**Pawsera – Smart Pet Care Management Requirements Document**

# Table of Contents

1. Introduction and Rationale
2. Content Evaluation
   1. Navigation
   2. Layout / Wireframes
   3. Content Descriptions and Sources
   4. Content-types / Real-world Entities
3. Functionality (User Stories)
   1. Must Have User Stories
   2. Should Have User Stories
   3. Nice to Have User Stories
   4. Hourly Estimates
4. Technical Specifications
   1. Technology Stack
   2. Hosting Information
   3. Other Third-Party Software
5. Data Design
   1. Data Flow Diagrams
   2. Database Schemas
6. Development Specifications
   1. Folder Structure
   2. File-naming Conventions
   3. Accessibility Standards
   4. Responsiveness Requirements
   5. GitHub Links
7. Schedule (Timeline Assignment)
8. Conclusion

# Introduction and Rationale

Pawsera is a smart pet care management platform designed for pet owners and veterinarians. Its purpose is to make pet care organized, accessible, and user-friendly. The platform allows pet owners to create pet profiles, track vaccinations, medical history, and appointments. Veterinarians can update medical records and share care recommendations. The system integrates external APIs for weather-based reminders and

Google Maps for locating nearby vets and pet services. The goal is to showcase professional-level Web Development capabilities, with emphasis on accessibility, responsiveness, and sophisticated functionality.

# Content Evaluation

## Navigation

* + - **Pet Owner Navigation:** Dashboard → Pet Profiles → Vaccinations → Appointments → Nearby Vets → Settings
    - **Veterinarian Navigation:** Dashboard → Pet Records → Care Notes → Analytics →

Settings

* + - **Admin Navigation:** Dashboard → User Management → Vet Approvals → Activity Logs

→ Settings

## Layout / Wireframes

**Wireframes Overview:** - **Week 2:** Initial sketches for Login, Dashboard, Pet Profile, Vet Dashboard, Admin Dashboard. - **Week 3:** Detailed Figma prototypes with full layouts for each screen, including forms, data tables, and navigation menus. - Wireframes are structured for responsiveness and accessibility.

**Pet Owner**

After creating an account, the **Pet Owner** can access the following sections displayed as widgets on the dashboard:

* **Dashboard:** Provides a quick overview of activities and key updates within the portal.
* **My Pets:** Displays all registered pets and allows users to add new pets.
* **Appointments:** Enables booking and managing appointments with veterinarians.
* **Nearby Vets:** Shows nearby veterinary clinics through an integrated map.
* **Settings:** Allows users to update and manage their profile and account preferences.

A screenshot of a login screen

AI-generated content may be incorrect. A screenshot of a phone

AI-generated content may be incorrect. A screenshot of a phone

AI-generated content may be incorrect.

Log in page Pet Owner – Dashboard Pet Owner can add pets – My Pet

A screenshot of a phone

AI-generated content may be incorrect. A screenshot of a calendar

AI-generated content may be incorrect. A screenshot of a appointment

AI-generated content may be incorrect.  
Add a new Pet – My Pets Schedule appointment – Pet Owner Appointment confirmation – Pet Owner A screenshot of a phone

AI-generated content may be incorrect. A screenshot of a phone

AI-generated content may be incorrect.

Nearby Vets – Pet Owner Settings – Pet Owner

**Vet and Admin view  
  
Vet and Admin Access Overview**

* **Vet:**  
  The veterinarian can view all **appointments** and **pet records**, as well as manage their **profile settings** within the portal.
* **Admin:**  
  The administrator has **full access** to the portal. They can **add or remove veterinarians**, manage user accounts, and oversee all system activities to ensure smooth operation and maintenance.

A screenshot of a phone

AI-generated content may be incorrect. A screenshot of a phone

AI-generated content may be incorrect.

Vet Dashboard Admin Dashboard

## Content Descriptions and Sources

* + - **Text Content:** Platform instructions, notifications, and help text – authored in- house.
    - **Images:** Pet photos uploaded by users, vet logos (with permissions), icons from open-source libraries.
    - **External Data Sources:** OpenWeather API (weather data), Google Maps API (vet C pet store locations).

## Content-types / Real-world Entities

* + - **Pet:** Name, breed, age, photo, vaccination records, medical history, milestones.
    - **Pet Owner:** Name, email, password, pet list.
    - **Veterinarian:** Name, email, password, patients list, care notes.
    - **Admin:** Name, email, password, system oversight functions.
    - **Appointment:** Pet, Vet, Date, Time, Purpose.
    - **Document:** File uploads such as vaccination certificates.

# Functionality (User Stories)

## Must Have User Stories

**Pet Owner**

* + - As a pet owner, I want to create an account so that I can securely manage my pet’s

information.

* + - As a pet owner, I want to log in to my account so that I can access my pet profiles and appointments.
    - As a pet owner, I want to create and edit pet profiles so that I can track my pets’ details,

vaccinations, and medical history.

* + - As a pet owner, I want to schedule an appointment with a veterinarian so that I can ensure my pet receives timely care.

**Veterinarian**

* + - As a veterinarian, I want to log in to the platform so that I can securely access my patients’

records.

* + - As a veterinarian, I want to update pet medical records so that I can keep treatment history accurate.
    - As a veterinarian, I want to add care notes after appointments so that pet owners have updated guidance.

**Admin**

* + - As an admin, I want to approve veterinarian accounts so that only verified professionals can provide care.
    - As an admin, I want to manage user accounts so that I can maintain system integrity and security.

## Should Have User Stories

**Pet Owner**

* + - As a pet owner, I want to receive vaccination reminders so that I do not miss important health updates for my pet.
    - As a pet owner, I want to search for nearby veterinarians using Google Maps so that I can conveniently locate pet care services.
    - As a pet owner, I want to upload vaccination certificates and medical documents so that I can keep all records in one place.

**Veterinarian**

* + - As a veterinarian, I want to view analytics of pet health trends so that I can better track patient outcomes.
    - As a veterinarian, I want to send care recommendations directly to pet owners so that they can follow treatment plans at home.

**Admin**

* + - As an admin, I want to monitor activity logs so that I can ensure appropriate system usage.

## Nice to Have User Stories

**Pet Owner**

* + - As a pet owner, I want to track pet milestones (e.g., birthdays, weight progress) so that I can celebrate and monitor growth.
    - As a pet owner, I want to receive weather-based reminders (e.g., avoid long walks in extreme heat) so that I can take better care of my pet.
    - As a pet owner, I want to receive in-app notifications for upcoming appointments so that I

don’t miss them.

**Veterinarian**

* + - As a veterinarian, I want to upload educational resources so that pet owners can learn about preventive care.

**Admin**

* + - As an admin, I want to generate system-wide reports so that I can evaluate platform performance and usage trends.

# Technical Specifications

## Technology Stack

* + - Frontend: React (with HTML C CSS), deployed on Firebase Hosting or Vercel
    - **Backend/Database:** Firebase Firestore
    - **Authentication:** Firebase Authentication
    - **External APIs:** Google Maps API, OpenWeather API
    - **Version Control:** GitHub
    - **Design s Prototyping:** Figma

## Hosting Information

* + - Frontend: Firebase Hosting (preferred) or Vercel
    - **Backend / Database:** Firebase Cloud Hosting C Firestore

## Other Third-Party Software

* + - Firebase SDK (latest stable)
    - Google Maps JavaScript API
    - OpenWeather API (latest version)
    - Figma (for prototypes)

# Data Design

## Data Flow Diagrams

* + - **Pet Owner Registration/Login → Dashboard → Pet Profile CRUD → Appointment**

**Tracking**

* + - **Veterinarian Login → Pet Records → Care Notes → Analytics**
    - **Admin Login → User Management → Vet Approvals → Activity Monitoring**

A flowchart of a computer program

AI-generated content may be incorrect.A flowchart of a pet

AI-generated content may be incorrect.

Verify

Settings

Nearby Vets

Book Appointments

Add pets

View my Pets

Pet Owner Dashboard

Update Record

Save Record

Check Pet

## Database Schemas

* + - **Users Table:** userID, name, email, passwordHash, role (Pet Owner / Vet / Admin)
    - **Pets Table:** petID, ownerID, name, breed, age, photo, vaccinationRecords, milestones
    - **Appointments Table:** appointmentID, petID, vetID, date, time, purpose
    - **Documents Table:** documentID, petID, fileName, fileURL, uploadDate
    - **Care Notes Table:** noteID, petID, vetID, date, noteText

# Development Specifications

## Folder Structure

/src

/components

/pages

/assets

/api

/utils

/styles

/public

## File-naming Conventions

* + - CamelCase for React components: PetProfile.js, VetDashboard.js
    - kebab-case for CSS/SCSS files: dashboard-styles.css
    - Constants in UPPER\_SNAKE\_CASE

## Accessibility Standards

* + - Semantic HTML
    - Keyboard navigable components
    - ARIA labels for forms and dynamic content
    - Sufficient color contrast

## Responsiveness Requirements

* + - Mobile-first design
    - Responsive grid for dashboards and pet profiles
    - Media queries for tablets and desktops

## GitHub Links

* + - Repository: https://github.com/RaginiS19/Pawsera
    - Branching strategy: main for production, dev for ongoing development, feature branches for new functionalities

# Schedule (Timeline Assignment)

**Week 2:** - Sketch initial wireframes for Login, Pet Profile, Vet Dashboard, Admin Dashboard - Define MVP features and user stories

**Week 3:** - Develop Figma prototypes for all screens - Refine layouts, forms, and navigation

**Week 4:** - Set up Firebase project, Authentication, Firestore schema - Implement registration/login system - Start CRUD for pet profiles

**Week 5:** - Implement vaccination C medical history tracking - Appointment booking system - Basic Admin dashboard functions

**Week 6:** - Integrate Google Maps API for nearby vet search - Weather-based reminders using OpenWeather API - Vet dashboard: care notes functionality

**Week 7:** - Implement Should Have features: Vet Analytics Dashboard - Nice to Have: Upload documents, pet milestones tracker, in-app notifications - Testing for accessibility, responsiveness, and functionality

**Week 8:** - Final testing C debugging - Content population (images, text, API data) - Prepare deployment

**Week G:** - Deployment on GitHub Pages and Firebase hosting - Final review, documentation, and submission

# Conclusion

Pawsera is designed to provide a seamless and professional experience for pet owners and veterinarians. With a clear requirements document, detailed user stories, technical specifications, and a structured timeline, this project demonstrates advanced web development skills including full-stack implementation, API integration, accessibility compliance, and responsive design. The final product will showcase the culmination of the Web Development program knowledge and practical capabilities.

*Document prepared by Ragini Shirwalkar*