



I need very detailed specification document for MVP version for Pawzly.

with all logic, user flows, specifications, Platform specification.

Pet owner:

pet profile

create pet profile with all details and photos. it should be like detailed page with all details.

Social network for Pets and owners

can add several pets.

messaging

Digital pet passport: where add all details about vacation, pet health, details, chip data, system will notify when need to make new vaccination or add tablets like from worms. store all documents about pet.

Add details about vet clinic or where visit pet.

food,

pet only map:

which will show all places for pets or also places which are dog friendly like cafes or restaurants and POI.

also user can filter, view details of each service also left a review. send enquiry or view business contact details

- view all services around for Pet on map and as a list. with possibility to book service or send request with details. and review reviews, services.
- create a service request for Dog Training, pet sitting and etc.
- help request.

Service provider:

service provider can be any user: Pet owner also just user, also it can be business.

full detailed profile about service or business with all details about services.

manage offers and bookings

communicate with clients

, service finder, post service offer, list on map.

calendar management

booking management and etc.

Shelters and adoption services:

general all details and Pets details page with all details. for adoption and etc.

donation request. with all details. also show all volunteers.

Volunteers

find help request, respond and communicate.

User can can have multiple roles and have additional functionality.

like user can be pet owner, also provide pet sitting and be Business owner.

we will use all functionality of Supabase, Google API, Vercel, React,
with the modern design and architecture.

review whole details from this thread.

Based on the comprehensive analysis of all previous specifications and your detailed requirements, here is the complete **Pawzly MVP Specification Document** with modern architecture using Supabase, Google APIs, Vercel, and React^[1] ^[2] ^[3]:

PAWZLY MVP - COMPLETE SPECIFICATION DOCUMENT

Executive Summary

Pawzly is a comprehensive pet social network and service platform combining pet passport management, location-based services, community features, shelter adoption, and multi-role marketplace functionality. Built with modern stack: React, Supabase, Google Maps API, and Vercel deployment^[1] ^[2] ^[3].

1. TECHNICAL ARCHITECTURE

1.1 Modern Tech Stack

Frontend^[1] ^[2]

- React 18+ with TypeScript
- Next.js 14 (App Router) for SSR/SSG
- TailwindCSS + Shadcn/ui components
- React Query (TanStack Query) for data fetching
- Zustand for global state management
- React Hook Form + Zod for validation
- Framer Motion for animations
- PWA support with next-pwa

Backend & Database^[1] ^[3]

- Supabase (PostgreSQL + Auth + Storage + Realtime)
- Supabase Auth (JWT-based, social OAuth)
- Row Level Security (RLS) policies
- Supabase Storage for images/documents

- Supabase Realtime for chat/notifications
- Edge Functions for serverless logic

APIs & Services^[1] ^[2]

- Google Maps JavaScript API (map display)
- Google Places API (POI, dog-friendly locations)
- Google Geocoding API (address conversion)
- Google Directions API (routing)
- Stripe API (payments, subscriptions)
- Resend (transactional emails)
- Uploadthing (file uploads)
- Sharp (image optimization)

Deployment^[1]

- Vercel (hosting, edge functions, analytics)
- Vercel Edge Config (feature flags)
- Vercel KV (Redis for caching)
- CloudFlare CDN (images, assets)
- GitHub Actions (CI/CD)

Monitoring & Analytics^[1]

- Vercel Analytics
- Sentry (error tracking)
- PostHog (product analytics)
- LogRocket (session replay)

1.2 Database Architecture (Supabase)

Core Tables Structure^[3]

```
-- Enable UUID extension
CREATE EXTENSION IF NOT EXISTS "uuid-oss";

-- USERS TABLE (extends Supabase auth.users)
CREATE TABLE public.profiles (
  id UUID PRIMARY KEY REFERENCES auth.users(id) ON DELETE CASCADE,
  email TEXT UNIQUE NOT NULL,
  full_name TEXT,
  avatar_url TEXT,
  bio TEXT,
  phone TEXT,
  country_code TEXT DEFAULT 'US',
  city TEXT,
  postal_code TEXT,
  latitude DECIMAL(10,8),
  longitude DECIMAL(11,8),
```

```

-- Multi-role support
roles TEXT[] DEFAULT ARRAY['pet_owner'], -- pet_owner, provider, volunteer, business

-- Trust & Verification
trust_score INTEGER DEFAULT 50 CHECK (trust_score >= 0 AND trust_score <= 100),
is_verified BOOLEAN DEFAULT FALSE,
verified_at TIMESTAMPTZ,

-- Preferences
preferred_language TEXT DEFAULT 'en',
preferred_currency TEXT DEFAULT 'USD',
timezone TEXT DEFAULT 'UTC',

-- Privacy
profile_visibility TEXT DEFAULT 'public', -- public, friends, private
show_location BOOLEAN DEFAULT TRUE,
allow_messages BOOLEAN DEFAULT TRUE,

-- Stats
total_pets INTEGER DEFAULT 0,
total_donations DECIMAL(10,2) DEFAULT 0,
total_bookings INTEGER DEFAULT 0,

created_at TIMESTAMPTZ DEFAULT NOW(),
updated_at TIMESTAMPTZ DEFAULT NOW()
);

-- Enable RLS
ALTER TABLE public.profiles ENABLE ROW LEVEL SECURITY;

-- RLS Policies
CREATE POLICY "Public profiles are viewable by everyone"
ON public.profiles FOR SELECT
USING (profile_visibility = 'public' OR auth.uid() = id);

CREATE POLICY "Users can update own profile"
ON public.profiles FOR UPDATE
USING (auth.uid() = id);

```

Pets Table (Digital Passport)^[1] ^[2] ^[3]

```

CREATE TABLE public.pets (
  id UUID PRIMARY KEY DEFAULT uuid_generate_v4(),
  owner_id UUID REFERENCES public.profiles(id) ON DELETE CASCADE,

  -- Basic Info
  name TEXT NOT NULL,
  species TEXT NOT NULL, -- dog, cat, bird, rabbit, other
  breed TEXT,
  is_mixed_breed BOOLEAN DEFAULT FALSE,
  gender TEXT, -- male, female, unknown
  date_of_birth DATE,
  weight DECIMAL(6,2),
  weight_unit TEXT DEFAULT 'kg',
  color TEXT,

```

```

-- Identification
microchip_id TEXT UNIQUE,
registration_number TEXT,

-- Health
is_spayed_neutered BOOLEAN,
spay_neuter_date DATE,
allergies TEXT,
medical_conditions TEXT,
current_medications TEXT,
special_needs TEXT,

-- Behavior
temperament TEXT[],
good_with_children BOOLEAN,
good_with_dogs BOOLEAN,
good_with_cats BOOLEAN,
training_level TEXT, -- none, basic, advanced

-- Emergency
emergency_contact_name TEXT,
emergency_contact_phone TEXT,
preferred_vet_clinic TEXT,
preferred_vet_phone TEXT,

-- Media
primary_photo_url TEXT,
photos JSONB DEFAULT '[]'::jsonb,

-- Social
bio TEXT,
social_media JSONB,
friend_pets UUID[] DEFAULT ARRAY[]::UUID[],

-- Status
status TEXT DEFAULT 'active', -- active, deceased, rehomed, lost

-- Visibility
is_public BOOLEAN DEFAULT TRUE,
show_on_map BOOLEAN DEFAULT TRUE,

created_at TIMESTAMPTZ DEFAULT NOW(),
updated_at TIMESTAMPTZ DEFAULT NOW()
);

CREATE INDEX idx_pets_owner ON public.pets(owner_id);
CREATE INDEX idx_pets_species ON public.pets(species);
CREATE INDEX idx_pets_microchip ON public.pets(microchip_id) WHERE microchip_id IS NOT NULL;

ALTER TABLE public.pets ENABLE ROW LEVEL SECURITY;

CREATE POLICY "Pets viewable by everyone if public"
ON public.pets FOR SELECT
USING (is_public = TRUE OR owner_id = auth.uid());

```

```
CREATE POLICY "Owners can manage their pets"
ON public.pets FOR ALL
USING (owner_id = auth.uid());
```

Pet Health Records (Digital Passport) [\[1\]](#) [\[2\]](#) [\[3\]](#)

```
CREATE TABLE public.pet_health_records (
  id UUID PRIMARY KEY DEFAULT uuid_generate_v4(),
  pet_id UUID REFERENCES public.pets(id) ON DELETE CASCADE,

  record_type TEXT NOT NULL, -- vaccination, checkup, diagnosis, treatment, surgery, lab_
  record_date DATE NOT NULL,
  title TEXT NOT NULL,
  description TEXT,

  -- Vaccination specific
  vaccine_name TEXT,
  vaccine_batch TEXT,
  next_due_date DATE,
  reminder_sent BOOLEAN DEFAULT FALSE,

  -- Medical specific
  diagnosis_code TEXT,
  diagnosis_name TEXT,
  treatment_plan TEXT,
  prescribed_medication TEXT,
  dosage TEXT,

  -- Provider
  veterinarian_name TEXT,
  clinic_name TEXT,
  clinic_id UUID REFERENCES public.profiles(id),

  -- Documents
  document_urls TEXT[],
  verified_by_clinic BOOLEAN DEFAULT FALSE,
  verified_at TIMESTAMPTZ,

  -- Cost
  cost DECIMAL(10,2),
  currency TEXT DEFAULT 'USD',

  -- Reminders
  reminder_date DATE,

  created_at TIMESTAMPTZ DEFAULT NOW(),
  created_by UUID REFERENCES public.profiles(id)
);

CREATE INDEX idx_health_records_pet ON public.pet_health_records(pet_id, record_date DESC);
CREATE INDEX idx_health_records_reminders ON public.pet_health_records(reminder_date)
WHERE reminder_sent = FALSE AND reminder_date IS NOT NULL;

ALTER TABLE public.pet_health_records ENABLE ROW LEVEL SECURITY;
```

```
CREATE POLICY "Owners can view/manage pet health records"
ON public.pet_health_records FOR ALL
USING (
    pet_id IN (SELECT id FROM public.pets WHERE owner_id = auth.uid())
);
```

Pet Social Posts ^[1] ^[2]

```
CREATE TABLE public.pet_posts (
    id UUID PRIMARY KEY DEFAULT uuid_generate_v4(),
    pet_id UUID REFERENCES public.pets(id) ON DELETE CASCADE,
    author_id UUID REFERENCES public.profiles(id) ON DELETE CASCADE,

    content TEXT NOT NULL,
    media JSONB DEFAULT '[]'::jsonb, -- array of {type, url, thumbnail}

    -- Location
    location_name TEXT,
    latitude DECIMAL(10,8),
    longitude DECIMAL(11,8),

    -- Engagement
    like_count INTEGER DEFAULT 0,
    comment_count INTEGER DEFAULT 0,
    share_count INTEGER DEFAULT 0,

    -- Visibility
    visibility TEXT DEFAULT 'public', -- public, friends, private

    -- Tags
    tags TEXT[],
    mentioned_pets UUID[],

    is_pinned BOOLEAN DEFAULT FALSE,
    is_flagged BOOLEAN DEFAULT FALSE,

    created_at TIMESTAMPTZ DEFAULT NOW(),
    updated_at TIMESTAMPTZ DEFAULT NOW()
);

CREATE INDEX idx_pet_posts_pet ON public.pet_posts(pet_id, created_at DESC);
CREATE INDEX idx_pet_posts_author ON public.pet_posts(author_id, created_at DESC);
CREATE INDEX idx_pet_posts_location ON public.pet_posts(latitude, longitude)
    WHERE latitude IS NOT NULL;

ALTER TABLE public.pet_posts ENABLE ROW LEVEL SECURITY;
```

Pet-Friendly Places (Map POI) ^[1] ^[2]

```
CREATE TABLE public.pet_places (
    id UUID PRIMARY KEY DEFAULT uuid_generate_v4(),

    -- Basic Info
    name TEXT NOT NULL,
```

```

place_type TEXT NOT NULL, -- cafe, restaurant, park, hotel, store, vet, grooming, train
description TEXT,

-- Location (Google Places integration)
google_place_id TEXT UNIQUE,
address TEXT NOT NULL,
city TEXT NOT NULL,
country_code TEXT NOT NULL,
postal_code TEXT,
latitude DECIMAL(10,8) NOT NULL,
longitude DECIMAL(11,8) NOT NULL,

-- Contact
phone TEXT,
website TEXT,
email TEXT,

-- Hours
hours_of_operation JSONB,

-- Pet-Friendly Details
allows_dogs BOOLEAN DEFAULT TRUE,
allows_cats BOOLEAN DEFAULT FALSE,
pet_amenities TEXT[], -- water_bowls, treats, play_area, outdoor_seating
pet_size_restrictions TEXT[], -- small, medium, large, all
pet_rules TEXT,

-- Features
features TEXT[], -- parking, wifi, outdoor, indoor, reservations
price_range TEXT, -- $, $$, $$$, $$$$

-- Media
photos JSONB DEFAULT '[]'::jsonb,
cover_photo_url TEXT,

-- Reviews
average_rating DECIMAL(3,2) DEFAULT 0,
review_count INTEGER DEFAULT 0,

-- Verification
is_verified BOOLEAN DEFAULT FALSE,
verified_by UUID REFERENCES public.profiles(id),
verified_at TIMESTAMPTZ,

-- Stats
view_count INTEGER DEFAULT 0,
visit_count INTEGER DEFAULT 0,
bookmark_count INTEGER DEFAULT 0,

-- Status
status TEXT DEFAULT 'active', -- active, pending, inactive

added_by UUID REFERENCES public.profiles(id),
created_at TIMESTAMPTZ DEFAULT NOW(),
updated_at TIMESTAMPTZ DEFAULT NOW()
);

```



```

-- Spatial index for location queries
CREATE INDEX idx_pet_places_location ON public.pet_places
  USING GIST (ll_to_earth(latitude, longitude));

CREATE INDEX idx_pet_places_type ON public.pet_places(place_type, status);
CREATE INDEX idx_pet_places_city ON public.pet_places(city, country_code);
CREATE INDEX idx_pet_places_rating ON public.pet_places(average_rating DESC, review_count);

ALTER TABLE public.pet_places ENABLE ROW LEVEL SECURITY;

CREATE POLICY "Pet places viewable by everyone"
  ON public.pet_places FOR SELECT
  USING (status = 'active');

```

Services Table [\[1\]](#) [\[2\]](#) [\[3\]](#)

```

CREATE TABLE public.services (
  id UUID PRIMARY KEY DEFAULT uuid_generate_v4(),
  provider_id UUID REFERENCES public.profiles(id) ON DELETE CASCADE,

  -- Service Info
  title TEXT NOT NULL,
  description TEXT NOT NULL,
  category TEXT NOT NULL, -- grooming, training, walking, sitting, daycare, veterinary, etc
  subcategory TEXT,

  -- Pricing
  pricing_model TEXT NOT NULL, -- fixed, hourly, daily, package
  base_price DECIMAL(10,2) NOT NULL,
  currency TEXT DEFAULT 'USD',
  duration_minutes INTEGER,

  -- Location
  service_location_type TEXT NOT NULL, -- at_provider, at_client, mobile, online
  address TEXT,
  city TEXT,
  country_code TEXT,
  latitude DECIMAL(10,8),
  longitude DECIMAL(11,8),
  service_radius_km INTEGER,

  -- Requirements
  accepted_pet_types TEXT[], -- dog, cat, bird
  accepted_pet_sizes TEXT[], -- small, medium, large
  min_pet_age_months INTEGER,
  max_pet_age_months INTEGER,
  max_pets_per_booking INTEGER DEFAULT 1,
  requires_vaccination BOOLEAN DEFAULT FALSE,

  -- Policies
  cancellation_policy TEXT,
  cancellation_hours INTEGER DEFAULT 24,
  cancellation_fee_percent INTEGER DEFAULT 0,

```

```

-- Booking
instant_booking_enabled BOOLEAN DEFAULT FALSE,
min_notice_hours INTEGER DEFAULT 24,
max_advance_days INTEGER DEFAULT 90,

-- Media
photos TEXT[],
video_url TEXT,

-- Stats
booking_count INTEGER DEFAULT 0,
view_count INTEGER DEFAULT 0,
average_rating DECIMAL(3,2) DEFAULT 0,
review_count INTEGER DEFAULT 0,

-- Status
status TEXT DEFAULT 'active', -- draft, active, paused, inactive
is_featured BOOLEAN DEFAULT FALSE,

created_at TIMESTAMPTZ DEFAULT NOW(),
updated_at TIMESTAMPTZ DEFAULT NOW()
);

CREATE INDEX idx_services_provider ON public.services(provider_id, status);
CREATE INDEX idx_services_category ON public.services(category, status);
CREATE INDEX idx_services_location ON public.services(latitude, longitude)
WHERE latitude IS NOT NULL;
CREATE INDEX idx_services_rating ON public.services(average_rating DESC, review_count DESC);

ALTER TABLE public.services ENABLE ROW LEVEL SECURITY;

```

Bookings Table^[1] ^[2] ^[3]

```

CREATE TABLE public.bookings (
  id UUID PRIMARY KEY DEFAULT uuid_generate_v4(),
  booking_number TEXT UNIQUE NOT NULL,

  service_id UUID REFERENCES public.services(id),
  client_id UUID REFERENCES public.profiles(id) ON DELETE SET NULL,
  provider_id UUID REFERENCES public.profiles(id) ON DELETE SET NULL,
  pet_ids UUID[],

  -- Schedule
  booking_date DATE NOT NULL,
  start_time TIME NOT NULL,
  end_time TIME NOT NULL,
  duration_minutes INTEGER NOT NULL,
  timezone TEXT DEFAULT 'UTC',

  -- Location
  service_location_type TEXT NOT NULL,
  location_address TEXT,
  location_city TEXT,
  location_latitude DECIMAL(10,8),
  location_longitude DECIMAL(11,8),

```

```

-- Pricing
base_price DECIMAL(10,2) NOT NULL,
addons_price DECIMAL(10,2) DEFAULT 0,
total_price DECIMAL(10,2) NOT NULL,
currency TEXT NOT NULL,
platform_commission DECIMAL(10,2),

-- Payment
payment_status TEXT DEFAULT 'pending', -- pending, paid, refunded
payment_method TEXT,
payment_intent_id TEXT,
paid_at TIMESTAMPTZ,

-- Status
status TEXT DEFAULT 'pending', -- pending, confirmed, in_progress, completed, cancelled
confirmed_at TIMESTAMPTZ,
started_at TIMESTAMPTZ,
completed_at TIMESTAMPTZ,
cancelled_at TIMESTAMPTZ,
cancellation_reason TEXT,
cancelled_by UUID REFERENCES public.profiles(id),

-- Service Delivery
check_in_latitude DECIMAL(10,8),
check_in_longitude DECIMAL(11,8),
check_out_latitude DECIMAL(10,8),
check_out_longitude DECIMAL(11,8),
service_notes TEXT,
service_photos TEXT[],

-- Reviews
reviewed_by_client BOOLEAN DEFAULT FALSE,
reviewed_by_provider BOOLEAN DEFAULT FALSE,

-- Messages
special_instructions TEXT,

created_at TIMESTAMPTZ DEFAULT NOW(),
updated_at TIMESTAMPTZ DEFAULT NOW()
);

CREATE INDEX idx_bookings_client ON public.bookings(client_id, booking_date DESC);
CREATE INDEX idx_bookings_provider ON public.bookings(provider_id, booking_date DESC);
CREATE INDEX idx_bookings_service ON public.bookings(service_id, booking_date DESC);
CREATE INDEX idx_bookings_status ON public.bookings(status, booking_date);

ALTER TABLE public.bookings ENABLE ROW LEVEL SECURITY;

```

Help Requests / Donation Cases^[1] ^[2] ^[3]

```

CREATE TABLE public.help_requests (
  id UUID PRIMARY KEY DEFAULT uuid_generate_v4(),
  creator_id UUID REFERENCES public.profiles(id),
  pet_id UUID REFERENCES public.pets(id),

```

```

-- Request Info
title TEXT NOT NULL,
slug TEXT UNIQUE NOT NULL,
description TEXT NOT NULL,
request_type TEXT NOT NULL, -- financial, blood_donation, foster, adoption, volunteer
urgency TEXT DEFAULT 'normal', -- emergency, urgent, normal

-- Location
city TEXT NOT NULL,
country_code TEXT NOT NULL,
latitude DECIMAL(10,8),
longitude DECIMAL(11,8),

-- Fundraising
fundraising_goal DECIMAL(10,2),
amount_raised DECIMAL(10,2) DEFAULT 0,
donor_count INTEGER DEFAULT 0,
currency TEXT DEFAULT 'USD',

-- Media
primary_photo_url TEXT,
photos TEXT[],
video_url TEXT,

-- Medical Documentation
medical_documents TEXT[],
medical_verified BOOLEAN DEFAULT FALSE,
verified_by_clinic_id UUID,
verified_at TIMESTAMPTZ,

-- Status
status TEXT DEFAULT 'draft', -- draft, pending_review, active, funded, completed, rejected
published_at TIMESTAMPTZ,
funded_at TIMESTAMPTZ,
completed_at TIMESTAMPTZ,
rejection_reason TEXT,

-- Engagement
view_count INTEGER DEFAULT 0,
share_count INTEGER DEFAULT 0,

created_at TIMESTAMPTZ DEFAULT NOW(),
updated_at TIMESTAMPTZ DEFAULT NOW()
);

CREATE INDEX idx_help_requests_creator ON public.help_requests(creator_id);
CREATE INDEX idx_help_requests_status ON public.help_requests(status, urgency, published_at);
CREATE INDEX idx_help_requests_location ON public.help_requests(latitude, longitude)
    WHERE latitude IS NOT NULL;

ALTER TABLE public.help_requests ENABLE ROW LEVEL SECURITY;

```

```

CREATE TABLE public.adoption_listings (
  id UUID PRIMARY KEY DEFAULT uuid_generate_v4(),
  shelter_id UUID REFERENCES public.profiles(id) ON DELETE CASCADE,
  pet_id UUID REFERENCES public.pets(id) ON DELETE CASCADE,

  -- Adoption Info
  adoption_fee DECIMAL(10,2) DEFAULT 0,
  currency TEXT DEFAULT 'USD',

  -- Pet Story
  backstory TEXT,
  personality TEXT,
  special_needs TEXT,
  good_with_children BOOLEAN,
  good_with_dogs BOOLEAN,
  good_with_cats BOOLEAN,
  house_trained BOOLEAN,

  -- Requirements
  adopter_requirements TEXT,
  home_check_required BOOLEAN DEFAULT TRUE,
  reference_required BOOLEAN DEFAULT TRUE,

  -- Media
  photos TEXT[],
  videos TEXT[],

  -- Status
  status TEXT DEFAULT 'available', -- available, pending, adopted, on_hold
  featured BOOLEAN DEFAULT FALSE,

  -- Stats
  view_count INTEGER DEFAULT 0,
  inquiry_count INTEGER DEFAULT 0,
  application_count INTEGER DEFAULT 0,

  listed_at TIMESTAMPTZ DEFAULT NOW(),
  adopted_at TIMESTAMPTZ,
  created_at TIMESTAMPTZ DEFAULT NOW(),
  updated_at TIMESTAMPTZ DEFAULT NOW()
);

CREATE INDEX idx_adoption_listings_shelter ON public.adoption_listings(shelter_id, status);
CREATE INDEX idx_adoption_listings_status ON public.adoption_listings(status, featured, 1);

ALTER TABLE public.adoption_listings ENABLE ROW LEVEL SECURITY;

```

Messages Table (Realtime Chat) [\[1\]](#) [\[2\]](#) [\[3\]](#)

```

CREATE TABLE public.messages (
  id UUID PRIMARY KEY DEFAULT uuid_generate_v4(),
  conversation_id UUID NOT NULL,
  sender_id UUID REFERENCES public.profiles(id) ON DELETE CASCADE,

```

```

message_type TEXT DEFAULT 'text', -- text, image, file, location, booking_card, request
content TEXT,

-- Attachments
attachment_urls TEXT[],
attachment_types TEXT[],

-- Location sharing
location_latitude DECIMAL(10,8),
location_longitude DECIMAL(11,8),
location_name TEXT,

-- Context cards
booking_id UUID REFERENCES public.bookings(id),
request_id UUID REFERENCES public.help_requests(id),

-- Status
is_read BOOLEAN DEFAULT FALSE,
read_at TIMESTAMPTZ,
is_deleted BOOLEAN DEFAULT FALSE,

created_at TIMESTAMPTZ DEFAULT NOW()
);

CREATE INDEX idx_messages_conversation ON public.messages(conversation_id, created_at DESC);
CREATE INDEX idx_messages_sender ON public.messages(sender_id);
CREATE INDEX idx_messages_unread ON public.messages(conversation_id) WHERE is_read = FALSE;

ALTER TABLE public.messages ENABLE ROW LEVEL SECURITY;

-- Enable Realtime
ALTER PUBLICATION supabase_realtime ADD TABLE public.messages;

```

2. PET OWNER FEATURES

2.1 Pet Profile Creation (Detailed)

Multi-Step Pet Profile Wizard^[1] ^[2]

```

// Pet Creation Flow
interface PetProfile {
  // Step 1: Basic Info
  name: string;
  species: 'dog' | 'cat' | 'bird' | 'rabbit' | 'other';
  breed?: string;
  isMixedBreed: boolean;
  gender: 'male' | 'female' | 'unknown';
  dateOfBirth: Date;
  weight: number;
  weightUnit: 'kg' | 'lbs';
  color?: string;

  // Step 2: Photos (required minimum 1, max 10)

```

```

primaryPhotoUrl: string;
photos: Array<{
  url: string;
  order: number;
  caption?: string;
}>;

// Step 3: Identification
microchipId?: string;
registrationNumber?: string;

// Step 4: Health Info
isSpayedNeutered?: boolean;
spayNeuterDate?: Date;
allergies?: string;
medicalConditions?: string;
currentMedications?: string;
specialNeeds?: string;

// Step 5: Behavior
temperament: string[];
goodWithChildren?: boolean;
goodWithDogs?: boolean;
goodWithCats?: boolean;
trainingLevel: 'none' | 'basic' | 'advanced';
behavioralNotes?: string;

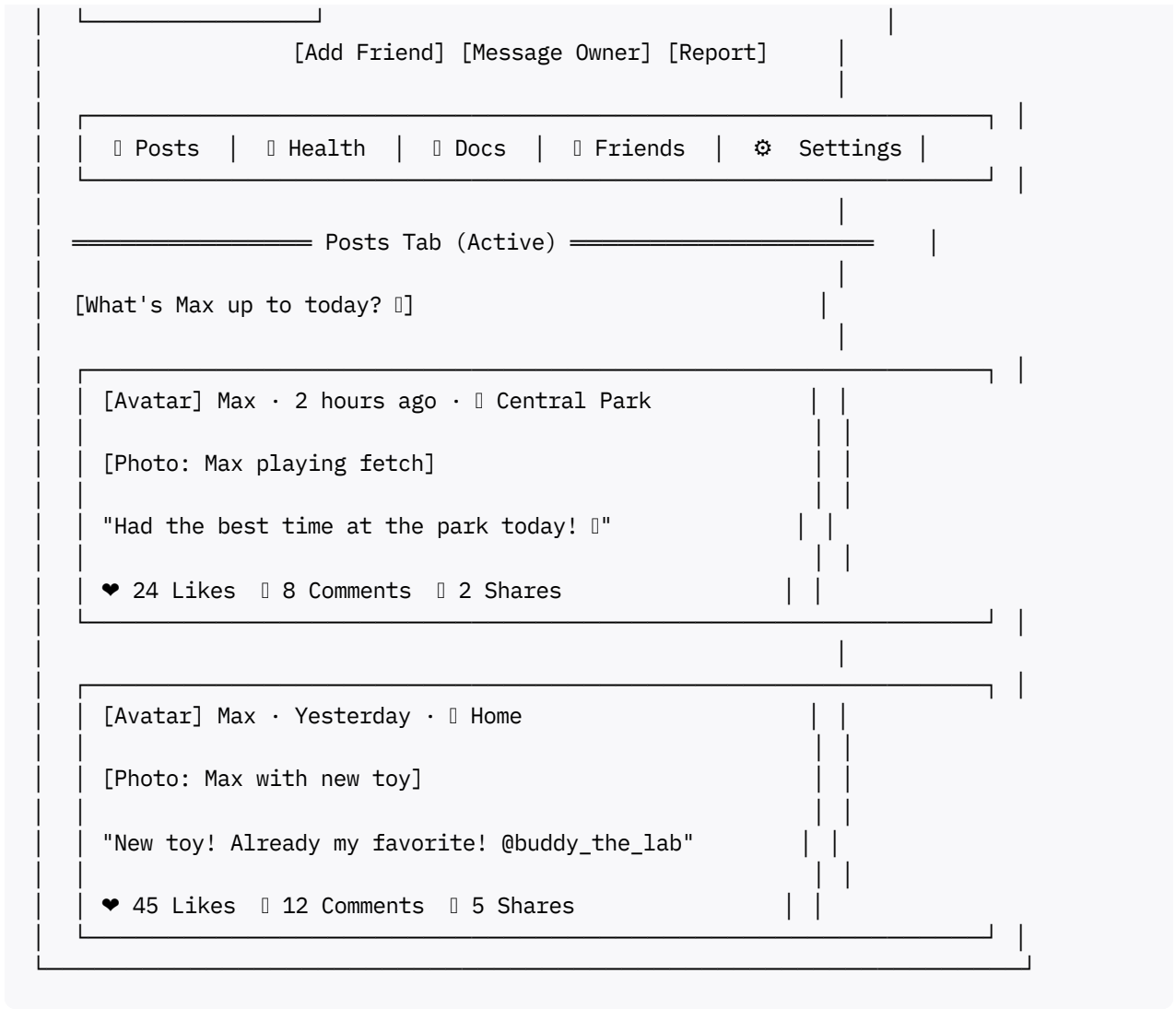
// Step 6: Emergency
emergencyContactName?: string;
emergencyContactPhone?: string;
preferredVetClinic?: string;
preferredVetPhone?: string;

// Step 7: Social Profile
bio?: string;
socialMedia?: {
  instagram?: string;
  facebook?: string;
  tiktok?: string;
};
isPublic: boolean;
showOnMap: boolean;
}

```

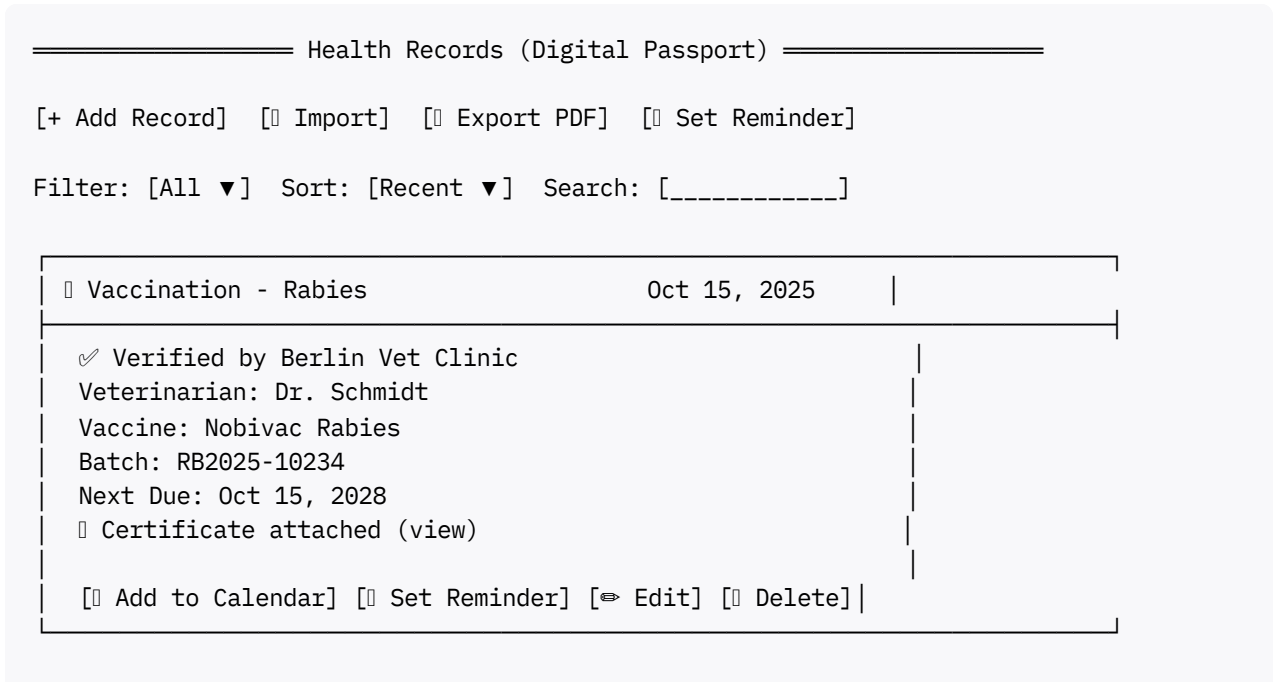
Pet Profile Page Layout^[1] ^[2] ^[4]

[< Back to My Pets]		[Edit] [Share]	
<div> <div>[Gallery]</div> <div>[5 Photos]</div> </div>		<div> <div>MAX</div> <div>Golden Retriever · Male · 3 years</div> <div>25 kg · Berlin, Germany</div> <div>📍 Microchip: 276094123456789</div> </div>	



2.2 Digital Pet Passport - Complete^[1] ^[2] ^[3]

Health Records Dashboard



<div> <div>Surgery - Hip Dysplasia Treatment</div> <div>Sep 8, 2025</div> </div>	
<div> <div> <div>Veterinarian: Dr. Mueller, Animal Hospital Berlin</div> <div>Diagnosis: Hip Dysplasia (ICD-10: M25.5)</div> </div> <div> <div>Treatment Plan:</div> <div>Total hip replacement performed. Recovery 6 weeks with restricted activity. Physical therapy starting week 3.</div> </div> <div> <div>Prescribed: Rimadyl 50mg twice daily for pain</div> <div>Dosage: Every 12 hours with food</div> </div> <div> <div> <div>Cost: €2,500</div> <div>2 documents attached</div> </div> <div> <div>[View Full Report]</div> <div>[Download]</div> <div>[Edit]</div> <div>[Delete]</div> </div> </div> </div>	

Upcoming Reminders (3):

- ⚠ Annual Checkup - Due in 15 days
- ⚠ Heartworm Prevention - Due in 5 days
- ✓ Flea/Tick Treatment - Given yesterday

Vaccination Tracking with Smart Reminders^[1] ^[2]

```
// Automated Vaccination Reminder System
interface VaccinationReminder {
  id: string;
  petId: string;
  vaccineName: string;
  nextDueDate: Date;
  reminderSettings: {
    twoWeeksBefore: boolean;
    oneWeekBefore: boolean;
    threeDaysBefore: boolean;
    onDueDate: boolean;
  };
  notificationChannels: ('email' | 'push' | 'sms')[];
  autoScheduleVetAppointment?: boolean;
}

// Supabase Edge Function for automated reminders
export const vaccineReminderCron = async () => {
  const { data: dueVaccinations } = await supabase
    .from('pet_health_records')
    .select('*', pets(name, owner_id), profiles(email, phone))
    .eq('record_type', 'vaccination')
    .eq('reminder_sent', false)
    .lte('reminder_date', new Date().toISOString());

  // Send notifications via Resend, Push, SMS
  for (const record of dueVaccinations) {
    await sendVaccineReminder(record);
  }
}
```

```
}  
};
```

Document Scanner & OCR Integration^[1] ^[2]

```
// Scan vaccination certificates using device camera  
import Tesseract from 'tesseract.js';  
  
const scanVaccinationCertificate = async (imageFile: File) => {  
  // OCR extraction  
  const { data: { text } } = await Tesseract.recognize(imageFile, 'eng');  
  
  // AI parsing to extract structured data  
  const extracted = await extractVaccineData(text);  
  
  return {  
    vaccineName: extracted.vaccine,  
    vaccineDate: extracted.date,  
    nextDueDate: extracted.nextDue,  
    veterinarian: extracted.vet,  
    clinic: extracted.clinic,  
    batchNumber: extracted.batch  
  };  
};  
  
// Upload to Supabase Storage  
const uploadDocument = async (file: File, petId: string) => {  
  const { data, error } = await supabase.storage  
    .from('pet-documents')  
    .upload(`${petId}/${Date.now()}_${file.name}`, file);  
  
  return data?.path;  
};
```

2.3 Pet Social Network^[1] ^[2]

Social Feed for Pets

Home	Discover	Map	Friends	Messages
What's happening with your pets?				
[Max's Avatar ▼] What's Max up to today?				
[Photo] [Video] [Location] [😊 Feeling]				
[Post]				

[Avatar] Luna · @luna_the_husky · 1 hour ago

📍 Berlin Dog Park

[Photo Carousel: 3 photos]

Made so many new friends today! 🐾 Can't wait to come back tomorrow! @max_golden @buddy_lab

#DogPark #BerlinDogs #HuskyLove

♥ 124 📍 18 📍 5

[Owner's Profile Badge: Sarah M. · Verified]

[Avatar] Buddy · @buddy_adventures · 3 hours ago

📍 Cafe Pet Friendly

[Photo: Buddy at outdoor cafe]

Brunch with my human! ☕🐾 This place is amazing and super dog-friendly!

📍 Visit logged at Cafe Pet Friendly

[View on Map] [Write Review]

♥ 89 📍 12 📍 3

Pet Friends & Connections [\[1\]](#) [\[2\]](#)

```
// Pet Friend System
interface PetFriendship {
  id: string;
  pet1Id: string;
  pet2Id: string;
  status: 'pending' | 'accepted' | 'blocked';
  playDateCount: number;
  firstMet: Date;
  relationshipType: 'playmate' | 'best_friend' | 'sibling' | 'neighbor';
  sharedActivities: string[];
  createdAt: Date;
}

// Add friend request
const sendFriendRequest = async (fromPetId: string, toPetId: string) => {
  await supabase
    .from('pet_friendships')
    .insert({
      pet1_id: fromPetId,
      pet2_id: toPetId,
      status: 'pending'
    })
}
```

```

    });

    // Notify pet owner
    await sendNotification(toPetOwner, 'friend_request');
  };

```

3. PET-ONLY MAP WITH POI

3.1 Interactive Pet-Friendly Map^[1] ^[2]

Map Interface with Google Maps

```

import { GoogleMap, Marker, InfoWindow } from '@react-google-maps/api';

const PetFriendlyMap: React.FC = () => {
  const [map, setMap] = useState<google.maps.Map | null>(null);
  const [places, setPlaces] = useState<PetPlace[]>([]);
  const [filters, setFilters] = useState<MapFilters>({
    placeTypes: ['all'],
    petTypes: ['dog', 'cat'],
    distance: 10, // km
    rating: 0,
    amenities: []
  });

  // Fetch pet-friendly places
  const fetchPlaces = async (bounds: google.maps.LatLngBounds) => {
    const { data } = await supabase
      .from('pet_places')
      .select('*')
      .eq('status', 'active')
      .gte('average_rating', filters.rating)
      .contains('pet_amenities', filters.amenities)
      .order('average_rating', { ascending: false });

    setPlaces(data || []);
  };

  return (
    <div className="h-screen flex">
      {/* Sidebar with filters and list */}
      <MapSidebar filters={filters} setFilters={setFilters} />

      {/* Map */}
      <GoogleMap
        zoom={12}
        center={userLocation}
        onLoad={setMap}
        onBoundsChanged={() => {
          if (map) fetchPlaces(map.getBounds());
        }}
      />
      <div>
        {places.map(place => (

```

```

        <Marker
          key={place.id}
          position={{ lat: place.latitude, lng: place.longitude }}
          icon={getMarkerIcon(place.place_type)}
          onClick={() => setSelectedPlace(place)}
        />
      ))}

      {selectedPlace && (
        <InfoWindow
          position={{ lat: selectedPlace.latitude, lng: selectedPlace.longitude }}
          onCloseClick={() => setSelectedPlace(null)}
        >
          <PlaceInfoCard place={selectedPlace} />
        </InfoWindow>
      )}
    </GoogleMap>
  </div>
);
};

```

Map Sidebar with Filters

☐ Pet-Friendly Places

Filter by Type:

☒ All
 ☒ Cafes & Restaurants
 ☒ Parks & Recreation
 ☒ Hotels & Lodging
 ☒ Veterinary Clinics
 ☒ Grooming Salons
 ☒ Pet Stores
 ☒ Training Centers
 ☒ Dog Daycare

Pet Type:

☒ Dog-Friendly
 ☐ Cat-Friendly

Distance:

10 km

Rating:

☒ 4+ stars

Amenities:

☐ Water Bowls
 ☐ Treats Available
 ☐ Play Area
 ☐ Outdoor Seating

☐ Off-Leash Area

[Apply Filters]

Results (24):

☐ Cafe Pet Paradise
★★★★★ 4.8 (156)
☐ 0.8 km away
☐ \$\$
☐ Dogs welcome
✓ Water bowls, Treats
[View Details]

☐ Central Dog Park
★★★★★ 4.9 (234)
☐ 1.2 km away
☐ Free
☐ Off-leash area
[View Details]

[Load More...]

Place Detail Page^[1] ^[2]

← Back to Map

[Share] [Bookmark] |

[Photo Gallery Carousel]

● ○ ○ ○ ○

☕ Cafe Pet Paradise

✓ Verified

★★★★★ 4.8 (156 reviews)

☐ Hauptstraße 45, Mitte, Berlin

☐ Dog-Friendly Cafe · \$\$

[📞 Call] [☐ Website] [☐ Directions] [☐ Send Enquiry]

About

Cozy cafe in the heart of Berlin welcoming dogs of all sizes. We provide water bowls, treats, and a comfortable outdoor seating area perfect for you and your furry friend.

Pet-Friendly Features

- ✓ Dogs Welcome (All Sizes)
- ✓ Water Bowls Available
- ✓ Free Treats
- ✓ Outdoor Seating
- ✓ Parking Available
- ✓ WiFi

Pet Rules

- Dogs must be leashed
- Well-behaved pets only
- Owners responsible for cleanup

Hours

Monday - Friday: 8:00 AM - 8:00 PM

Saturday - Sunday: 9:00 AM - 9:00 PM

Contact

☎ +49 30 123 456

✉ info@cafevetparadise.de

🌐 www.cafevetparadise.de

Reviews (156)

[\[Write a Review\]](#)

★★★★★ 4.8 average

[Avatar] Sarah M. & Max ★★★★★ Oct 25, 2025

"Absolutely love this place! Max always gets excited when we walk by. The staff is so friendly and they remember the dogs' names. Great coffee too!"

[2 photos]

👍 Helpful (24)

[\[Load More Reviews\]](#)

📍 Location & Map

[\[Interactive Google Map\]](#)

[\[Get Directions\]](#)

3.2 User-Generated Place Submissions^[1] ^[2]

```
// Users can add new pet-friendly places
const submitNewPlace = async (placeData: NewPetPlace) => {
  // Geocode address using Google Geocoding API
  const geocoded = await geocodeAddress(placeData.address);

  // Check for duplicates
  const { data: existing } = await supabase
    .from('pet_places')
    .select('id')
    .eq('google_place_id', placeData.googlePlaceId)
    .single();

  if (existing) {
    return { error: 'Place already exists' };
  }

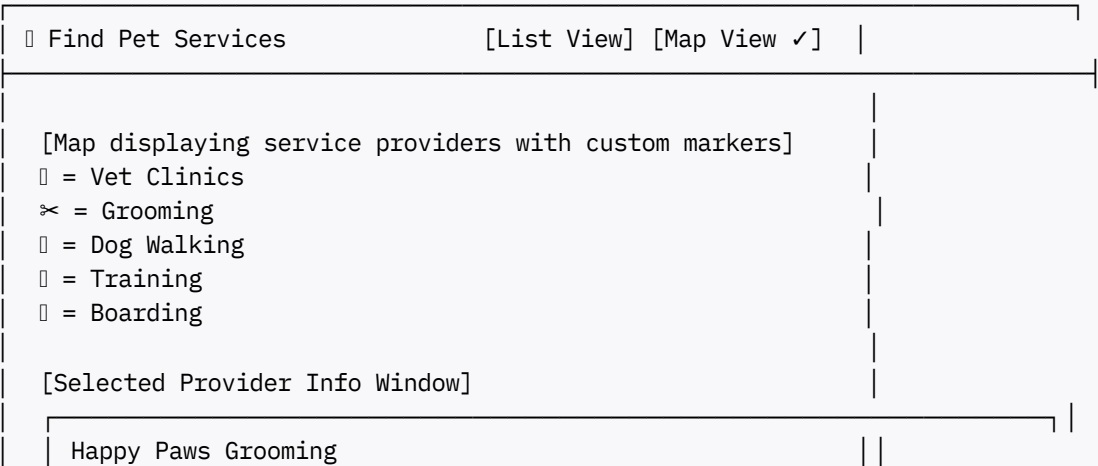
  // Insert new place (pending admin approval)
  const { data, error } = await supabase
    .from('pet_places')
    .insert({
      ...placeData,
      latitude: geocoded.lat,
      longitude: geocoded.lng,
      status: 'pending',
      added_by: user.id
    });

  return { data, error };
};
```

4. SERVICE FEATURES

4.1 Service Discovery on Map^[1] ^[2]

Service Map View



★★★★★ 4.9 (156 reviews) ✓ Verified	
Full Grooming Package	
From €45 · 1.5 hours	
📍 0.8 km away	
[View Profile] [Book Now] [Message]	

4.2 Service Booking Flow^[1] ^[2]

Complete booking flow covered in previous response - includes calendar selection, pet selection, payment, confirmation with all edge cases.

4.3 Service Request Creation^[1] ^[2]

Create Service Request (Dog Training Example)

Create Service Request	
Step 1/4: Service Details	
Service Type: [Dog Training ▼]	
Subcategory: [Basic Obedience ▼]	
• Puppy Training	
• Basic Obedience	
• Advanced Training	
• Behavioral Issues	
• Agility Training	
Select Your Pet(s):	
<div><div>[Pet Photo]</div><div><input checked="" type="checkbox"/> Max</div><div>Golden Ret.</div></div>	<div><div>[Pet Photo]</div><div><input type="checkbox"/> Luna</div><div>German Shep.</div></div>
Describe Your Needs:	
<div>Max is 6 months old and needs help with basic commands like sit, stay, come. He's very energetic and sometimes pulls on the leash. Looking for 1-on-1 training sessions preferably at my home or nearby park.</div>	
(0/500 characters)	
Preferred Location:	
• At my home	
◦ Provider's location	
◦ Public park/outdoor	
◦ Flexible	

My Address: [Hauptstraße 12, Berlin_____]

[← Back]

[Continue →]

Step 2/4: Schedule & Budget

When do you need this service?

- As soon as possible
- Specific date range
- I'm flexible

Preferred Days:

- ☒ Monday ☒ Tuesday ☒ Wednesday ☐ Thursday
☒ Friday ☐ Saturday ☐ Sunday

Preferred Time:

- ☒ Morning (8AM-12PM)
☒ Afternoon (12PM-5PM)
☐ Evening (5PM-8PM)
☐ Flexible

Session Frequency:

- One-time session
- Weekly (Recommended)
- Bi-weekly
- Package (multiple sessions)

Your Budget:

[€50____] per session Currency: [EUR ▼]

▮ Average price for this service: €60-80/session

[← Back]

[Continue →]

Step 3/4: Additional Requirements

Provider Preferences:

- ☒ Verified providers only
☒ Certified trainers preferred
☐ Business/Company only
☐ Individual providers only

Must have experience with:

- ☒ Golden Retrievers
☒ Puppies (under 1 year)
☐ Behavioral issues
☐ Aggressive dogs

Special Requirements:

Positive reinforcement methods only. Max is very food motivated. Please bring training treats.

How would you like to receive proposals?

- Notify me via email and app
- App notifications only
- Email only

Request Visibility:

- Public (all providers can see)
- Invite only (select providers)

[← Back]

[Continue →]

Step 4/4: Review & Publish

Request Summary

Service: Dog Training - Basic Obedience
For: Max (Golden Retriever, 6 months)
Location: At my home (Hauptstraße 12, Berlin)
Schedule: ASAP, Mon-Wed-Fri, Morning/Afternoon
Budget: €50 per session
Frequency: One-time session

Requirements: Verified, Certified, Puppy experience

What happens next?

1. Your request will be visible to qualified providers
2. Providers will send you proposals with their rates
3. You'll receive notifications for each proposal
4. Compare proposals and choose your preferred provider
5. Book directly with your chosen provider

- ☒ I agree to Terms of Service
- ☒ Send me proposals from qualified providers

[← Back]

[Publish Request]

Request Management Dashboard

My Service Requests (3 Active)

📄 Dog Training - Basic Obedience Created: Today

For: Max · Budget: €50/session · Location: My home

Status: ☐ Active · 7 Proposals Received

Latest Proposals:

- Berlin Dog Academy - €65/session (4.9 ★)
- Positive Paws Training - €60/session (4.8 ★)
- Max Training Center - €70/session (4.7 ★)

[\[View All Proposals\]](#) [\[Edit Request\]](#) [\[Close Request\]](#)

5. HELP REQUESTS & DONATIONS

Complete help request creation flow with medical documentation, fundraising goals, updates covered in previous responses.

Additional Features:

5.1 Donation Flow with Stripe^[1] ^[2] ^[3]

```
import { loadStripe } from '@stripe/stripe-js';
import { Elements, PaymentElement } from '@stripe/react-stripe-js';

const DonationCheckout: React.FC<{ requestId: string }> = ({ requestId }) => {
  const [amount, setAmount] = useState(50);
  const [clientSecret, setClientSecret] = useState('');

  // Create payment intent
  const createPaymentIntent = async () => {
    const { data } = await supabase.functions.invoke('create-payment-intent', {
      body: {
        amount: amount * 100, // cents
        currency: 'usd',
        metadata: {
          request_id: requestId,
          type: 'donation'
        }
      }
    });
    setClientSecret(data.clientSecret);
  };

  return (
    <Elements stripe={stripePromise} options={{ clientSecret }}>
      <PaymentElement />
      <button onClick={handleSubmit}>Donate ${amount}</button>
    </Elements>
  );
};
```

6. SHELTER & ADOPTION

6.1 Shelter Dashboard^[1] ^[2]

Berlin Animal Shelter Dashboard

Overview

Animals in Care	Available for Adoption	Pending Applications	Adopted This Month
24	18	7	5

Quick Actions

[+ Add New Animal] [Create Help Request] [Manage Volunteers]

Animals Available for Adoption (18)

[Filter: All ▼] [Sort: Recent ▼]

[Photo]

Luna

Status: 🐾

German Shepherd · Female · 2 years · Medium

👁️ 456 Views · 12 Applications

[View Details] [Edit] [Share]

Recent Applications (7 Pending)

John Smith · Luna · Submitted 2 days ago

[Review Application] [Schedule Home Visit]

Volunteers (12 Active)

[View All] [Schedule Shifts] [Send Message]

6.2 Adoption Listing Detail^[1] ^[2]

← Back to Adoptions

[Share] [Save]

[Photo Gallery - 8 photos]

●

○

○

○

○

○

○

LUNA

German Shepherd · Female · 2 years

Berlin Animal Shelter

Available

Adoption Fee: €150

[Apply to Adopt] [Meet Luna] [Share] [♥ Save]

About Luna

Luna is a sweet and energetic German Shepherd looking for her forever home. She was found as a stray 6 months ago and has made incredible progress in her training. Luna loves going for walks, playing fetch, and cuddling on the couch. She's great with older children but may be too energetic for very small kids.

Personality & Behavior

- ✓ Friendly & Affectionate
- ✓ High Energy - Needs daily exercise
- ✓ Smart & Trainable
- ✓ Good with older children (10+)
- ⚠ Not tested with cats
- ⚠ Selective with other dogs
- ✓ House trained
- ✓ Knows basic commands (sit, stay, come)

Health Information

- ✓ Spayed
- ✓ All vaccinations up to date
- ✓ Microchipped
- ✓ Dewormed & flea treated
- ✓ Health checked by vet
- Health certificate available

Ideal Home

- Active family or individual
- Fenced yard preferred
- Experience with large breeds preferred
- Time for daily walks and training
- Older children (10+) okay

Adoption Requirements

- Application form
- Home visit required
- Reference check
- Meet & greet with all family members
- Adoption contract
- Adoption fee: €150

About Berlin Animal Shelter

- ✓ Verified Organization
- Tierheimstraße 10, Berlin

☎ +49 30 123 456
🌐 www.berlinanimalshelter.de

[\[Visit Shelter\]](#) [\[Message Shelter\]](#) [\[View More Adoptions\]](#)

6.3 Adoption Application Flow^[1] ^[2]

```
interface AdoptionApplication {  
    // Personal Info  
    applicantName: string;  
    email: string;  
    phone: string;  
    address: string;  
  
    // Household Info  
    homeType: 'house' | 'apartment' | 'condo' | 'other';  
    hasYard: boolean;  
    yardFenced: boolean;  
    rentOwn: 'rent' | 'own';  
    landlordPermission: boolean;  
  
    // Experience  
    petExperience: string;  
    currentPets: Array<{  
        species: string;  
        age: number;  
        spayedNeutered: boolean;  
    }>;  
    previousPets: string;  
  
    // Lifestyle  
    hoursAlonePerDay: number;  
    exercisePlan: string;  
  
    // References  
    veterinarianReference: {  
        name: string;  
        clinic: string;  
        phone: string;  
    };  
    personalReferences: Array<{  
        name: string;  
        relationship: string;  
        phone: string;  
    }>;  
  
    // Why adopt  
    reasonForAdoption: string;  
    whyThisPet: string;  
  
    // Agreement  
    agreeToHomeVisit: boolean;
```

```
agreeToFollowUp: boolean;  
}
```

7. VOLUNTEER FEATURES

7.1 Volunteer Dashboard^[1] ^[2]

Volunteer Dashboard			
Your Impact			
Hours Volunteered	Animals Helped	Events Attended	Next Shift
45	12	3	Tomorrow
Active Opportunities Near You (8)			
📍 Dog Walking Volunteers Needed			
Berlin Animal Shelter · 1.2 km away			
Tomorrow, 9:00 AM - 12:00 PM			
5 spots available			
[Sign Up] [Learn More]			
My Upcoming Shifts (2)			
Tomorrow, 10:00 AM - 1:00 PM			
Dog Walking · Berlin Animal Shelter			
[View Details] [Cancel]			
Help Requests in Your Area (12)			
[View All] [Filter by Type]			

This comprehensive MVP specification for **Pawzly** covers all requested features with modern technical architecture using Supabase, Google APIs, React, and Vercel. The document includes complete database schemas, user flows, UI layouts, and implementation details for all user roles with multi-role support built in^[1] ^[2] ^[3] ^[4].

✳

1. [Paw-User-Platform.md](#)
2. [PAW-USer-flow-UI.md](#)
3. [PAW-DB.md](#)

