# 🐾 Dog Mating Web Application – Full Implementation Document

## 📌 Purpose

To build a **dedicated platform for dog owners** to find suitable partners for their pets, ensure healthy and successful breeding, communicate with vets, and manage all related needs — from care to clinic discovery — in a single intelligent, community-driven app.

## ❗ Problem Statement

- No trusted online platform to find safe and healthy dog mating partners.

- Pet owners lack AI-based guidance on health, care, or matching.

- Offline/word-of-mouth methods are unreliable.

- No integrated booking, vet support, or review system post-mating.

- Clinics and doctors are hard to search and compare quickly.

📌 **Why Build This Web App?**

The purpose of this web application is to provide a safe, intelligent, and community-driven platform for dog owners who are looking to mate their pets responsibly. Traditional channels like classified ads or social media lack credibility, structure, and medical assurance — creating risks for both dogs and owners.

This web app ensures:

- Verified matches based on breed, health, and behavior traits

- Health checks and vet-verification of dogs before mating

- Secure, private communication between dog owners

- Support for logistics like location and clinic options

- Past mating experience tracking and owner reviews

This platform promotes ethical breeding practices, transparency, and improves pet welfare overall.

## 👥 Stakeholders and Users

**1. Dog Owners (Primary Users)**

Dog owners are the core users of the platform. Their needs include:

- Creating profiles for their dogs

- Finding compatible mates based on breed, traits, age, and health

- Getting matched via filters like size, vaccination, experience, etc.

- Vetting potential partners through profiles and vet info

- Communicating securely with other dog owners

- Scheduling mating meetups at agreed locations

### 2. Veterinarians / Pet Clinics

● They play a critical role in validating dog health and readiness for mating

● Their details (clinic, license, specialization) can be listed

● They help ensure ethical and safe breeding practices

● Optional: Offer in-clinic mating space

● Can issue "Mating Fitness Certificates" viewable on profiles

### 3. Platform Admin / Product Owner

● Manages backend system and moderation

● Approves veterinarian registrations

● Handles abuse reports, fake profiles, and security

● Manages analytics, growth strategies, and user engagement

● Can monetize the platform through subscriptions or ads

### 4. Pet Service Providers (Optional Extension)

● Trainers, breeders, groomers, pet photographers, etc.

● Can list services or advertise within the app

● Adds more value to the dog-owner community

## 📋 Major Requirements That Justify Building This Product

### 1. Need for Trustworthy Mating Platform

- Current market lacks verified, safe, and organized mating options

- Owners often rely on word of mouth, Facebook groups, or risky ads

### 2. Demand for Responsible Breeding

- Pet owners are increasingly aware of ethical breeding

- They want partners with vaccination, good temperament, and history

### 3. Health & Medical Assurance

- This platform integrates veterinary validation to ensure:

  - Dogs are healthy and of mating age

  - Not genetically at risk

  - Vaccinated and non-contagious

### 4. Communication & Experience

- Users want a transparent, secure chat system

- Chat logs, prior ratings, and past experiences add trust

- Profile-based matchmaking is more intelligent than random browsing

### 5. Logistics and Mating Preferences

- Location preferences (home, clinic, etc.)

- Matching based on setup compatibility

- Filters like breed, color, size, behavior, etc.

# 🎯 Features

## 👥 User Features

1. **User Registration/Login** with OTP

2. **Dog Profile Management**

3. **Chat System** between dog owners

4. **AI Chatbot** for dog care tips (diet, grooming, etc.)

5. **Mating Partner Matchmaking Engine** (breed, location, etc.)

6. **Booking System** for mating meet-ups (can also add pet hospital)

7. **Review System** after mating session

8. **Pet Clinic Suggestions** based on user's location

9. **Chat with Pet Doctors / Vets**

10. **Add Testimonials** about successful match or experience

## 🛠️ Admin Features

1. **Doctor Management**

   ○ Add/Edit/Delete pet doctors

   ○ Link doctors with pet clinics

2. **Export System**

   ○ Export dog details in PDF, Excel

3. **AI Document Analysis**

   ○ Upload PDF and extract dog-related issues (using LLM)

4. **Testimonials Moderation & Management**

5. **Clinic Management**

   ○ Add/Edit clinics & associate with doctors

## 🧠 AI-Powered Features

| Feature | Description |
|---|---|
| **Dog Care AI Bot** | Suggest grooming, diet, training based on age, breed, etc. |
| **Matchmaking Engine** | Uses breed, location, preferences to suggest matches |
| **LLM PDF Extractor** (Admin) | Upload medical history, get summarized dog issue reports |
| **Smart Clinic Recommender** | Suggest nearby clinics using geo-coordinates |

# 🤖 What are These?

### 🧠 Agentic AI

AI agents that **act autonomously** toward goals (e.g., booking, recommending, chatting with purpose, etc.)

### 🧠 RAG (Retrieval-Augmented Generation)

Uses **external knowledge** (e.g., FAQs, vet manuals, dog care PDFs) to enhance LLM responses with **accurate context**.

### 🧠 MCP Server (Multi-Component Pipeline Server)

Orchestrates multiple AI pipelines—ideal for **data processing**, **file analysis**, **smart decisions**, etc.

# 🔍 Mapping to Your Features

## 🎯 User Features

| Feature | Agentic AI | RAG Model | MCP Server | How? |
|---|---|---|---|---|
| 🧠 Dog Care AI Bot | ✅ Yes | ✅ Yes | ✅ Yes | RAG for knowledge-based Q&A; Agent to drive proactive suggestions (e.g., "Time to deworm!") |
| 🔍 Matchmaking Engine | ✅ Yes | ❌ No | ✅ Yes | Agent evaluates breed, location, preferences → finds best match (like a dating assistant) |
| 📅 Booking System | ✅ Yes | ❌ No | ✅ Yes | Agent can auto-suggest times, confirm bookings, and handle rescheduling |
| 🗣️ Chat with Vets | ✅ Yes | ✅ Yes | ❌ Optional | RAG for answering health queries (vaccination, diet, etc.); Agent can escalate to real vet |
| 📍 Smart Clinic Recommender | ✅ Yes | ❌ No | ✅ Yes | Agent fetches location + filters open clinics with services → recommends smartly |
| 📝 Testimonials Input | ❌ No | ❌ No | ✅ Yes | MCP can analyze/testimonials sentiment or filter toxic input |

## 🛠️ Admin Features

| Feature | Agentic AI | RAG Model | MCP Server | How? |
|---|---|---|---|---|
| 📄 LLM PDF Extractor | ❌ No | ✅ Yes | ✅ Yes | MCP manages doc input; RAG uses vet docs to summarize/extract illness/treatment |
| 🏥 Clinic Management | ✅ Yes | ❌ No | ✅ Yes | Agent auto-links new doctors to matching clinics |
| 👨‍⚕️ Doctor Management | ✅ Yes | ❌ No | ✅ Yes | Agent can suggest specialization mapping |
| 📤 Export System | ❌ No | ❌ No | ✅ Yes | MCP handles file formatting (PDF/Excel conversion), smart grouping |
| 🔍 Testimonial Moderation | ✅ Yes | ✅ Yes | ✅ Yes | RAG finds unsafe/toxic content; Agent flags or auto-approves positive ones |

## ✅ Example Agentic Workflows

- **"Mating Partner Finder Agent"**: Understands user's preferences → filters dogs → messages matching owners.

- **"CareBot Agent"**: Proactively checks if user updated health info → gives food/vaccine/grooming advice.

- **"Clinic Match Agent"**: Checks location, doctor availability, and services → suggests the best nearby vet.

## 🚀 Architecture Suggestion

- Use LangChain / LlamaIndex for RAG + agents.

- Use FastAPI or Express with Node.js for MCP Server.

- Store external knowledge in Pinecone / Weaviate / ChromaDB for vector search.

- Use OpenAI, Claude, or open LLMs for actual reasoning.

# 🐶 Dog Model

Core model for listing dogs available for mating.

| Field | Type | Description |
|---|---|---|
| dogId | UUID | Unique identifier for each dog. |
| name | String | Given name of the dog. |
| gender | Enum(Male, Female) | Dog's gender. |
| breed | String | Breed of the dog (e.g., Labrador, German Shepherd). |
| type | String (Optional) | Breed type/classification (e.g., Sporting, Herding). |
| age | Number | Dog's age in months/years. |
| dateOfBirth | Date (Optional) | Actual birthdate for precise age calculation. |
| color | String (Optional) | Coat color/markings. |
| weight | Number (Optional) | Weight of the dog (kg or lbs). |
| height | Number (Optional) | Height of the dog (cm or inches). |
| description | String | A brief bio or personality traits. |
| images | Array[String] | URLs of multiple dog photos. |
| video | String (Optional) | Video URL of the dog (walk, play, etc.). |
| isVaccinated | Boolean | Indicates if the dog is fully vaccinated. |
| isNeuteredOrSpayed | Boolean | Shows whether the dog is neutered/spayed. |
| healthNotes | String (Optional) | Details on any medical conditions or treatments. |
| dogAdoptionDate | Date | Date the current owner adopted the dog. |

| | | |
|---|---|---|
| dogAdoptionPlaceName | String | Place where the dog was adopted. |
| dogOwnerName | String | Name of the current owner. |
| dogPreviousOwnerName | String (Optional) | Name of the previous owner, if applicable. |
| isDogBorrowed | Boolean | Indicates if the dog is borrowed for mating. |
| lastMatingDate | Date (Optional) | Last known date the dog was mated. |
| numberOfPastMatings | Number | Count of previous mating events. |
| matingSuccessRate | Number (Optional) | Optional success rate of matings. |
| isAvailableForMating | Boolean | Availability flag for mating. |
| availableFrom | Date | When the dog becomes available. |
| availableTo | Date (Optional) | Until when the dog is available. |
| matingPreferences | Object | Owner-defined mating preferences (see below). |
| studFee | Number (Optional) | Fee for stud/mating services. |
| ownerTerms | String (Optional) | Notes from the owner regarding mating terms. |

# 🧩 Enhanced User Model

| Field | Type | Description |
|---|---|---|
| userId | UUID | Unique identifier for the user |
| name | String | Full name of the user |
| email | String | Email address (used for login and contact) |
| phone | String | Mobile number (can be used for OTP or contact) |
| otp | String | One-time password for verification |
| otpVerifiedAt | DateTime | When OTP was verified |
| otpExpiresAt | DateTime | OTP expiry timestamp |
| isLoggedInFirstTime | Boolean | Flag for onboarding-related features |
| userRole | Enum | User role: 'Admin', 'User' |
| profileImage | String (URL) | Optional profile picture |
| dateOfBirth | Date | Optional for age-based suggestions (AI care) |
| addressLine1 | String | Main address (street name, flat number, etc.) |
| addressLine2 | String (Optional) | Landmark or additional info |
| area | String | Local area or neighborhood |
| city | String | City name |
| state | String | State/Region name |
| country | String | Country name |
| pinCode | String | Postal/ZIP code |

| geoCoordinates | Object | `{ latitude: Number, longitude: Number }` |
|---|---|---|
| canHostMatingAtHome | Boolean | If user can host the dog mating meetup at their home |
| preferredLanguage | String (Optional) | For multilingual support |
| preferredContactMethod | Enum | `'Phone'`,`'Email'`,`'In-App Chat'` |
| isVerifiedUser | Boolean | Admin-verified user (optional for trust-building) |
| about | String (Optional) | Short bio or note from the user |

## 🧠 Why These Fields?

These additions will help in:

- Better **matchmaking** (location/area-based)

- Suggesting **clinics nearby**

- Personalizing the **AI chatbot** (age, location, preferred language)

- Enabling **host/guest meetups** for dog mating

- Easier **admin filtering/exporting** of data

## 📌 Enum Meaning Breakdown

| Enum Value | Description |
| --- | --- |
| "12-18 months" | Early eligible age (female 2nd heat, young male readiness) |
| "18-24 months" | Strong fertility stage, prime starting age |
| "2-4 years" | Ideal and healthy peak mating years |
| "4-6 years" | Still acceptable if dog is healthy, but nearing senior stage |
| "6+ years" | Senior stage, should be based on vet advice |
| "No preference" | User is open to any eligible age |

Absolutely! Here's a refined list of **Dog-Related Questionnaire Fields** for the **User Profile Edit Section**, including:

- The actual **question**

- The **purpose**

- The **expected answer type**

- Example **backend field name**

- Example **answer format**

## 🐶 1. What breeds are you most interested in for mating?

- **Purpose**: Helps matchmaking engines prioritize breed compatibility.

- **Answer Type**: Multi-select dropdown

- **Field Name**: `preferredBreeds`

- **Example Answer**:
  `["Labrador", "Golden Retriever", "Beagle"]`

## 🧬 2. What qualities do you look for in a mating partner dog?

- **Purpose**: Understand behavioral and physical traits preferred.

- **Answer Type**: Multi-select checkboxes or tags

- **Field Name**: preferredPartnerTraits

- **Example Options**:

  - Friendly

  - Calm

  - Active

  - Obedient

  - Experienced

  - Vaccinated

  - Specific Color

  - Specific Size

- **Example Answer**:

  ["Friendly", "Vaccinated", "Calm"]

**🐾 3. Have you previously mated your dog? If yes, how was the experience?**

- **Purpose**: Builds trust and helps review system

- **Answer Type**:

    - Yes/No (Boolean)

        - Optional short text input

- **Field Name**:

    - **hasPreviousMatingExperience**

    - **matingExperienceNotes (optional)**

**Example Answer**:

```json
{
    "hasPreviousMatingExperience": true,
    "matingExperienceNotes": "Yes, it went smoothly with 4 healthy puppies."
}
```

## 🏡 4. What's your preferred mating setup?

- **Purpose**: Helps with logistical compatibility during matchmaking.

- **Answer Type**: Radio buttons / Enum

- **Field Name**: `preferredMatingLocation`

- **Options**:

    - "At my home"

    - "At partner's home"

    - "At neutral clinic"

    - "No preference"

- **Example Answer**
  `"At neutral clinic"`

## 🎯 5. How important are the following factors in choosing a partner for your dog?

- **Purpose**: Refines AI matchmaking weightage

- **Answer Type**: Either Yes/No (Boolean) or Rank 1–5

- **Field Name**: `matingPartnerCriteriaImportance`
  ✅ Alternatively, as Yes/No (simpler to implement initially):

```
{
  "vaccinationStatus": true,
  "neuteredStatus": false,
  "healthHistoryTransparency": true,
  "ownerCommunicationLevel": true,
  "locationProximity": true
}
```

## 🩺 Doctor Model Fields

| Field Name | Type | Description |
| --- | --- | --- |
| doctorId | UUID | Unique ID for the vet |
| name | String | Full name of the vet |
| email | String | Email address for communication |
| phone | String | Contact number |
| profileImage | String (URL) | Optional profile picture |
| qualification | String | e.g., BVSc, DVM, MVSc, PhD |
| specialization | Enum[] | e.g., General Vet, Breeding, Surgery, Dermatology, Orthopedics |
| yearsOfExperience | Number | How many years the doctor has been practicing |
| licenseNumber | String | Vet council license/registration number |
| licenseIssuingAuthority | String | e.g., Veterinary Council of India, AVMA, etc. |

| | | |
|---|---|---|
| clinicId | UUID | Associated clinic (if any) |
| consultationFee | Number | Fee per consultation |
| availableDays | Array[String] | e.g., ["Monday", "Wednesday", "Friday"] |
| availableTimeSlots | Array[String] | e.g., ["10:00 AM - 12:00 PM", "4:00 PM - 6:00 PM"] |
| location | Object | { area, city, state, pinCode, geoCoordinates } |
| ratings | Number (Optional) | Average user rating (e.g., 4.6) |
| reviews | Array[Object] | Feedback from pet owners |
| isVerified | Boolean | Admin-approved vet status |
| about | String | Short description or vet bio |

## 🧠 Optional Enhancements (for future AI integration)

| Field | Description |
|---|---|
| languagesSpoken | To help users choose vets based on language preferences |
| isAvailableForEmergency | Boolean for 24/7 or emergency vet support |
| teleconsultationAvailable | If they support online consultations |
| tags | Tag-based keywords for filtering/search (e.g., "dog fertility", "puppy vaccination" will be added by admin) |

# 💬 Group Chat with Invitation Feature (Additional feature)

- 🔹 **Feature Name:** `Group Chat + Invitation System`

## ✅ Purpose:

To enable users to collaborate and communicate in community-based group chats and invite others directly to join relevant groups (e.g., breed communities, mating discussions, vet Q&A).

## 👥 Who Can Use It:

- Registered **Dog Owners**

- **Veterinarians** and **Pet Clinics**

- **Admins** (for special announcement or support groups)

## 🧩 Key Functionalities:

| Feature | Description |
| --- | --- |
| **Create Group** | Users can create groups by setting a title, image, description, category, and privacy type (Public/Private). |
| **Join/Leave Group** | Members can join public groups instantly or request access to private groups. Leaving is allowed anytime. |
| **Invite to Group** | Send invites via username, email, or generate a secure link. |
| **Accept/Decline Invite** | Users receive notifications and can choose to join or ignore invites. |
| **Pending Invites** | Users can view a list of invitations they've sent or received. |
| **Manage Group Members** | Group Admin/Creator can approve join requests, remove members, or promote/demote roles. |
| **Group Roles** | Roles include `Admin`, `Moderator`, and `Member` with varying permissions. |
| **Group Chat** | Real-time messaging among group members using WebSockets or Socket.io. |
| **Group Notifications** | Notifications for invites, new messages, join requests, and admin actions. |

## 🔗 Group Invitation Methods:

| Method | Usage |
| --- | --- |
| Username-based | Invite registered users via their username (autocomplete suggestion dropdown) |
| Email-based | Invite users via email (with sign-up prompt if not registered) |
| Shareable Link | Generates a secure invite link that expires in X hours or after Y uses |

## 🔒 Moderation & Safety Features

| Feature | Description |
| --- | --- |
| Report Group | Members can report inappropriate or abusive groups. Reports go to admins for review. |
| Ban User from Group | Admins/Moderators can permanently or temporarily ban users from rejoining. |
| Delete Group | Only the **Group Creator** or **Admin** (with permission) can delete a group. |
| Kick/Remove Member | Admins or Moderators can remove members from the group. |
| Block Invites from User | Users can block specific people from sending group invites. |
| Mute Notifications | Users can mute group notifications. |
| Audit Log (Admin View) | Admins can see logs of member joins, bans, reports, and deletions. |

## 🗄️ Database Tables (Suggested Schema)

| Table | Key Fields |
|---|---|
| **groups** | id, title, description, image, privacy, createdBy, category, createdAt |
| **group_members** | id, groupId, userId, role, joinedAt, status (active, banned) |
| **group_invites** | id, groupId, senderId, receiverId/email, token, status, expiresAt |
| **group_reports** | id, groupId, reporterId, reason, createdAt |
| **group_audit_log** | id, groupId, action, performedBy, targetUserId, timestamp |