

Product Requirements Document (PRD)

CIDM 6325 – Weeks 3–4 Unified Assignment

Product: Pet Care Basics Web Application

Author: [Tan Vu / GitHub Username: boyhamgirl]

Date: September 2025

Version: 0.1 Draft

Abstract

This document defines the requirements for *Pet Care Basics*, a multi-page web application designed to provide dog and cat owners with accessible, beginner-friendly care information. Building on a CRUD-style foundation, this iteration introduces authentication, structured content workflows, and form handling to create a more interactive and portfolio-ready product. The PRD captures the project's purpose, goals, scope, functional and non-functional requirements, and success metrics while ensuring traceability to broader course outcomes. The document follows professional standards, integrates lessons from Matt Layman's *Understand Django* (Chapters 3–4), and explicitly documents AI-assisted contributions.

1. Introduction

- **Vision:** Create a simple, intuitive, and secure platform where pet owners can learn basic care, log in, and submit care inquiries.
 - **Problem Statement:** Current pet care resources online are fragmented, ad-heavy, and sometimes inaccurate. Beginners need a trusted, lightweight, and accessible web app.
 - **Goals:**
 - Provide well-structured educational pages for dog washing and cat care.
 - Allow authenticated users to submit questions via a form.
 - Offer consistent navigation and design across devices.
 - **Link to Strategy:** Builds web development and Django skills; extends CRUD knowledge into authentication, templates, and workflows per CIDM 6325.
-

2. Objectives & Success Metrics

- **Objectives:**
 - Implement login/logout and user registration.
 - Extend content beyond static HTML to dynamic templates.
 - Collect user form data (e.g., care questions, feedback).
 - **Success Metrics:**
 - $\geq 80\%$ task completion rate in usability test (finding content, submitting form).
 - $\geq 90\%$ CSS/HTML validation compliance.
 - Authentication response within 1s for 95% of cases.
-

3. Scope

- **In Scope:**
 - Authentication (login, logout, registration).
 - Multi-page navigation (Home, Dog Care, Cat Care).
 - Form to capture at least 3 inputs (pet type, question, email).
 - External CSS with max-width layout, justified paragraphs, styled headers.
 - **Out of Scope:**
 - Mobile app version.
 - Advanced vet-integrated APIs.
 - Payment or monetization features.
-

4. User Stories & Use Cases

- **User Persona:** New pet owner seeking basic care guidance.
- **User Stories:**
 - As a pet owner, I want to log in, so I can access personalized content.

- As a visitor, I want to submit a pet care question, so I can get guidance.
 - As a student developer, I want clear templates, so I can demonstrate reusable UI.
 - **Use Cases:**
 - *Happy Path:* User signs up → logs in → navigates to Cat Care page → submits a form.
 - *Edge Case:* User enters invalid credentials → system displays error with retry option.
-

5. Functional Requirements

- **FR-001:** Users can create accounts with username, email, password.
 - **FR-002:** Authenticated users can log in and log out.
 - **FR-003:** Site navigation is consistent across all pages.
 - **FR-004:** Form submissions validate required fields before acceptance.
 - **FR-005:** Content (dog/cat care) uses templates for reusable layout.
-

6. Non-Functional Requirements

- **Performance:** Login response < 1s at p95.
 - **Scalability:** Up to 50 concurrent users (class demo scale).
 - **Accessibility:** WCAG 2.1 AA compliance; alt text on images.
 - **Security:** Passwords hashed; CSRF protection enabled (per Django defaults).
 - **Reliability:** ≥ 99% uptime during class demonstration.
-

7. Dependencies & Risks

- **Dependencies:** Django authentication framework; external CSS stylesheet.
- **Risks:**
 - Scope creep into non-required features.

- Misconfigured authentication → user frustration.
 - **Mitigation:** Stick to MVP; peer review authentication flows.
-

8. Acceptance Criteria

- **AC-001:** User can register and then immediately log in.
 - **AC-002:** User sees consistent navigation on Home, Dog Care, and Cat Care pages.
 - **AC-003:** Form requires 3 inputs; submits successfully; errors handled gracefully.
 - **AC-004:** Pages validate with W3C HTML/CSS checkers.
-

9. Success Metrics

- Authentication working $\geq 95\%$ in test cases.
 - Navigation consistency rated $\geq 4/5$ in peer review.
 - $\geq 80\%$ of test users complete a form without error on first attempt.
-

10. Rollout & Release Plan

- **Phase 1:** Static HTML + external CSS (already implemented).
 - **Phase 2:** Add Django templates and authentication.
 - **Phase 3:** Implement and test form submission logic.
 - **Training:** Developer README + Loom walkthrough.
-

11. Open Questions

- Should care form responses be emailed to admin, or stored locally only?
 - Do we restrict content (Dog/Cat Care pages) to authenticated users, or keep them public?
-

12. References

- Layman, M. (2023). *Understand Django*, Chapters 3–4 (*Views on Views, Templates for User Interfaces*).
- W3C. (2024). *Web Content Accessibility Guidelines (WCAG) 2.1*.