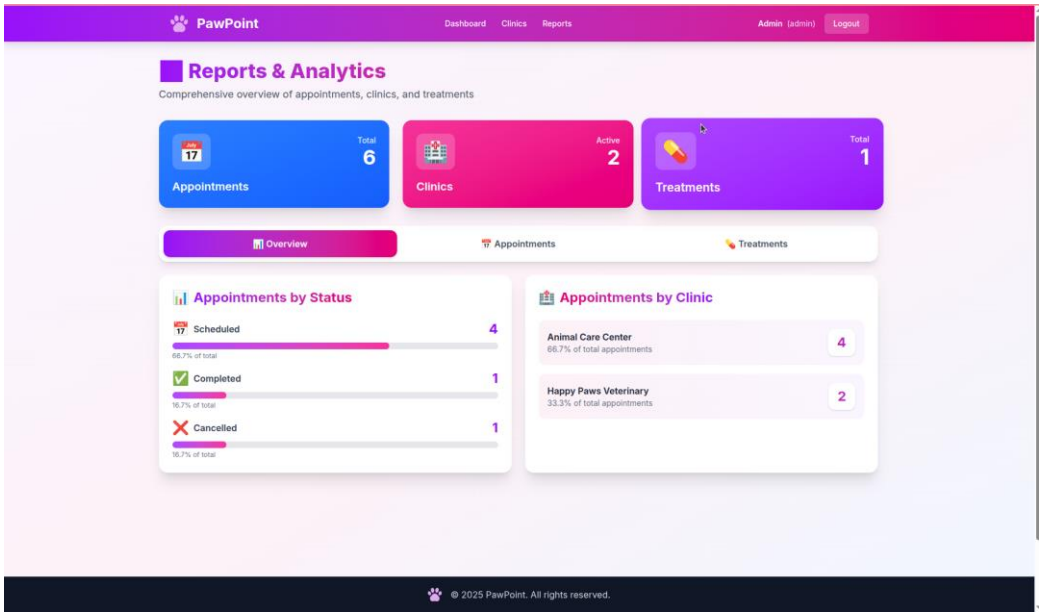


Image 6. The admin can see how each clinic is managing statistically

5.1.7. Dashboard Page (Pet Owner)

The Pet Owner Dashboard serves as the personal hub for pet owners, displaying key statistics about their pets and appointments. The page shows two main metrics: Total Pets (number of pets registered to the owner) and Total Appointments (all appointments for those pets). Below the statistics, the dashboard displays a "Recent Appointments" section listing all of the owner's appointments in chronological order. Each appointment card shows the pet name, clinic name, appointment date, and current appointment status (scheduled, completed, or cancelled) with color-coded status badges for easy visual reference.

This dashboard provides pet owners with a quick overview of their pets' healthcare status and upcoming or past appointments. The interface includes navigation buttons or links to access the full Pets page (to manage pet information) and the Appointments page (to book new appointments). The dashboard's purpose is to give busy pet owners a quick glance at what matters most: their pets' health and upcoming care schedules. The simple, focused design reduces cognitive load while providing essential information at a glance.

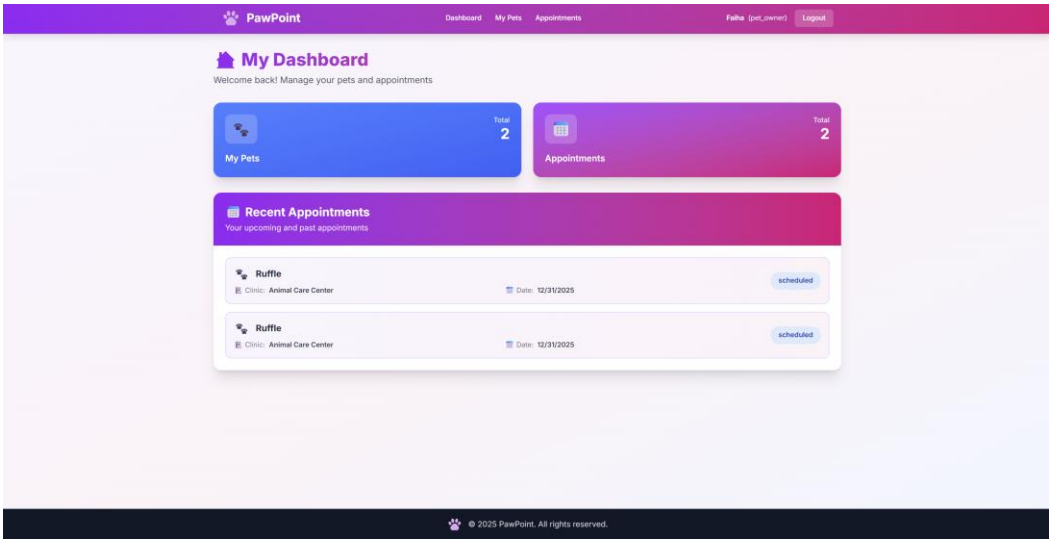


Image 7. The Dashboard of what a pet owner would see when they login into the website

5.1.8. Appointments Page (Pet Owner)

The Pet Owner Appointments Management page enables owners to view all their appointments and book new ones through an intuitive interface. The page displays existing appointments in a scrollable list, with each appointment showing the pet's name, assigned veterinarian, clinic, appointment date/time, and status. Owners can click an "Add Appointment" button to reveal a comprehensive booking form where they select: which pet to book for (dropdown of their pets), which clinic to use (dropdown of available clinics), which veterinarian at that clinic (dynamically filtered based on clinic selection), and the appointment date/time.

The form includes intelligent cascading dropdowns and validation: when an owner selects a clinic, the veterinarian dropdown is immediately populated with only the veterinarians affiliated with that clinic. When a veterinarian is selected, the form displays their available working hours for each day of the week, with color coding showing which times are available. The date/time picker validates that the selected appointment time falls within the veterinarian's scheduled availability, displaying a "✅ Time is available" or "❌ Time is NOT available" message. This intelligent validation prevents booking conflicts and ensures all appointments are scheduled during actual clinic operating hours. Once submitted, the appointment is created with "scheduled" status and appears in the appointments list.

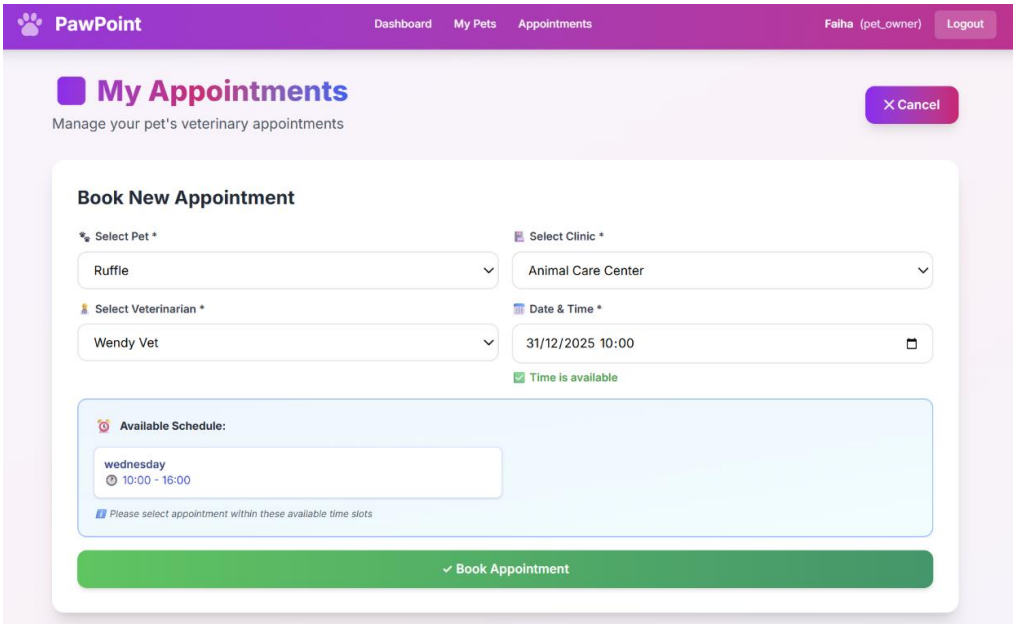


Image 8. The pet owner can book a new appointment by clicking on the 'Add Appointment' button on the top right

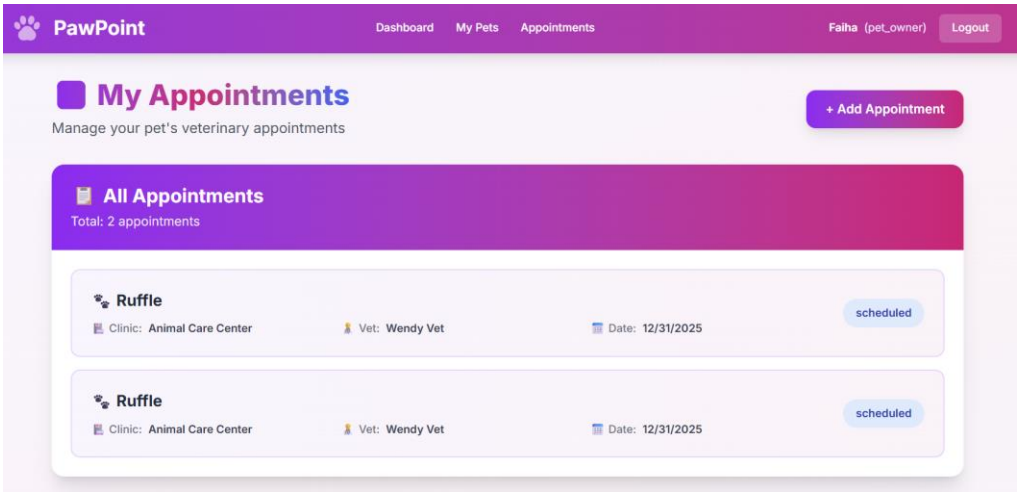


Image 9. The pet owner can see all the appointments that they have for their pets, which vet, the date, and whether the appointment is scheduled/cancelled.

5.1.9. Pet Detail Page (Pet Owner)

The Pet Owner Pet Detail Page provides comprehensive information about an individual pet, serving as the pet's complete health and information record. The page displays all the pet's basic information (name, species, breed, gender, age, birth date) in an easy-to-read format with section headers and icons. The page includes a dedicated section showing all treatment records associated with that pet, with each treatment displaying the date, veterinarian's name and license number, diagnosis, and notes from the treatment.

Additionally, the page displays all appointments scheduled for or completed for that specific pet, showing appointment dates, times, clinics, assigned veterinarians, and current status. This consolidated view allows pet

owners to track their pet's complete healthcare journey on one page. Owners can see patterns in their pet's health issues, track which veterinarians have treated their pet, and review past treatment recommendations. The page often includes an "Edit Pet Info" button allowing owners to update the pet's information, and a "View Medical Records" link that may provide additional PDF reports or documentation. This comprehensive detail page empowers pet owners to be fully informed advocates for their pets' healthcare.

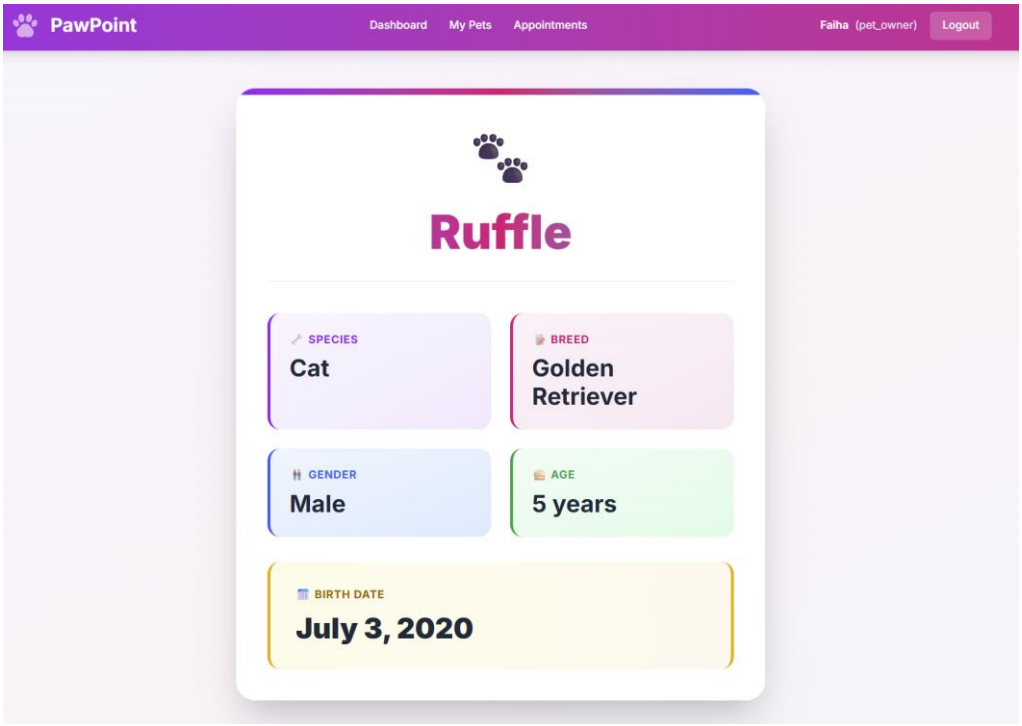


Image 10. The pet owner can click on each individual pet and see the details of their pet

5.1.10. My Pets Page (Pet Owner)

The Pet Owner Pets Management page displays all pets belonging to the owner and allows them to add new pets to the system. The page features an "Add Pet" button that reveals a form for registering new pets. The form collects comprehensive pet information including: pet name (required), species such as "Dog" or "Cat" (required), breed, gender (male/female/unknown), birth date, and age in years. Once submitted, the pet is created in the database and immediately appears on the page as a visually appealing card in a grid layout.

Each pet card displays the pet name prominently in a gradient text style, along with an emoji icon (paw print) for visual appeal. The card also shows key pet information such as species, breed, gender, and age. The cards are clickable, allowing owners to navigate to the individual pet detail page where they can view the pet's full medical history, treatment records, and appointment history. This design encourages owners to keep their pets' information current and provides easy access to each pet's complete health record. The pet cards use a responsive grid layout that adapts to different screen sizes, ensuring usability on both desktop and mobile devices.

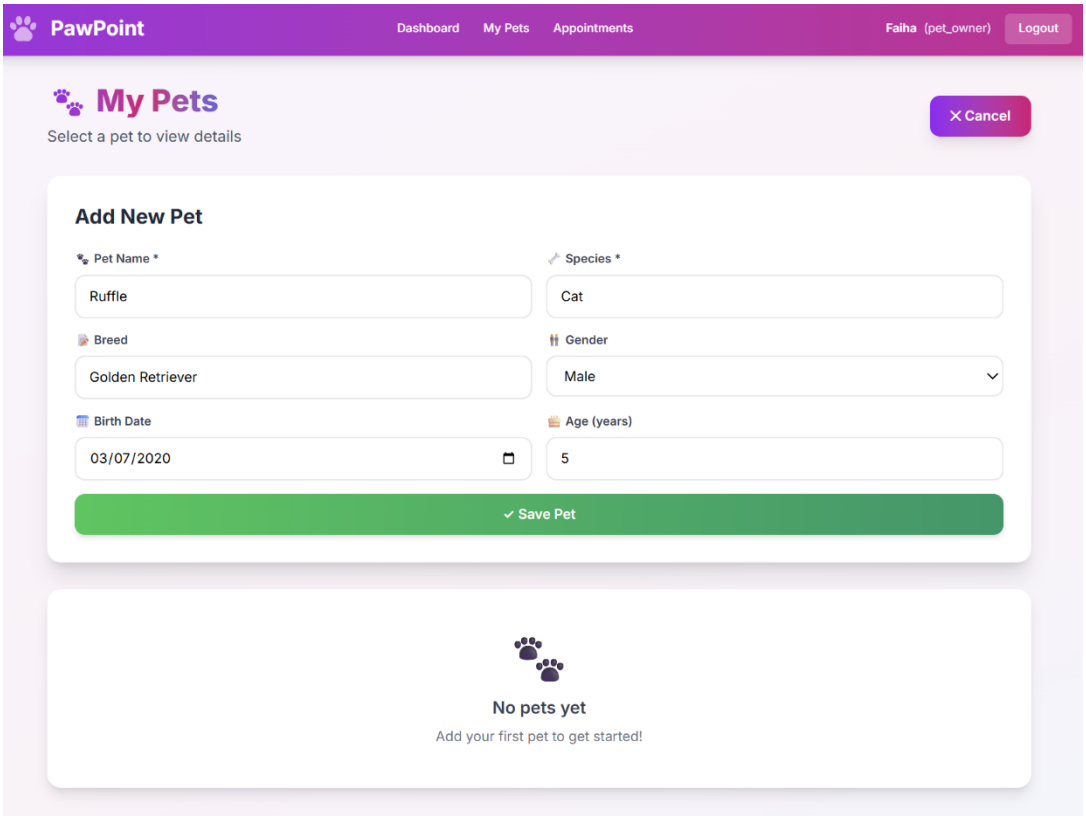


Image 11. The pet owner can add a new pet when they click on the ‘My pets’ tab at the top and add all the necessary details of this new pet

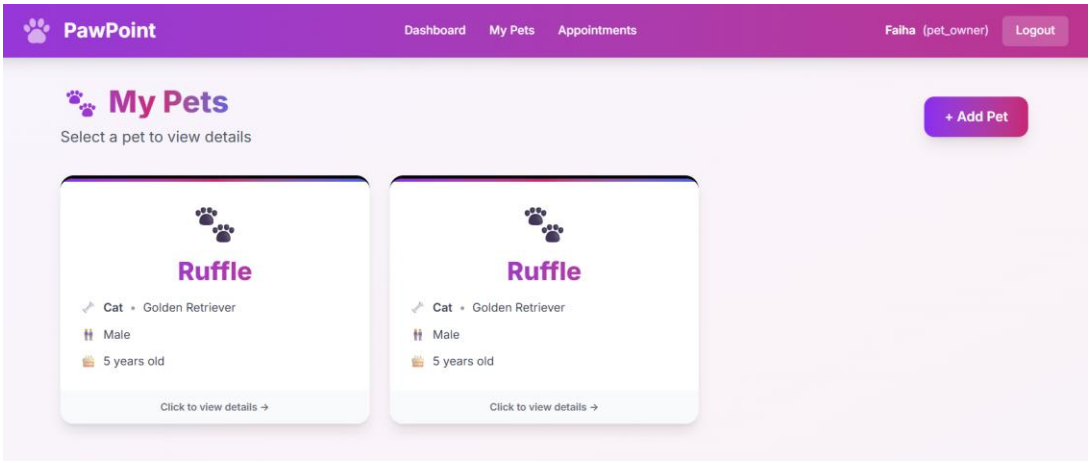


Image 12. The pet owner can see all the pets that they have added into the PawPoint system

5.1.11. Dashboard Page (Veterinarian)

The Veterinarian Dashboard provides veterinarians with an overview of their professional activity, displaying key metrics in three gradient cards: Total Appointments (all appointments assigned to them), Scheduled Appointments (pending appointments not yet completed), and Completed Appointments (appointments already finished). Below these metrics, a "My Appointments" section displays all appointments assigned to the veterinarian in a list format,

showing pet names, clinic location, appointment date/time, pet owner names, and appointment status with color-coded badges.

The dashboard serves as the veterinarian's command center for managing their patient schedule. Veterinarians can quickly see how many appointments they have, how many are still pending, and how many they've completed, providing insight into their productivity and workload. The appointments list allows veterinarians to click on individual appointments to view details or navigate to the appointments management page to take actions such as completing appointments. The simple, focused dashboard design ensures veterinarians can quickly assess their day's schedule without distractions, helping them manage their time effectively.

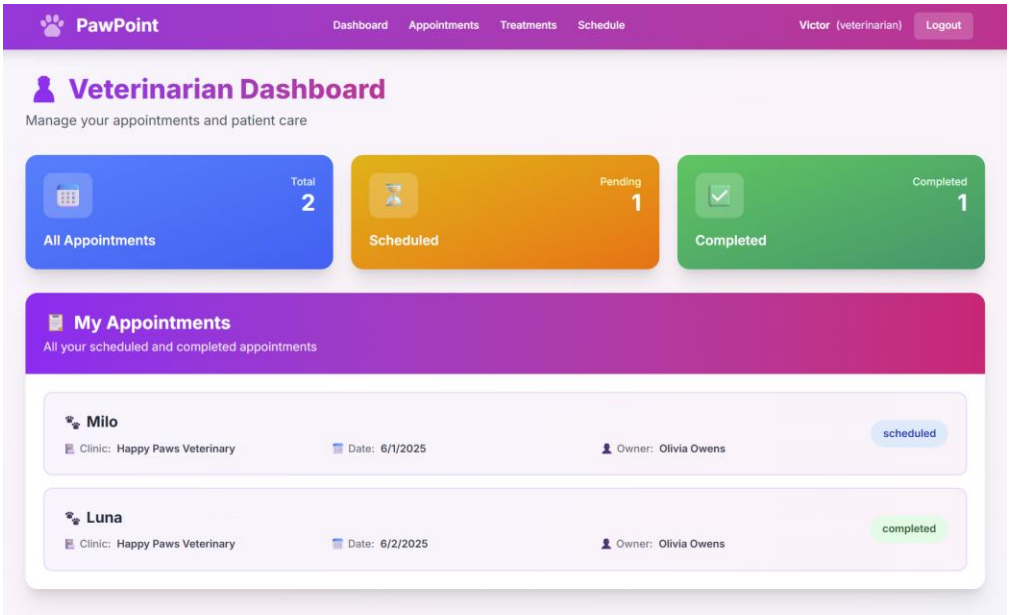


Image 13. The Dashboard of the Veterinarian. They can see all the appointments that they have scheduled on that day, and the status of the appointments; scheduled, cancelled, or completed.

5.1.12. Appointments Page (Veterinarian)

The Veterinarian Appointments Management page displays all appointments assigned to the logged-in veterinarian and provides tools to manage appointment status. The page shows a comprehensive list of appointments with details including pet name, clinic, appointment date/time, owner name, and current status. For appointments with "scheduled" status, veterinarians see "Complete" and "Cancel" buttons that allow them to update the appointment status.

When a veterinarian clicks "Complete" on a scheduled appointment, the appointment status changes to "completed" and a modal form appears prompting them to enter a treatment record with a diagnosis and optional notes. This workflow ensures that every completed appointment has an associated treatment record documenting what care was provided. The treatment record is saved to the database and becomes part of the pet's permanent medical history. If a

veterinarian needs to cancel an appointment, they can click "Cancel" to change its status to "cancelled", and no treatment record is created. This page essentially serves as the veterinarian's patient management interface, documenting care provided and maintaining accurate medical records.

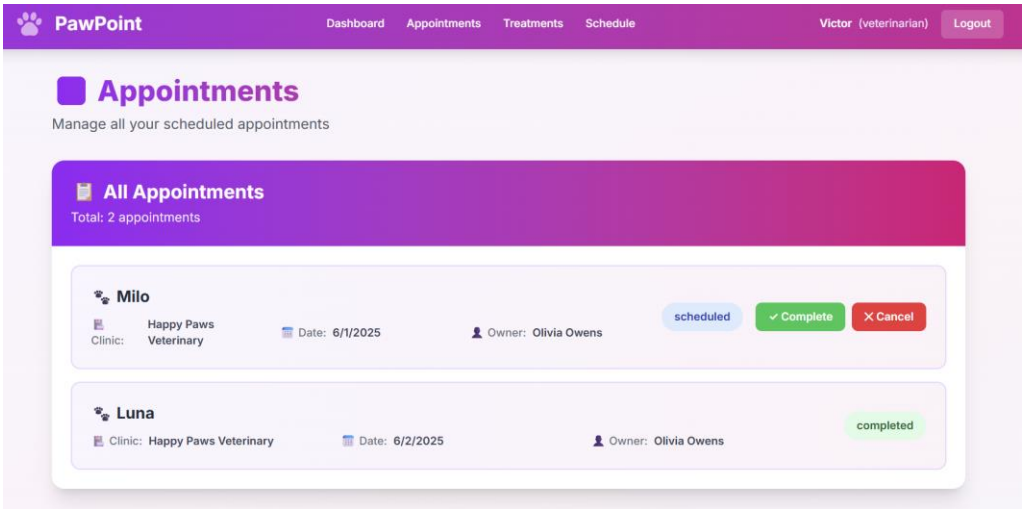


Image 14. The veterinarian can see the appointments they have on that day, and once one is done they can update the website by clicking on ‘complete’, or ‘cancel’ if anything happens.

5.1.13. Schedule Page (Veterinarian)

The Veterinarian Schedule Management page allows veterinarians to set and manage their weekly work schedule, defining when they are available to see patients. When the veterinarian first accesses this page, the system automatically loads their existing schedule if one has been created during registration or previous sessions. The schedule is displayed in a weekly calendar format, showing each day of the week (Monday through Sunday) with the veterinarian's start and end times for that day.

Veterinarians can add new schedule entries for days they don't yet have scheduled using an "Add New Schedule" form. The form includes a day-of-week dropdown selector, start time input, and end time input. The system validates that the end time is after the start time and prevents duplicate schedules for the same day (only one schedule entry per day is allowed, though this can be modified by editing or deleting the existing entry). Once added, the schedule appears in the calendar view sorted by day of the week. This schedule information is critical because it's used by the appointment booking system to validate that appointment times fall within the veterinarian's working hours. When pet owners book appointments, they can only select times that match a veterinarian's scheduled availability, ensuring that no appointments are booked outside working hours.

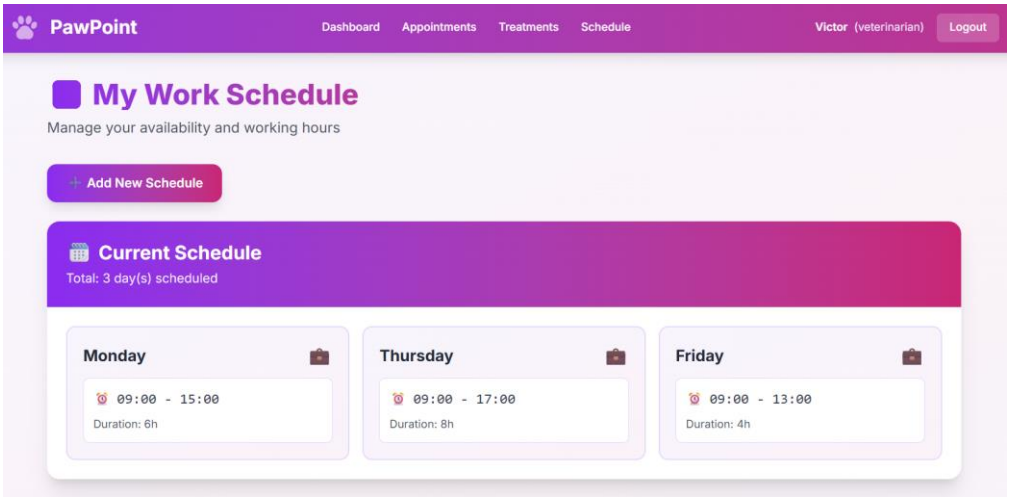


Image 15. The veterinarian's schedule for the week can be seen in the 'schedule' tab

5.1.14. Treatments Page (Veterinarian)

The Veterinarian Treatments View page displays all treatment records created by the logged-in veterinarian, providing a comprehensive log of patient care delivered. The page lists each treatment record with the following details: pet name, treatment date, veterinarian name and license number, diagnosis, and treatment notes. The records are displayed in a card-based list format with alternating background colors for visual distinction. The treatment records are automatically created when a veterinarian marks an appointment as "completed" and enters treatment information in the appointment management page.

This page serves as the veterinarian's treatment history and medical documentation record. Veterinarians can use it to review their past cases, verify that treatment records were properly documented, and provide information if needed for audits or follow-up questions. The page displays only treatments created by the logged-in veterinarian (filtered by user_id), ensuring veterinarians can only see their own records. This page may also support filtering by date range or pet, allowing veterinarians to quickly locate specific treatment records for reference or follow-up care planning.

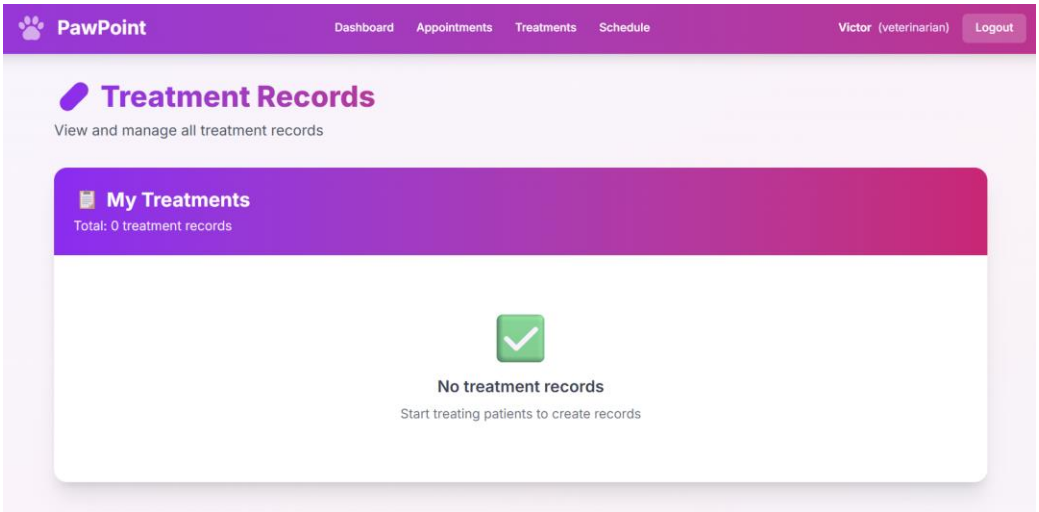


Image 16. The veterinarian can see all the treatment records that they have for all the pets they have done treatment on, and what they have prescribed for each pet.

6. Conclusion and Reflection

6.1. Lesson Learned

The development of the PawPoint ER diagram provided the team with significant insights into the importance of data normalization and relationships integrity. Through this process, we learned how to effectively separate user authentication data from role-specific details (PetOwners and Veterinarians) to eliminate data redundancy. Furthermore, the team gained a deeper understanding of how associative entities, such as VeterinarianClinic, are vital for resolving many-to-many relationships, ensuring the system remains scalable as more clinics and staff are integrated.

6.2. Challenges

One of the primary challenges faced by the group was the logical mapping of the appointment entity. As the central hub connecting the Pet, the Veterinarian, and the Clinic, it requires careful planning of cardinality to ensure data remained consistent, the team also had to critically evaluate which attributes should be stores statically and which should be derived, specifically the pet's age, to ensure that the database provide accurate, real time information without requiring constats manual updates.

6.3. Further Improvements

While the current schema successfully covers core clinical operations, the group has identified several areas for future enhancement:

a. Billing and Invoicing: Incorporating a Payment entity linked to TreatmentRecord to automate the billing process.
b. Automated reminders: Implementing a notification system to alert owners of upcoming vaccinations or follow-up visits based on the pet's medical history.
c. Multimedia Integration: Extending the TreatmentRecord to support file attachments such as X-rays, lab results, and digital prescriptions

7. Appendix

All codes are available on Github
https://github.com/deiraaisyar/DBProject_Group10_PawPoint.