Capstone Project Proposal: Waggr

Dog Walker Match App

---

1. Tech Stack

I will use a **full-stack JavaScript approach** with:

- **Frontend**: React.js (with React Router for page navigation)
- **Backend**: Node.js with Express.js for building API endpoints
- **Database**: PostgreSQL for storing user accounts, dog profiles, and bookings
- **Other Tools**:
    - Authentication: JWT for secure login and user sessions
    - API Integration: Petfinder API for adoptable dogs and Google Maps API for shelter locations
    - Bcrypt for password hashing

---

2. Project Focus

The project will be an **evenly focused full-stack application**:

- **Frontend**: Prioritizes a modern, user-friendly interface with swipe/match functionality similar to dating apps.
- **Backend**: Handles user accounts, shelters, scheduling, and integrates external APIs.

---

3. Platform

This will be a **responsive web application** that users can access via desktop browsers. Future enhancements could include turning it into a Progressive Web App (PWA) for mobile-like experience.

---

4. Project Goal

The goal is to **connect dog lovers with shelters** by allowing users to match with dogs available for walks.

- **Problem**: Shelters often struggle to find volunteers to walk their dogs, while many people want a way to engage with dogs without full adoption.
- **Solution**: Provide a matchmaking experience that encourages volunteerism and makes it fun and easy for users to book walking sessions with dogs near them.

---

5. Target Users

- **Primary Users**: Dog lovers who want to volunteer for dog walking.
- [redacted]
- **Demographic**:
    - Ages 18–45
    - Urban or suburban dwellers near animal shelters
    - People who cannot own dogs but love interacting with them

---

6. Data Source

- **Petfinder API**: Fetch dog profiles and shelter details (name, address, available dogs)
- **Google Maps API**: Show nearby shelters and walking routes (if there's time)

- **Custom Database**:
    - Users (name, email, preferences)
    - Dogs (sourced from Petfinder + shelter-specific data)
    - Bookings (scheduling walks)

---

7. Approach to Building the Project

- **Backend**:
    - Build REST API with Express.js for user authentication and booking management
    - Integrate Petfinder API to display real-time dog profiles
    - Connect Google Maps API for geolocation and directions
- **Frontend**:
    - Create React components for dog cards, swipe functionality, booking forms
    - Add authentication flow (register/login)
- **Database**:
    - Tables for users, dogs, shelters, and bookings
- **Optional Features**:
    - Photo upload after walks
    - Push notifications for confirmed bookings
    - Reviews and ratings for shelters

New Additions:

For Frontend use CSS for styling.

Tailwind CSS or Chakra UI
Framer Motion for UI animations (for swipes, and tail wag animation?)

Further implementation

Knex.js manages the postgresSQL migration

Hosting - render.com , supabase for database and authentication

Jest and React Testing Library - Write the unit tests for the code

Consider rate limiting (express-rate-limit), input validation

Consider rate limiting (express-rate-limit), input validation (e.g., **Joi**), and CORS setup.

If the user goes on 5 walks, they get a badge - gamified

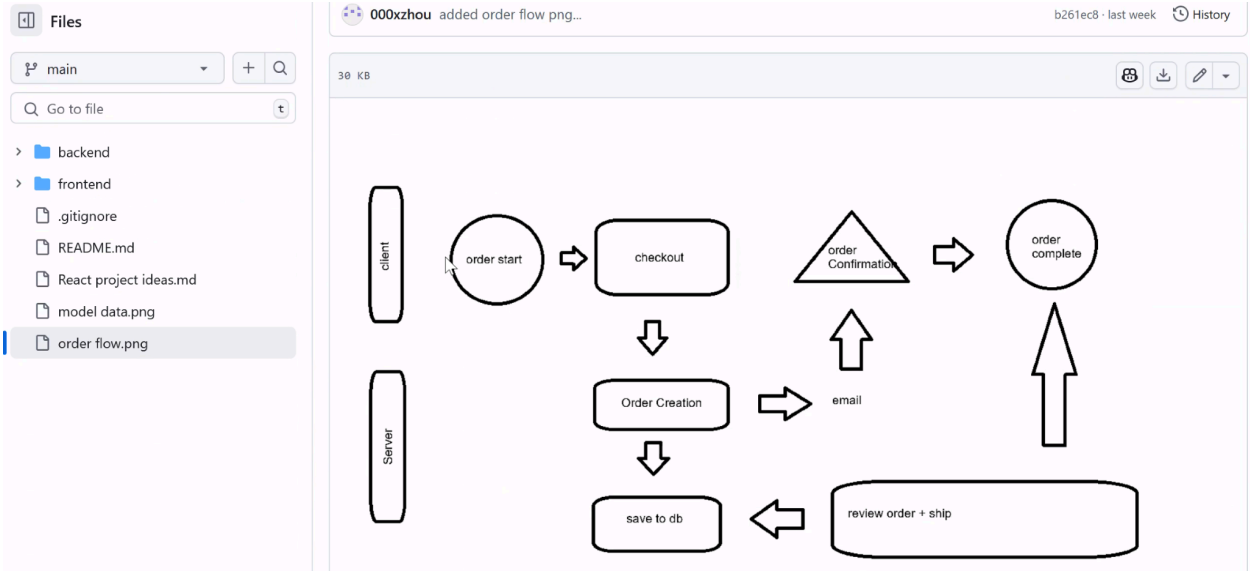
Sort dogs by size, breed, energy,

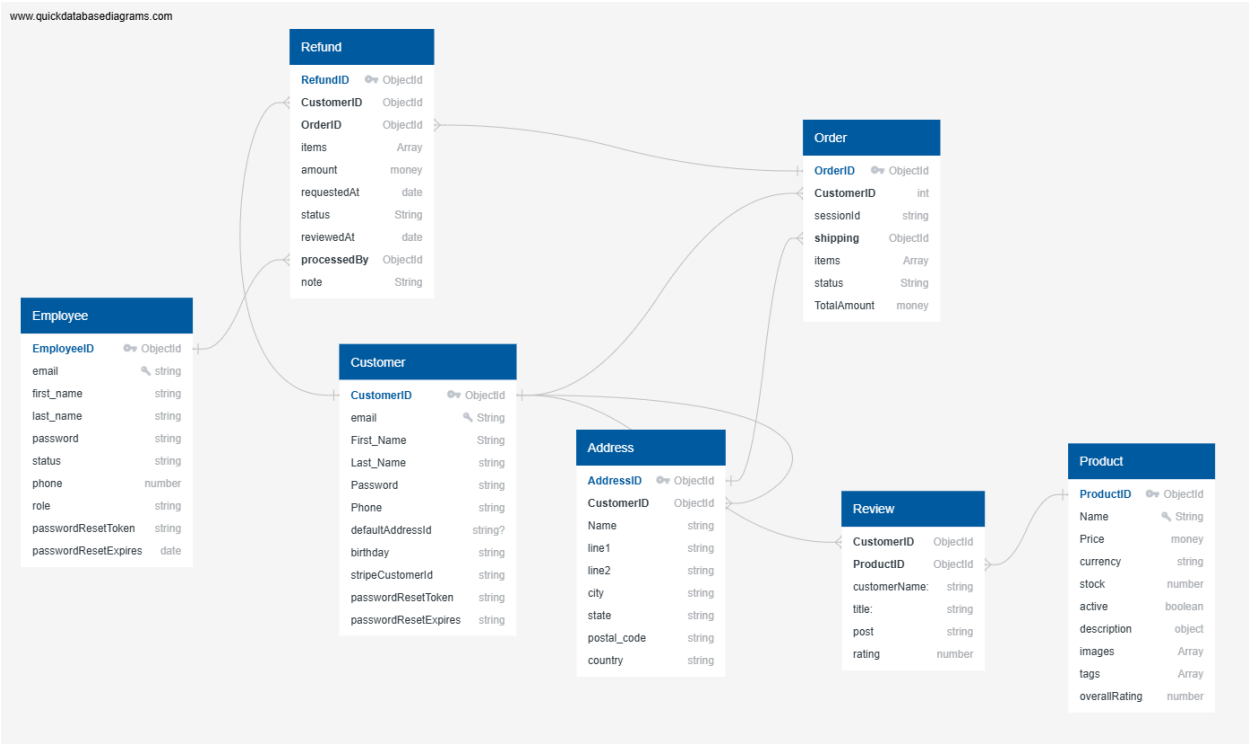
Dashboard for shelters to view bookings.

Distinct user types:

Walker,
Shelter admin (approve walk requests)
Volunteer
Superadmin (me the webmaster)

Time slot, availability, and documentation





https://github.com/000xzhou/Nichirin
https://www.quickdatabasediagrams.com/