PET CLINIC DATABASE MANAGEMENT SYSTEM

UE23CS351A

Database Management System

TEAM	
K RAJEEV	PES1UG23AM130
KANNIYAPPA RANJITH M	PES1UG23AM136

Abstract

This project implements a comprehensive Database Management System (DBMS) for a pet clinic management platform, utilizing MySQL for data persistence and Python Flask for web application development. The system manages core entities including pet owners, pets, veterinarians, appointments, treatment records, and billing, with enforced referential integrity through foreign key constraints and cascade operations.

The application features a modern web interface with professional design elements, including a left-sidebar navigation for improved usability. Key functionalities encompass appointment scheduling with automatic veterinarian assignment, treatment record management, billing integration with payment tracking, and comprehensive reporting capabilities. The system supports complex database queries including joins, nested queries, and aggregate functions to provide business intelligence and operational insights.

The DBMS architecture ensures data consistency through proper normalization, constraint validation, trigger-based automation, and error handling mechanisms. This project demonstrates practical implementation of relational database design principles combined with full-stack web development, creating an efficient solution for managing multi-veterinarian pet clinic operations with medical records tracking and integrated billing systems.

User Requirement Specifications

Purpose of the Project

- ✓ The purpose of this project is to design and develop a comprehensive Database Management System (DBMS) for a pet clinic management platform. The system aims to streamline operations by managing complex relationships between pet owners, pets, veterinarians, appointments, medical treatments, and billing in a centralized database.
- ✓ By implementing a robust relational database with proper data integrity constraints and a user-friendly web interface, the system enables efficient appointment scheduling, medical record tracking, and billing management.
- ✓ The primary goal is to provide stakeholders—including clinic administrators—with a unified platform to manage daily operations while maintaining data consistency, accessibility, and medical record integrity.

Scope of the Project

- ✓ The scope of this DBMS project encompasses the design and implementation of a complete pet clinic ecosystem.
- ✓ The system will manage pet owners with multiple pets, veterinarians with specializations, appointment scheduling, treatment records with medical details, and integrated billing.
- ✓ The project includes database design with proper normalization, implementation of referential integrity through foreign key constraints, comprehensive triggers for business logic automation, and a web-based interface for user interaction.

The system will support core operations such as:

- ✓ Owner registration and management
- ✓ Pet registration and medical history tracking
- ✓ Veterinarian management with specializations
- ✓ Appointment scheduling with automatic primary vet assignment
- ✓ Treatment record documentation with costs
- ✓ Billing and payment tracking with multiple payment modes
- ✓ Advanced reporting on clinic performance and statistics

The project also incorporates advanced SQL queries including joins, nested queries, and aggregate functions for reporting purposes, along with stored procedures for complex operations. However, the scope excludes prescription management, laboratory integration, insurance claim processing, and external API integrations.

Detailed Description

The Pet Clinic Management System is a multi-tiered application that connects pet owners with veterinary services. The system maintains detailed information about all stakeholders and their interactions throughout the appointment and treatment lifecycle.

Core Entities and Relationships:

Owners & Contact Management: The Owner table stores owner details including contact information, addresses, and phone numbers. Owners can have multiple email addresses tracked in the Owner_Email table (one-to-many relationship with cascade delete).

Pets & Medical Records : The Pet table maintains information about all pets, linked to owners through foreign keys. Each pet has details such as name, species, breed, and date of birth. Pets can have a one-to-many relationship with appointments.

Veterinarians & Specializations : The Veterinarian table stores veterinarian details including name, specialization, phone, and years of experience. The Vet_Treats_Pet junction table maintains a many-to-many relationship between veterinarians and pets, with a special attribute (is_primary_vet) to track primary veterinarian assignments.

Appointments & Scheduling : The Appointment table records appointment details including date, time, reason, and status (Scheduled, Completed, Canceled). Appointments link pets to veterinarians and trigger automatic billing.

Treatment Records & Medical History : The Treatment_Record table stores detailed treatment information including description, medicines prescribed, veterinarian notes, and treatment costs. This creates a complete medical history for each pet.

Billing & Payment Management : The Billing table tracks all payments, linking to appointments through foreign keys. It maintains payment status (Paid, Unpaid) and payment mode, with automatic Bill IDs using AUTO_INCREMENT.

Data Integrity: Foreign key constraints are implemented to ensure referential integrity between all tables, preventing orphaned records. CASCADE DELETE operations ensure data consistency when records are removed.

Software Used

Backend Development

Python 3.x – Primary programming language for server-side logic

Flask – Web framework for building the application and routing HTTP requests

MySQL 8.0 – Relational Database Management System for data storage and queries

Frontend Development

HTML5 – Markup language for web page structure

CSS3 – Styling with gradients, animations, responsive 2-column layouts, and collapsible sections

JavaScript (ES6) – Client-side interactivity, form validation, and dynamic UI updates

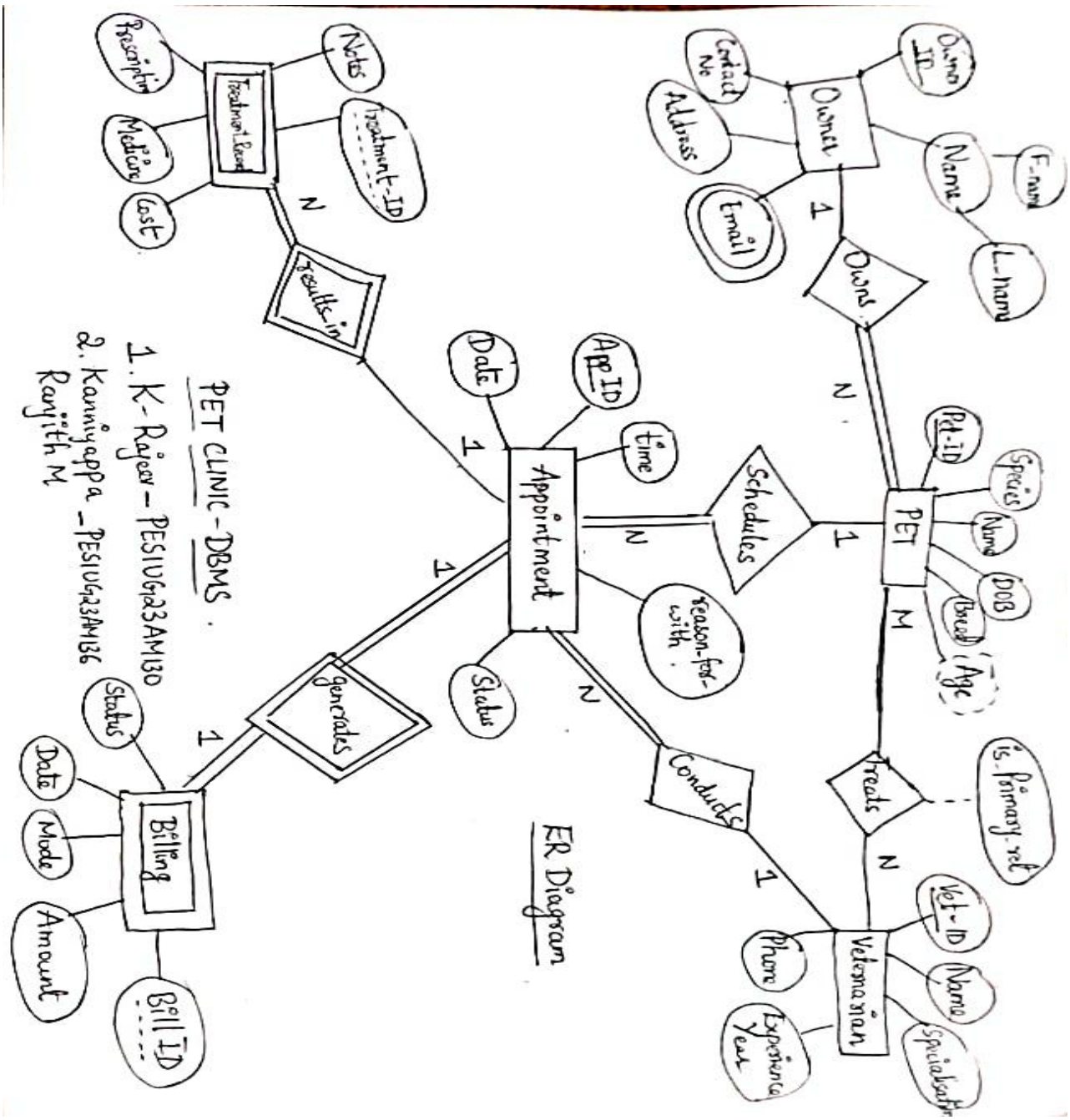
Libraries & Tools

Flask-MySQL– Python connector for MySQL database integration

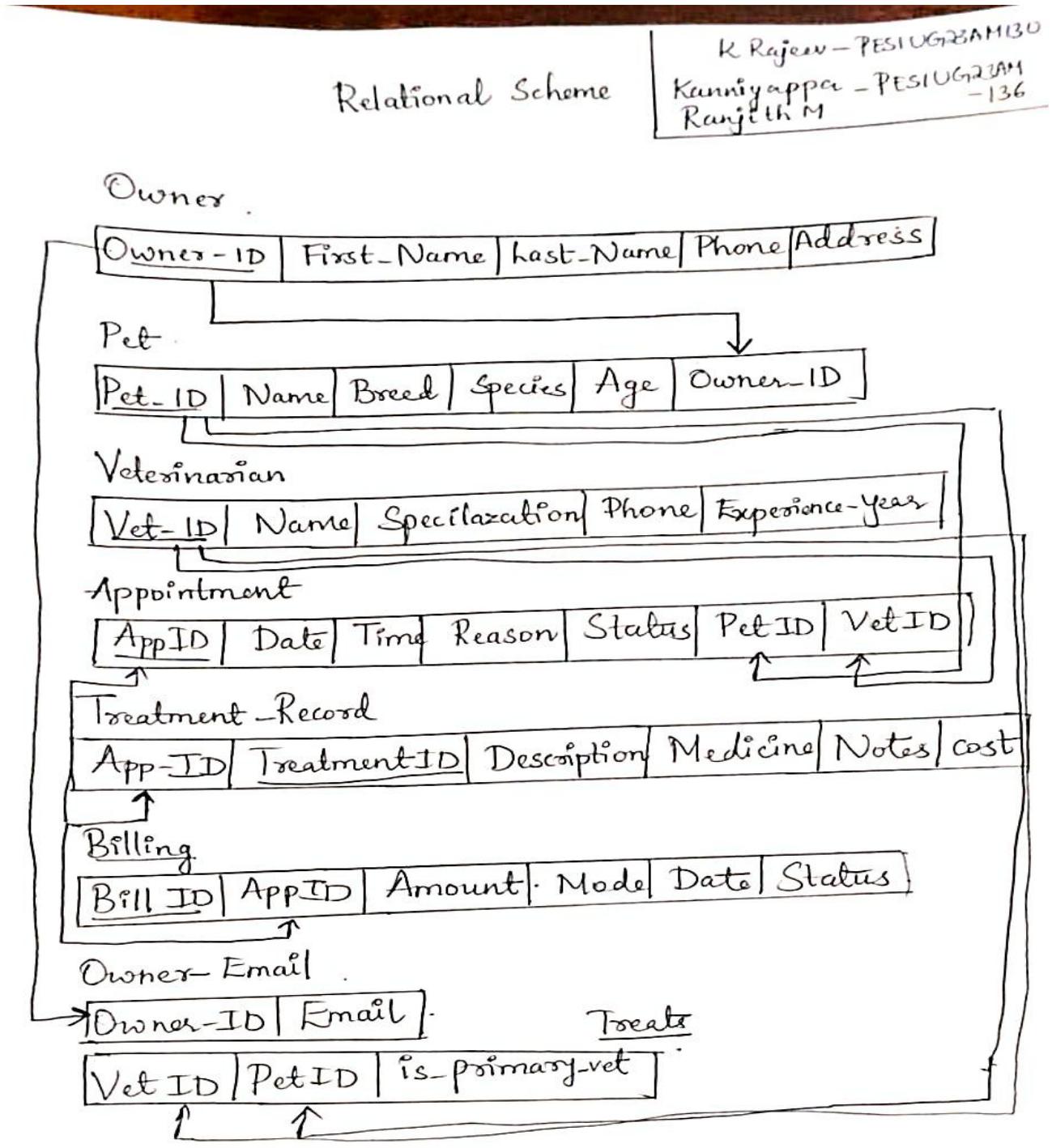
Jinja2– Template engine for dynamic HTML rendering

Date/Time Handling – TIMESTAMPDIFF for age calculations

ER Diagram


PET CLINIC - DBMS .
1. K- Rajeev - PES1UG23AM130
2. Kanniyappa - PES1UG23AM136
Ranjith M
ER Diagram
Owner
Owner ID
Name
F_name
L_name
Contact No
Address
Email
Owns
1
N
PET
Pet-ID
Species
Name
DOB
Breed
Age
M
Treats
N
Is_Primary_vet
Veterinarian
Vet-ID
Name
Specialization
Phone
Experience Years
1
Conducts
N
Appointment
App ID
Date
time
Reason-for-visit
Status
Schedules
1
N
1
N
Generates
1
1
Billing
Bill ID
Amount
Mode
Date
Status
Treatment Record
Treatment-ID
Notes
Cost
Medicine
Prescription
Results in
N

Relational Schema



DDL Commands

1. CREATE TABLE

Used to create database tables with their structure and constraints.

Examples from project:

CREATE TABLE Owner – Stores owner information with phone uniqueness

CREATE TABLE Owner_Email– Stores multiple emails per owner (composite primary key)

CREATE TABLE Veterinarian – Stores vet details with specialization

CREATE TABLE Pet – Stores pet information linked to owners

CREATE TABLE Appointment – Stores appointment records with status tracking

CREATE TABLE Treatment_Record – Stores medical treatment details (composite key)

CREATE TABLE Billing – Stores billing with auto-increment Bill IDs

CREATE TABLE Vet_Treats_Pet– Junction table with many-to-many relationship

Key Features:

Composite primary keys for complex relationships

Foreign key constraints with CASCADE DELETE

UNIQUE constraints on critical fields

ENUM types for status fields

AUTO_INCREMENT for ID generation

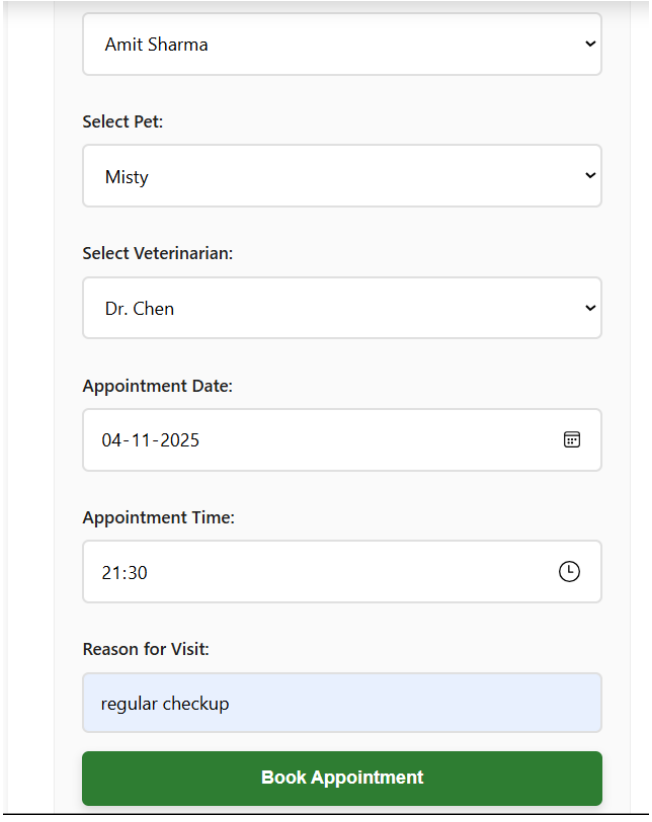
2. CREATE TRIGGER

Defines triggers to automatically execute actions on table events.

Examples in project:

before_appointment_insert – Validates appointment date to prevent past bookings

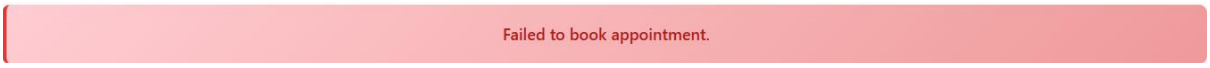
```
CREATE TRIGGER before_appointment_insert
BEFORE INSERT ON Appointment
FOR EACH ROW
BEGIN
    IF NEW.Date < CURDATE() THEN
        SIGNAL SQLSTATE '45000'
        SET MESSAGE_TEXT = 'Error: Cannot book an appointment in the past.';
    END IF;
END;
```



Entering past date for appointment

Triggered 'Error: cannot book appointment in the past.'

After booking appointment



3. CREATE FUNCTION

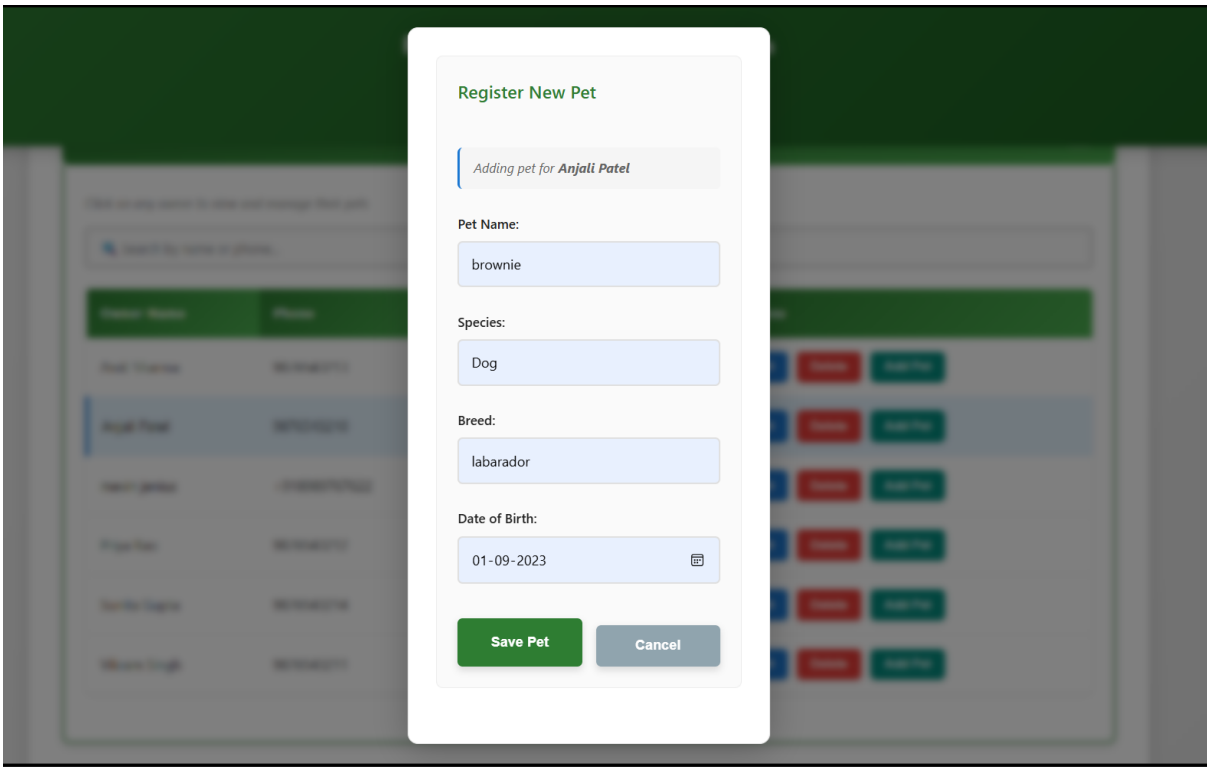
Defines reusable functions for calculations.

Example in project:

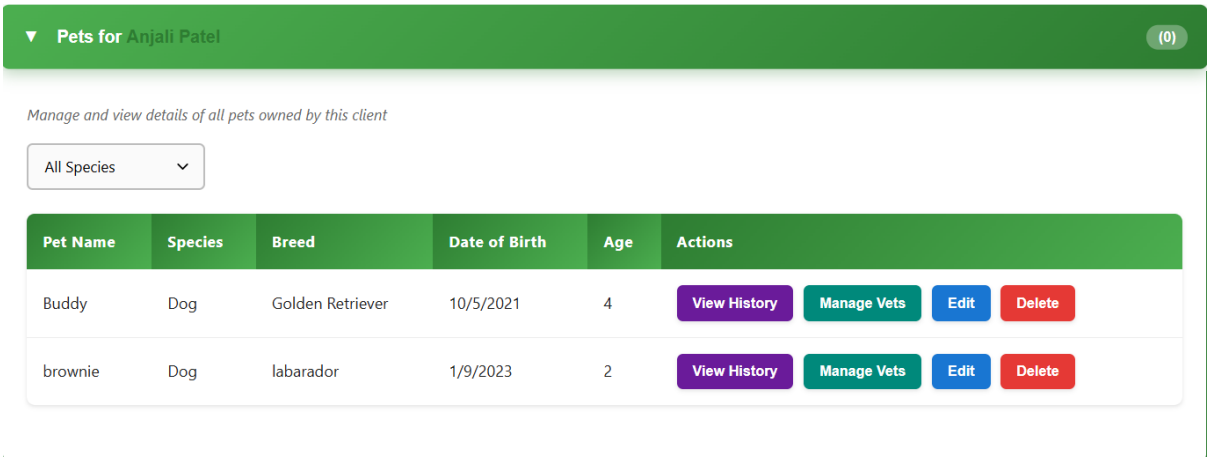
CalculateAge(p_DoB DATE) – Calculates pet age in years

```
CREATE FUNCTION CalculateAge(p_DoB DATE) RETURNS INT
DETERMINISTIC
BEGIN
    RETURN TIMESTAMPDIFF(YEAR, p_DoB, CURDATE());
END;
```

Registering new pet by giving Date Of Birth



It automatically calculates the Age of the pet and saves it.



4. CREATE PROCEDURE

Defines stored procedures for complex database operations.

Example in project:

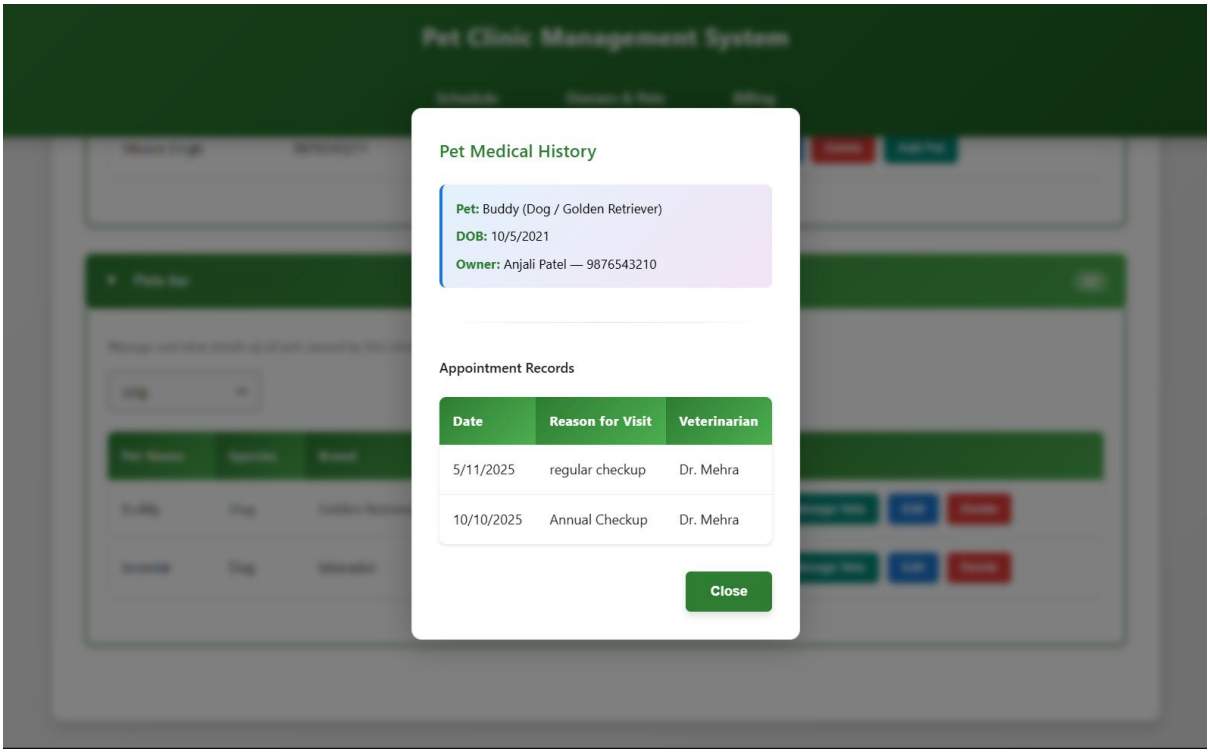
GetPetHistory(p_PetID INT)– Retrieves complete pet history with owner and vet information

```
CREATE PROCEDURE GetPetHistory(IN p_PetID INT)
BEGIN
    SELECT
        p.Name AS PetName,
        p.Species,
        p.Breed,
        p.DoB,
```

```
        o.FirstName,
        o.LastName,
        o.Phone AS OwnerPhone,
        a.AppID,
        a.Date AS AppointmentDate,
        a.Reason,
        v.Name AS VetName
    FROM Pet p
    JOIN Owner o ON p.OwnerID = o.OwnerID
    LEFT JOIN Appointment a ON p.PetID = a.PetID
    LEFT JOIN Veterinarian v ON a.VetID = v.VetID
    WHERE p.PetID = p_PetID
    ORDER BY a.Date DESC;
END;
```

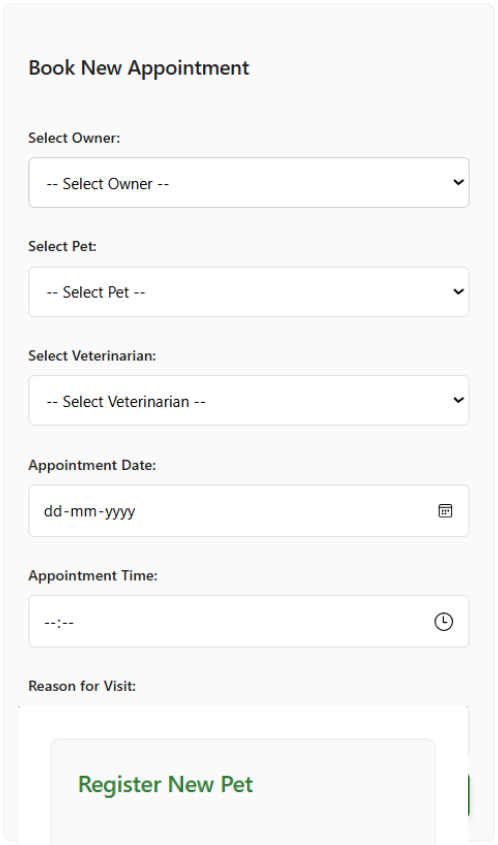
Usage: `CALL GetPetHistory(1);`

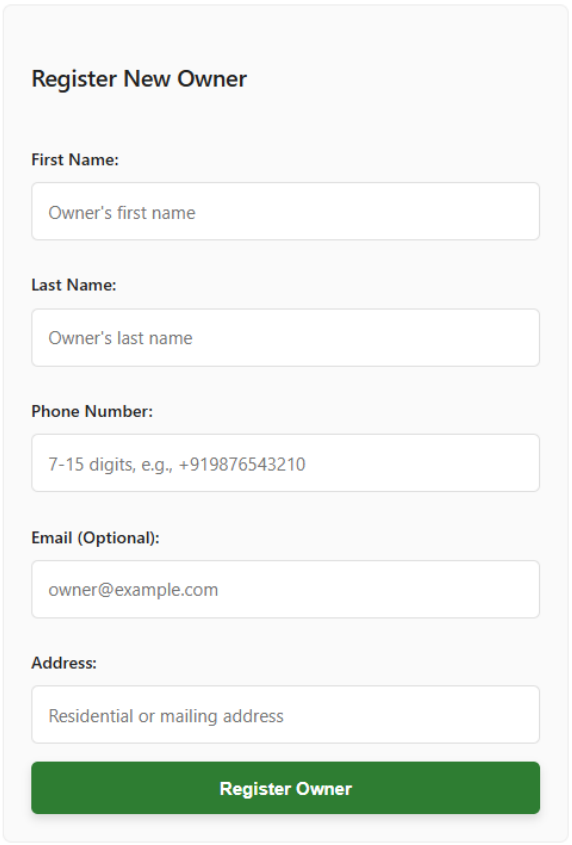
Retrieves complete pet history with owner and vet information

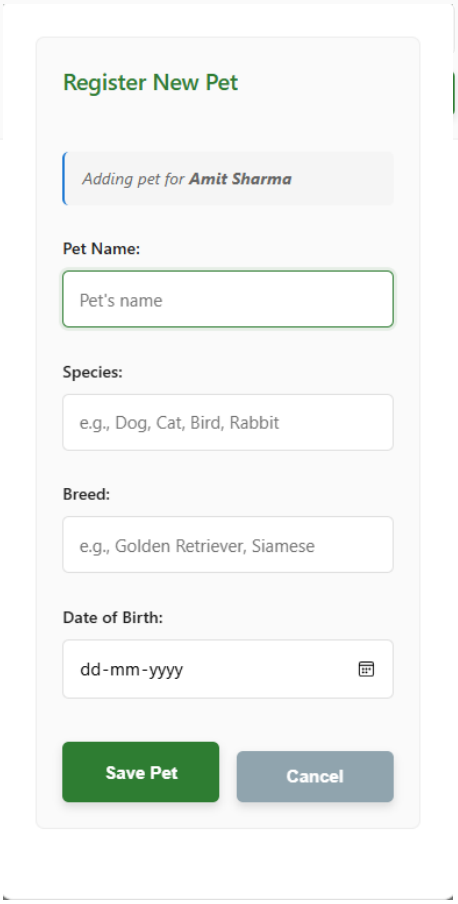

Pet Clinic Management System
Pet Medical History
Pet: Buddy (Dog / Golden Retriever)
DOB: 10/5/2021
Owner: Anjali Patel — 9876543210
Appointment Records
Date
Reason for Visit
Veterinarian
5/11/2025
regular checkup
Dr. Mehra
10/10/2025
Annual Checkup
Dr. Mehra
Close

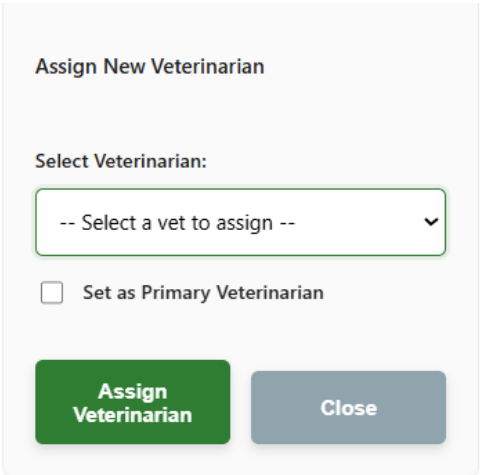
CRUD OPERATOINS SCREENSHOTS

1. CREATE OPERATIONS

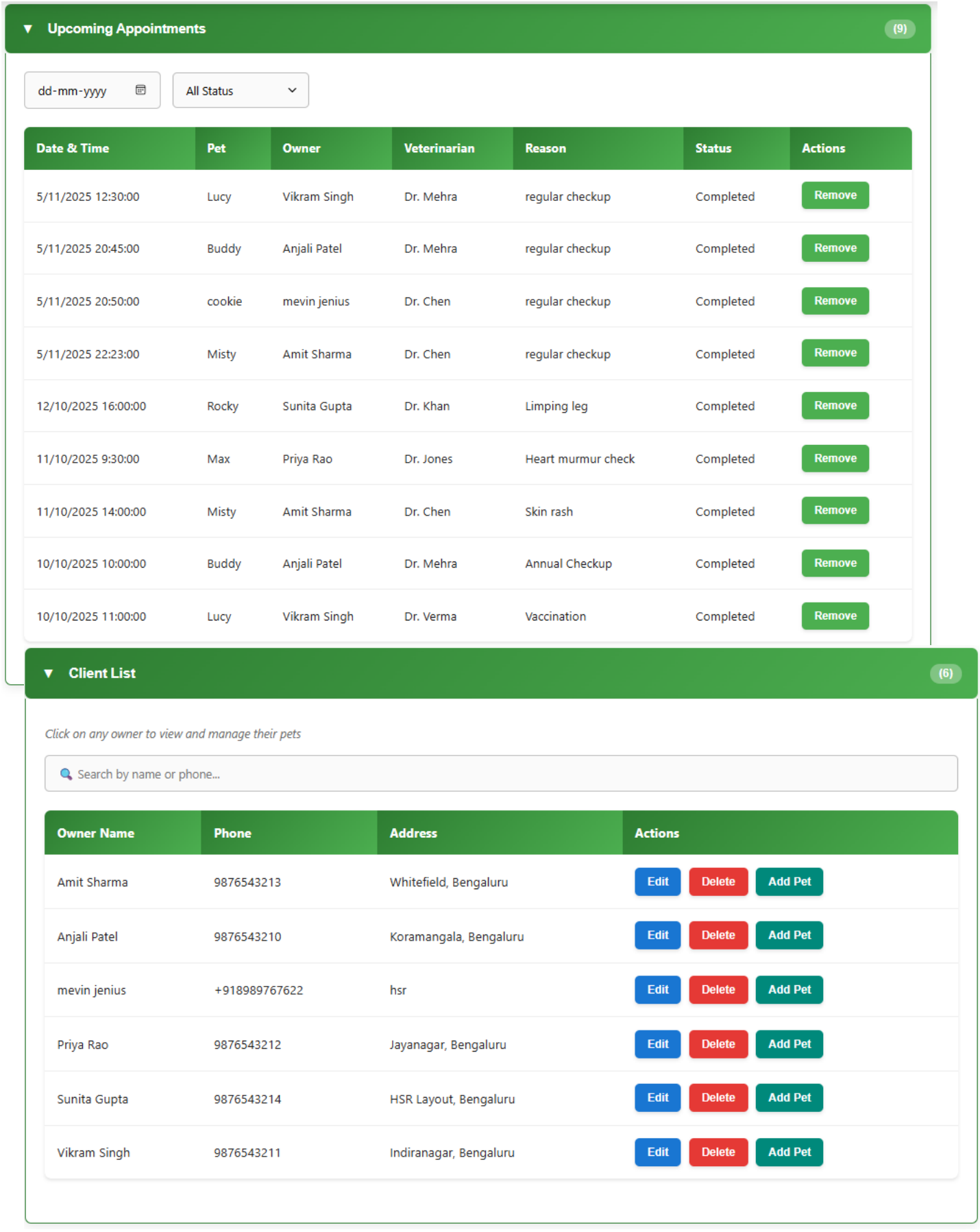








2. READ OPERATIONS



▼ Upcoming Appointments (9)

dd-mm-yyyy | All Status

Date & Time	Pet	Owner	Veterinarian	Reason	Status	Actions
5/11/2025 12:30:00	Lucy	Vikram Singh	Dr. Mehra	regular checkup	Completed	Remove
5/11/2025 20:45:00	Buddy	Anjali Patel	Dr. Mehra	regular checkup	Completed	Remove
5/11/2025 20:50:00	cookie	mevin jenius	Dr. Chen	regular checkup	Completed	Remove
5/11/2025 22:23:00	Misty	Amit Sharma	Dr. Chen	regular checkup	Completed	Remove
12/10/2025 16:00:00	Rocky	Sunita Gupta	Dr. Khan	Limping leg	Completed	Remove
11/10/2025 9:30:00	Max	Priya Rao	Dr. Jones	Heart murmur check	Completed	Remove
11/10/2025 14:00:00	Misty	Amit Sharma	Dr. Chen	Skin rash	Completed	Remove
10/10/2025 10:00:00	Buddy	Anjali Patel	Dr. Mehra	Annual Checkup	Completed	Remove
10/10/2025 11:00:00	Lucy	Vikram Singh	Dr. Verma	Vaccination	Completed	Remove

▼ Client List (6)

Click on any owner to view and manage their pets

Search by name or phone...

Owner Name	Phone	Address	Actions
Amit Sharma	9876543213	Whitefield, Bengaluru	Edit Delete Add Pet
Anjali Patel	9876543210	Koramangala, Bengaluru	Edit Delete Add Pet
mevin jenius	+918989767622	hsr	Edit Delete Add Pet
Priya Rao	9876543212	Jayanagar, Bengaluru	Edit Delete Add Pet
Sunita Gupta	9876543214	HSR Layout, Bengaluru	Edit Delete Add Pet
Vikram Singh	9876543211	Indiranagar, Bengaluru	Edit Delete Add Pet

▼ Outstanding Bills (9)

Manage and track all billing records for appointments

All Bills

Bill ID	Date	Client Name	Pet Name	Amount	Status	Payment Mode	Actions
1762334147	5/11/2025	Amit Sharma	Misty	₹50.00	Paid	Cash	
1762334146	5/11/2025	Anjali Patel	Buddy	₹40.00	Paid	Cash	
1762334144	5/11/2025	mevin jenius	cookie	₹100.00	Paid	Cash	
1762334145	5/11/2025	Vikram Singh	Lucy	₹100.00	Paid	Cash	
105	12/10/2025	Sunita Gupta	Rocky	₹1200.00	Paid	Cash	
104	11/10/2025	Amit Sharma	Misty	₹900.00	Paid	Card	
103	11/10/2025	Priya Rao	Max	₹1500.00	Paid	Card	
102	10/10/2025	Vikram Singh	Lucy	₹750.00	Paid	Cash	
101	10/10/2025	Anjali Patel	Buddy	₹500.00	Paid	Card	

▼ Pets for Anjali Patel (0)

Manage and view details of all pets owned by this client

All Species

Pet Name	Species	Breed	Date of Birth	Age	Actions
Buddy	Dog	Golden Retriever	10/5/2021	4	View History Manage Vets Edit Delete
brownie	Dog	labarador	1/9/2023	2	View History Manage Vets Edit Delete

Pet Medical History

Pet: Buddy (Dog / Golden Retriever)
DOB: 10/5/2021
Owner: Anjali Patel — 9876543210

Appointment Records

Date	Reason for Visit	Veterinarian
5/11/2025	regular checkup	Dr. Mehra
10/10/2025	Annual Checkup	Dr. Mehra

Close

💰 Total Bills Per Owner

View billing summary with payment breakdown for each owner

Owner ID	Owner Name	Total Bills	Total Amount	Paid Amount	Unpaid Amount
5	Sunita Gupta	1	₹1200.00	₹1200.00	₹0.00
8	mevin jenius	1	₹100.00	₹100.00	₹0.00
1	Anjali Patel	1	₹40.00	₹40.00	₹0.00

🐾 Pet Demographics

Pet count by species with average age calculations

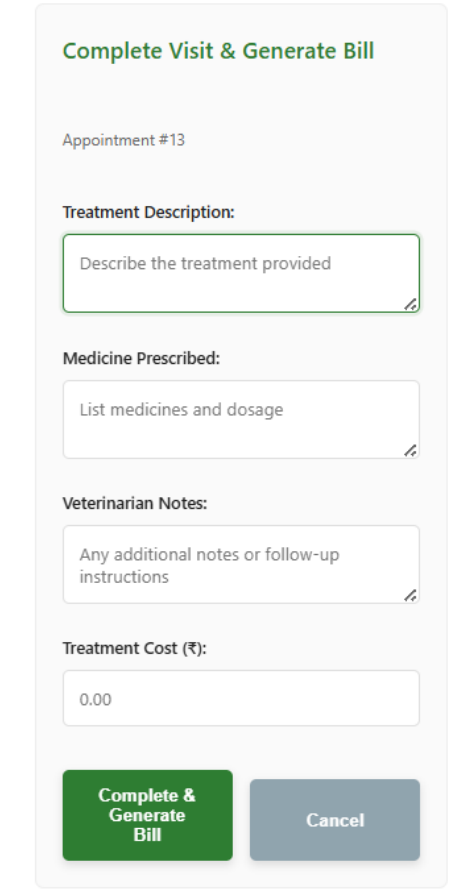
Species	Total Pets	Average Age (Years)

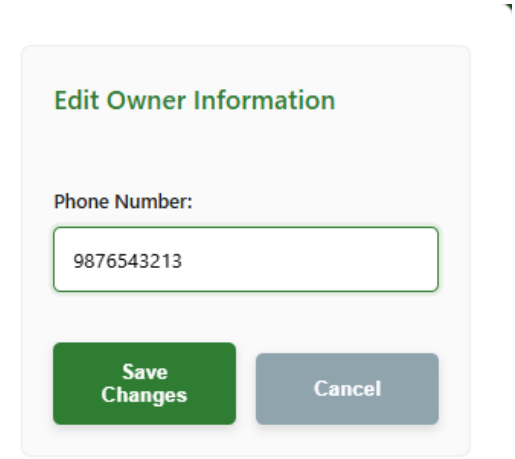
👩‍⚕️ Veterinarian Performance

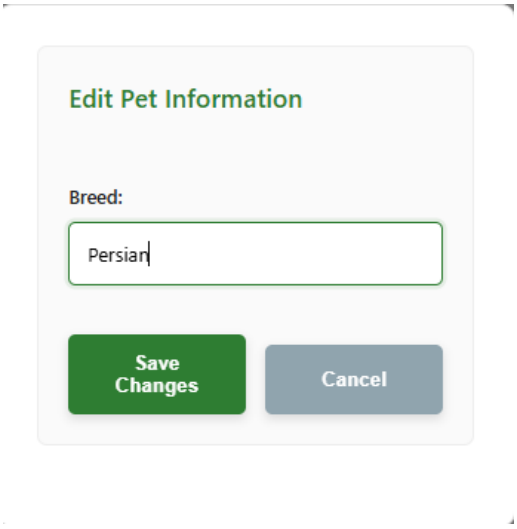
Vet workload analysis with revenue metrics

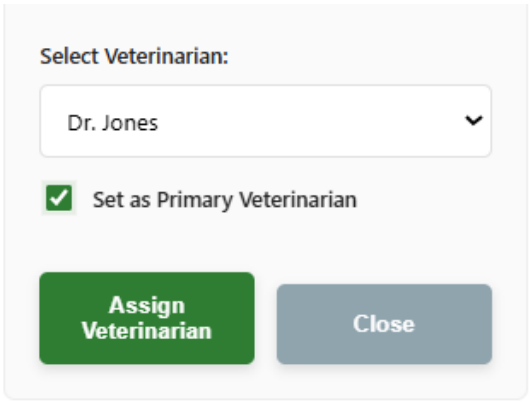
Veterinarian	Specialization	Total Appointments	Total Revenue	Average Bill Amount
Dr. Mehra	Surgery	2	₹40.00	₹40.00
Dr. Chen	Dermatology	1	₹100.00	₹100.00
Dr. Khan	General Practice	1	₹1200.00	₹1200.00
Dr. Verma	General Practice	0	₹0.00	₹0.00
Dr. Jones	Cardiology	0	₹0.00	₹0.00

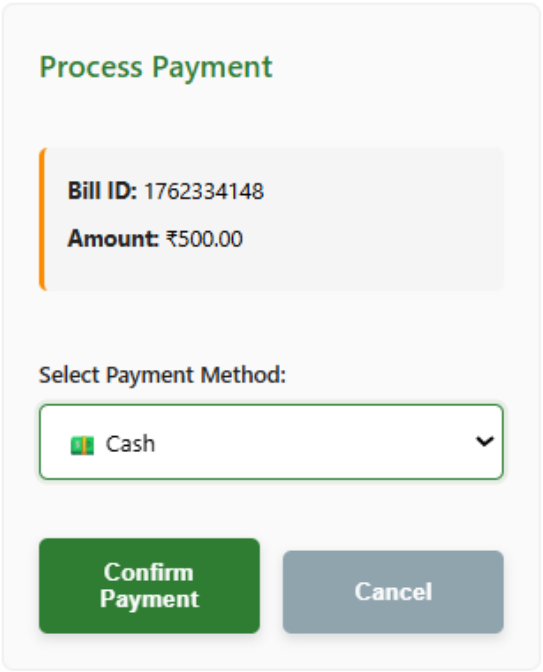
3. UPDATE Operations

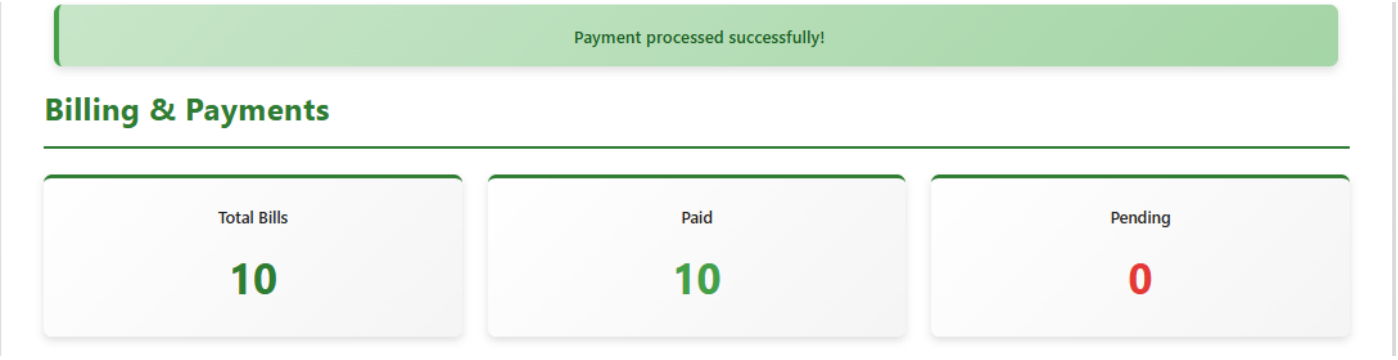




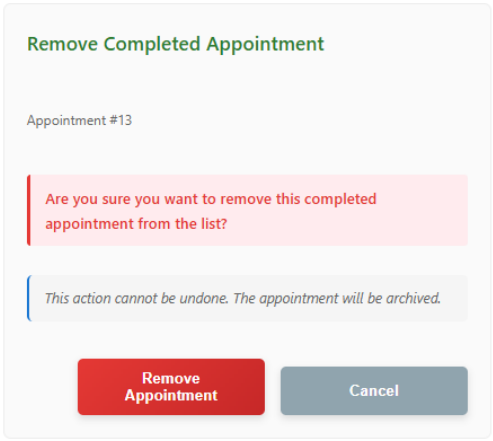


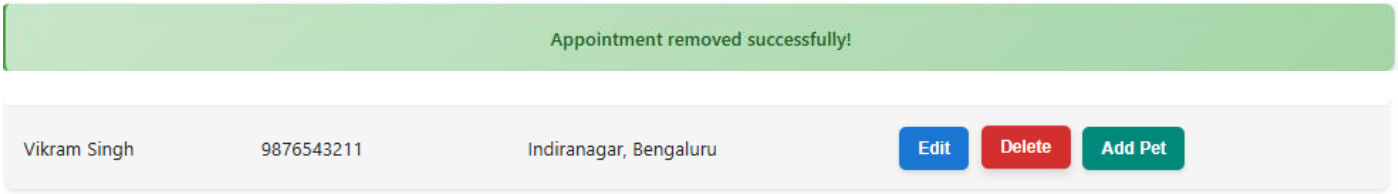

Select Veterinarian:
Dr. Jones
Set as Primary Veterinarian
Assign Veterinarian
Close

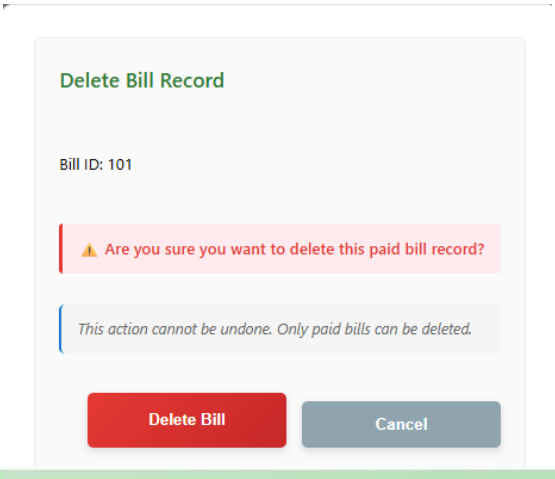

Process Payment
Bill ID: 1762334148
Amount: ₹500.00
Select Payment Method:
Cash
Confirm Payment
Cancel


Payment processed successfully!
Billing & Payments
Total Bills
10
Paid
10
Pending
0

4. DELETE Operations







Features Of The Application

1. Dashboard

- ✓ Central hub displaying system overview
- ✓ Quick access to all major functions
- ✓ Real-time statistics display
- ✓ Key metrics at a glance

2. Owner Management

- ✓ Add, view, edit, and delete pet owners
- ✓ Store multiple email addresses per owner
- ✓ Track owner contact information and delivery addresses
- ✓ View owner's complete pet collection
- ✓ Cascade delete owners with all related pets, appointments, and billing

3. Pet Management

- ✓ Register and manage multiple pets per owner
- ✓ Track pet details (name, species, breed, date of birth)
- ✓ Automatic age calculation from date of birth
- ✓ Assign primary veterinarian to each pet
- ✓ View complete medical history for each pet
- ✓ Delete pets with integrity protection

4. Veterinarian Management

- ✓ Register veterinarians with specializations
- ✓ Track years of experience

- ✓ Assign veterinarians to multiple pets
- ✓ Mark primary veterinarian for each pet
- ✓ View vet workload and statistics
- ✓ Delete vets with constraint validation

5. Appointment Scheduling

- ✓ Book appointments with automatic date validation (no past bookings)
- ✓ Automatic primary vet assignment when pet is selected
- ✓ View appointment schedule in collapsible sections
- ✓ Track appointment status (Scheduled, Completed, Canceled)
- ✓ Collapsible appointment section with dynamic count
- ✓ Complete appointment with treatment details
- ✓ Automatic billing generation on completion
- ✓ Remove completed appointments with confirmation

6. Treatment & Medical Records

- ✓ Document treatment details (description, medicine, notes)
- ✓ Track treatment costs
- ✓ Maintain complete medical history per pet
- ✓ Link treatments to specific appointments
- ✓ View treatment-to-billing workflow

7. Billing & Payment Management

- ✓ Automatic Bill ID generation with AUTO_INCREMENT
- ✓ Track payment status (Paid, Unpaid)
- ✓ Support multiple payment modes (Cash, Card, etc.)

- ✓ Collapsible billing section with count and filter options
- ✓ Separate views for pending and paid bills
- ✓ Auto-refresh billing stats after payment
- ✓ Calculate outstanding bills
- ✓ Payment mode tracking

8. Advanced Search & Filtering

- ✓ Search owners by name or phone
- ✓ Filter pets by species or owner
- ✓ Filter appointments by date or status
- ✓ Filter bills by payment status

9. Advanced Queries & Reports

- ✓ Multi-table JOIN queries (Pet-Owner-Appointment-Vet joins)
- ✓ Nested queries for complex filtering
- ✓ Billing Summary Report - Total bills per owner with payment breakdown
- ✓ Pet Demographics Report - Pet count by species with average age
- ✓ Veterinarian Performance Report - Workload analysis with revenue metrics

10. Business Reports

- ✓ Customer (Owner) Summary – Track appointments per owner
- ✓ Veterinarian Performance Metrics – Revenue and appointment analytics
- ✓ Pet Statistics – Population by species and breed
- ✓ Billing Analytics – Outstanding payments and revenue trends
- ✓ Treatment History – Complete medical records with costs

11. User Interface

- ✓ Left Sidebar Navigation – Always-accessible menu with hierarchical organization
- ✓ 2-Column Responsive Layouts – Side-by-side form and content display
- ✓ Collapsible Sections – Toggle tables to reduce visual clutter
- ✓ Professional Design – Gradient backgrounds, smooth animations
- ✓ Responsive Layout – Works on desktop and mobile devices
- ✓ Intuitive Forms – Clean, organized data entry interfaces with validation
- ✓ Data Tables – Organized data display with action buttons
- ✓ Visual Feedback – Color-coded alerts (success, error, warning)
- ✓ Real-time Updates – Dynamic content refresh after operations
- ✓ Smooth Animations – Hover effects, transitions, and bounce animations

12. Data Integrity & Error Handling

- ✓ Foreign key constraint validation
- ✓ Cascade delete protection with helpful error messages
- ✓ Transaction rollback on failures
- ✓ Graceful error handling with user-friendly messages
- ✓ Prevention of orphaned records
- ✓ Duplicate prevention (unique constraints)
- ✓ Date validation for appointments
- ✓ Business logic validation through triggers

13. Advanced Database Features

- ✓ Relational database design normalized to 3NF
- ✓ Automated triggers for business logic

- ✓ Stored procedures for complex operations
- ✓ User-defined functions for calculations
- ✓ Referential integrity constraints
- ✓ Proper indexing for performance
- ✓ Composite keys for many-to-many relationships
- ✓ CASCADE DELETE for data consistency

Database Features Implementation

Triggers (1)

- ✓ before_appointment_insert – Prevents booking appointments in the past

Stored Procedures (1)

- ✓ GetPetHistory(p_PetID) – Retrieves complete pet history with 4-table join

Functions (1)

- ✓ CalculateAge(p_DoB) – Calculates pet age in years

JOIN Queries (2)

- ✓ Pet Treatment Details – 7-table join showing complete treatment and billing info
- ✓ Pets and Primary Veterinarian – Many-to-many relationship with composite key

Nested Queries (2)

- ✓ Owners with Unpaid Bills – 4-level nested subquery
- ✓ Pets Treated by Specific Vet – Multi-level filtering with EXISTS/IN

Aggregate Queries (3)

- ✓ Total Bills Per Owner – COUNT, SUM with conditional logic
- ✓ Pet Count by Species– COUNT, SUM with date calculations

AGGREGATE QUERIES

Query 1: Total Bills Per Owner with Payment Status

```
SELECT
    o.OwnerID,
    o.FirstName,
    o.LastName,
    COUNT(b.BillID) as TotalBills,
    SUM(b.Amount) as TotalAmount,
    SUM(CASE WHEN b.Status = 'Paid' THEN b.Amount ELSE 0
END) as PaidAmount,
    SUM(CASE WHEN b.Status = 'Unpaid' THEN b.Amount ELSE 0
END) as UnpaidAmount
FROM Owner o
LEFT JOIN Pet p ON o.OwnerID = p.OwnerID
LEFT JOIN Appointment a ON p.PetID = a.PetID
LEFT JOIN Billing b ON a.AppID = b.AppID
GROUP BY o.OwnerID, o.FirstName, o.LastName
ORDER BY TotalAmount DESC;
```

Aggregate Functions: COUNT, SUM (3 variants)

Query 2: Pet Count by Species with Average Age

```
SELECT
    Species,
    COUNT(*) as PetCount,
    AVG(TIMESTAMPDIFF(YEAR, DoB, CURDATE())) as AvgAge
FROM Pet
GROUP BY Species
ORDER BY PetCount DESC;
```

Aggregate Functions: COUNT, AVG

Query 3: Veterinarian Workload Analysis

```
SELECT
    v.VetID,
    v.Name,
    v.Specialization,
    COUNT(a.AppID) as TotalAppointments,
    SUM(b.Amount) as TotalRevenue,
    AVG(b.Amount) as AvgBillAmount
FROM Veterinarian v
LEFT JOIN Appointment a ON v.VetID = a.VetID
LEFT JOIN Billing b ON a.AppID = b.AppID
GROUP BY v.VetID, v.Name, v.Specialization
ORDER BY TotalAppointments DESC;
```

API Endpoints (33 Total)

Owner Management (6)

- ✓ `GET /api/owners` – Fetch all owners
- ✓ `POST /api/owners` – Create new owner
- ✓ `GET /api/owners/<id>` – Get owner details
- ✓ `PUT /api/owners/<id>` – Update owner
- ✓ `DELETE /api/owners/<id>` – Delete owner
- ✓ `GET /api/owners/<id>/pets` – Get owner's pets

Pet Management (5)

- ✓ `GET /api/pets` – Fetch all pets
- ✓ `POST /api/pets` – Create new pet
- ✓ `GET /api/pets/<id>` – Get pet details
- ✓ `PUT /api/pets/<id>` – Update pet
- ✓ `DELETE /api/pets/<id>` – Delete pet

Appointment Management (6)

- ✓ `GET /api/appointments` – Fetch all appointments
- ✓ `POST /api/appointments` – Book appointment
- ✓ `GET /api/appointments/<id>` – Get appointment details
- ✓ `POST /api/appointments/<id>/complete` – Mark completed
- ✓ `DELETE /api/appointments/<id>` – Remove appointment
- ✓ `GET /api/pets/<id>/vets` – Get pet's vets

Veterinarian Management (4)

✓ `GET /api/veterinarians` – Fetch all vets

✓ `POST /api/veterinarians` – Register vet

✓ `PUT /api/veterinarians/<id>` – Update vet

✓ `DELETE /api/veterinarians/<id>` – Delete vet

Vet-Pet Assignment (2)

✓ `POST /api/pets/<pet_id>/vets/<vet_id>` – Assign vet to pet

✓ `DELETE /api/pets/<pet_id>/vets/<vet_id>` – Remove vet assignment

Billing Management (5)

✓ `GET /api/billing` – Fetch all bills

✓ `GET /api/billing/<id>` – Get bill details

✓ `PUT /api/billing/<app_id>/<bill_id>/pay` – Record payment

✓ `GET /api/billing/stats` – Get billing statistics

✓ `DELETE /api/billing/<id>` – Delete bill

Treatment Records (3)

✓ `POST /api/treatment` – Create treatment record

✓ `GET /api/treatment/<app_id>` – Get treatment details

✓ `DELETE /api/treatment/<id>` – Delete treatment

Additional (2)

✓ `GET /` – Render home page

✓ `GET /owners` – Render owners management page

GitHub Repository :

rajeev8008/pet-clinic-management-system: DBMS mini-project which is a management system for a pet clinic database.