Software Requirements Specification (SRS)

Project Title: SafeTails

Course: CSE470 - Software Engineering (BRACU)

Author: Ahsan Habib

Email: ahsan.habib1@g.bracu.ac.bd

ID: 22201027

Date: 04 August, 2025

Semester: Summer 25

Section: 14

Document Version: Sprint 1

1. Introduction

1.1 Purpose

This Software Requirements Specification (SRS) outlines the complete functional and non-functional requirements for SafeTails — a full-stack web application designed to assist in pet welfare through lost pet recovery, adoption, fostering, and emergency services using geolocation and community-driven efforts.

1.2 Scope

SafeTails will allow users to:

- Report and locate lost, abandoned, or injured pets.
- Submit pet adoption and foster requests.
- View and interact with location-based emergency vet contact directories.

- Receive geo-alerts and filter posts by pet type, location, and need.
- Access a moderation-enabled community space overseen by an admin panel.

1.3 Definitions, Acronyms, and Abbreviations

- MVC: Model View Controller
- NGO: Non-Governmental Organization
- Leaflet.js: JavaScript library for interactive maps
- OSM: OpenStreetMap
- JWT: JSON Web Token.
- UI: User Interface.
- API: Application Programming Interface.

1.4 References

https://petcolove.org/lost/

https://www.pawboost.com/

https://pet911.co.uk/

https://findpet.com/

https://leafletjs.com/

https://resend.com/

1.5 Overview

This document outlines the product’s overall features, user roles, system architecture, development timeline, and success metrics. Sections 2–7 continue in the full version with all technical details, features, architecture, and development plan as previously written.

2. Overall Description

2.1 Product Perspective

SafeTails is a standalone system based on the MERN-like architecture using:

- Frontend: Next.js, TypeScript,Tailwind CSS, Leaflet.js
- Backend: Express.js, MongoDB, Axios, Node.js
- Geolocation: Leaflet + OSM

- Email Service : Resend Api + Local server (OTP Verification)

Optional integrations include Firebase for media hosting and push notifications.

2.2 Product Features

1. User Profile registration / Deletion (Creating a Profile, Deleting a profile)
2. UI dynamically changing based on Guest user or Logged in user or a Vet.
3. Profile management (Editing / Updating Profile)
4. Create and manage posts (lost, adoption, foster, etc.)
5. Upload Multiple photos for posts
6. Map integration for pet location pinning (Leaflet.js + OpenStreetMap)
7. Emergency vet contact directory in nearby areas (searchable and clickable)
8. Search & filtering (by type, location, pet category)
9. Geolocation-based alert system
10. Temporary foster request and response feature
11. Post reporting and moderation tools
12. Admin dashboard for reviewing and flagging inappropriate content
13. Sending Email to Registered user for OTP verification

2.3 User Classes and Characteristics

- Guest: Can browse public pet posts including pet’s last known location and picture
- Registered User: Can create/manage posts, Find vets, request fosters, Request for

veteran consultant and receive alerts

- Veterinarian: Can provide emergency contact info and assist in verification, Can manage and respond to vet requests.

- Admin: Manages reports, user (Pet owner, Vet) accounts, and platform moderation

2.4 Operating Environment

Frontend: Modern browsers

Backend: Node.js + Express on cloud

Database: MongoDB

Map Service: Leaflet + OpenStreetMap

Mobile Compatibility: Android/iOS browsers -

2.5 Constraints

• Must be operational within 8–10 weeks

• Use free APIs/services (Leaflet, OSM, Resend)

• Open-source or educational licenses only

• Limited cloud deployment resources

2.6 Assumptions and Dependencies

• Users allow location access

• Emergency vet info is verified

• Community posts are moderated

• Image hosting may rely on Firebase

3. System Requirements

3.1 Functional Requirements

• FR-1: User registration, login

• FR-2: Role-based dashboards

• FR-3: Sending Email With OTP verification

- FR-4: Post management with image uploads
- FR-5: Location tagging
- FR-6: Post search/filter
- FR-7: Foster request features
- FR-8: Emergency vet contact directory
- FR-9: Admin DashBoard Moderation (Reporting and moderation tools)
- FR-10: Vet Dashboard Moderation (Block user, List all users)
- FR-11: Send / Receive Notifications based on Geo-location
- FR-12: Profile management (Edit / Update) Own profile.

3.2 Non-Functional Requirements

- NFR-1: Login with OTP verification with JWT authentication
- NFR-1: Page load < 2s
- NFR-2: Map render < 1.5s
- NFR-3: HTTPS only
- NFR-4: Token-based auth
- NFR-5: ≥99.5% uptime
- NFR-6: Daily backups
- NFR-7: Modular MVC code
- NFR-8: Search and filter optimization

3.3 External Interface Requirements

- UI-1: Responsive UI for all devices
- UI-2: Role-specific dashboards
- HI-1: Standard server/cloud
- SI-1: OSM API
- SI-2: Firebase (optional)

- CI-1: RESTful API (Axios)
- CI-2: Optional notifications

4. Technology Stack & Architectural Overview

4.1 Stack

Frontend: Next.js, Tailwind CSS, TypeScript

Backend: Express.js, MongoDB

Map: Leaflet.js + OSM

Email Service: Resend + Local server

Media: Firebase (optional)

4.2 Architecture

- Presentation Layer: React components (Next.js)
- Logic Layer: Express routes/controllers
- Data Layer: Mongoose models
- Map Layer: Leaflet integration

5. Tentative Agile Development Plan

Sprint 1 (Week 1–2):

1. Environment Setup, User account (Pet Owner, Vet) Registration,
2. Sending Users Email for Verification, Congratulations mail.
3. Role based Dashboard (Pet Owner , Vet dashboard only)
4. New Pet Post creation

Sprint 2 (Week 3–4):

1. Admin Creation, Dashboard Management
2. Profile management (Edit / Update) - Own user profile.
3. Map pinning With Leaflet and OSM
4. Filter / Search Posts for 4 criteria

Sprint 3 (Week 5–6):

1. Emergency vet contacts
2. Image upload of Pets
3. Admin, Vet Dashboard Management

Sprint 4 (Week 7–8):

1. Geo-location based post alerts / Notifications
2. Final touchup / Debugging Entire Project
3. Build Project for deployment
4. Deploy Project (Optional)

6. Acceptance Criteria

- Posts with location and images function correctly
- Foster requests are stored and processed
- Admin can moderate content
- System meets uptime and security standards

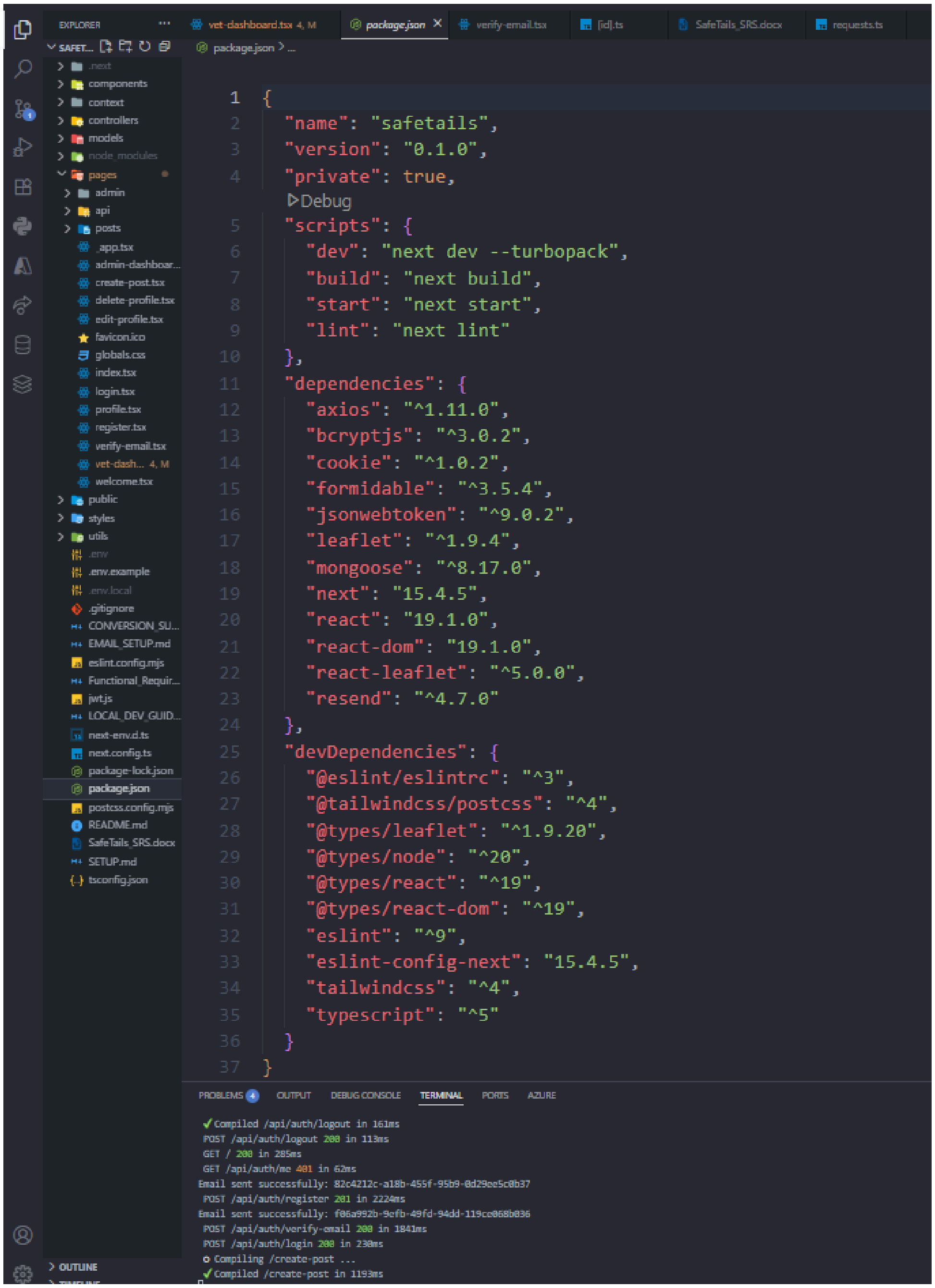
7. Conclusion

SafeTails is a scalable, secure, and community-centric platform for pet welfare. It combines geolocation, role-based features, and moderation tools to address critical needs around pet safety and rescue.

Sprint - 1

Feature 1: Environment setup and Role based Dashboard (Pet Owner, Admin dashboard and Vet Dashboard)

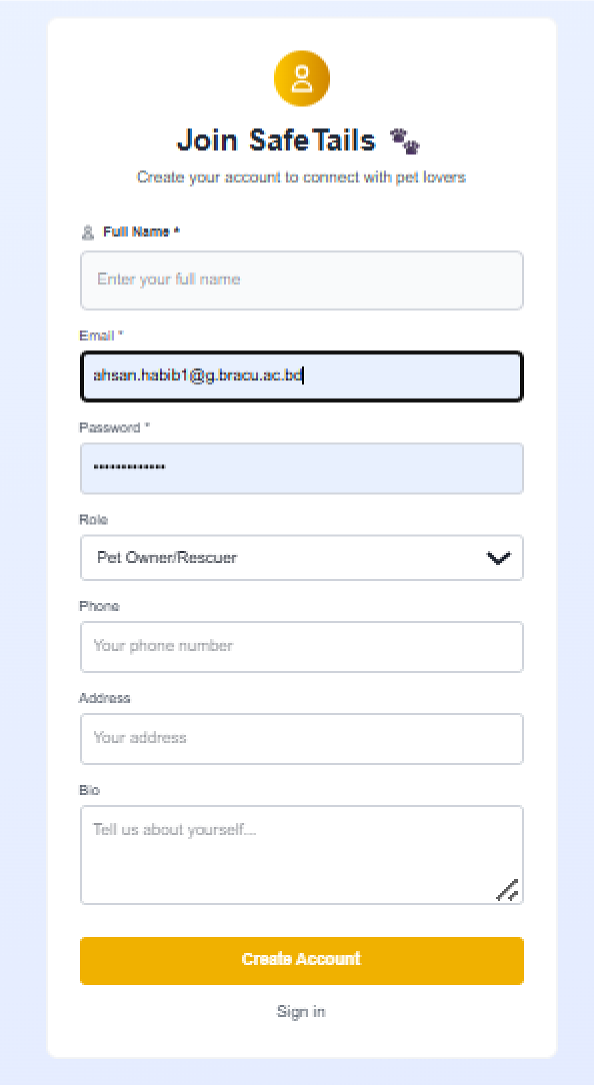
1.1 (Using npm and npx - Used Next.js and TypeScript For Frontend, .env file to store database and API keys) Using .gitignore to stop them from pushing sensitive info to github, All the Dependencies that i will be using is mentioned here



```json
{
  "name": "safetails",
  "version": "0.1.0",
  "private": true,
  "scripts": {
    "dev": "next dev --turbopack",
    "build": "next build",
    "start": "next start",
    "lint": "next lint"
  },
  "dependencies": {
    "axios": "^1.11.0",
    "bcryptjs": "^3.0.2",
    "cookie": "^1.0.2",
    "formidable": "^3.5.4",
    "jsonwebtoken": "^9.0.2",
    "leaflet": "^1.9.4",
    "mongoose": "^8.17.0",
    "next": "15.4.5",
    "react": "19.1.0",
    "react-dom": "19.1.0",
    "react-leaflet": "^5.0.0",
    "resend": "^4.7.0"
  },
  "devDependencies": {
    "@eslint/eslintrc": "^3",
    "@tailwindcss/postcss": "^4",
    "@types/leaflet": "^1.9.20",
    "@types/node": "^20",
    "@types/react": "^19",
    "@types/react-dom": "^19",
    "eslint": "^9",
    "eslint-config-next": "15.4.5",
    "tailwindcss": "^4",
    "typescript": "^5"
  }
}
```

```
 ✓ Compiled /api/auth/logout in 161ms
 POST /api/auth/logout 200 in 113ms
 GET / 200 in 285ms
 GET /api/auth/me 401 in 62ms
Email sent successfully: 82c4212c-a18b-455f-95b9-0d29ee5c0b37
 POST /api/auth/register 201 in 2224ms
Email sent successfully: f06a992b-9efb-49fd-94dd-119ce068b036
 POST /api/auth/verify-email 200 in 1841ms
 POST /api/auth/login 200 in 230ms
 ○ Compiling /create-post ...
 ✓ Compiled /create-post in 1193ms
```

1.2 User Account registration - PetOwner, Vet Registration:



Join SafeTails 🐾

Create your account to connect with pet lovers

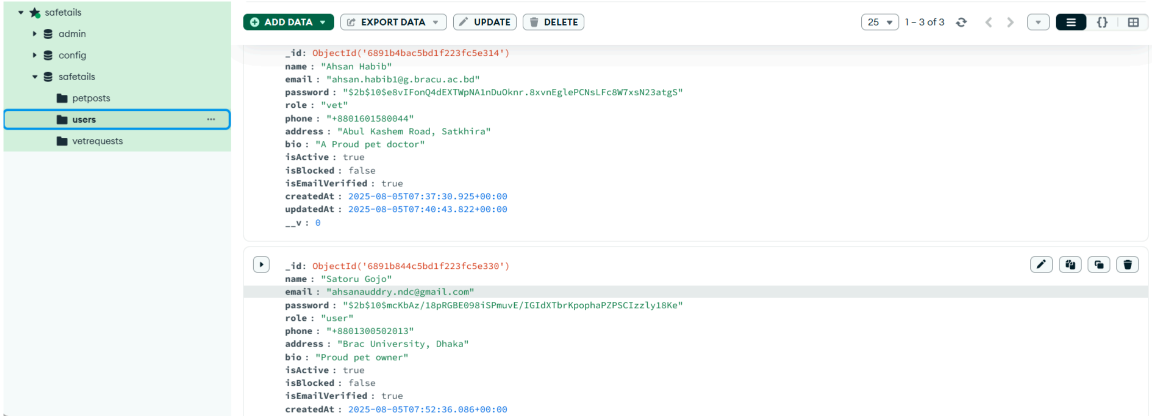
Full Name *

Email * ahsan.habib1@g.bracu.ac.bd

Password * ••••••••••••••

Role: Pet Owner/Rescuer

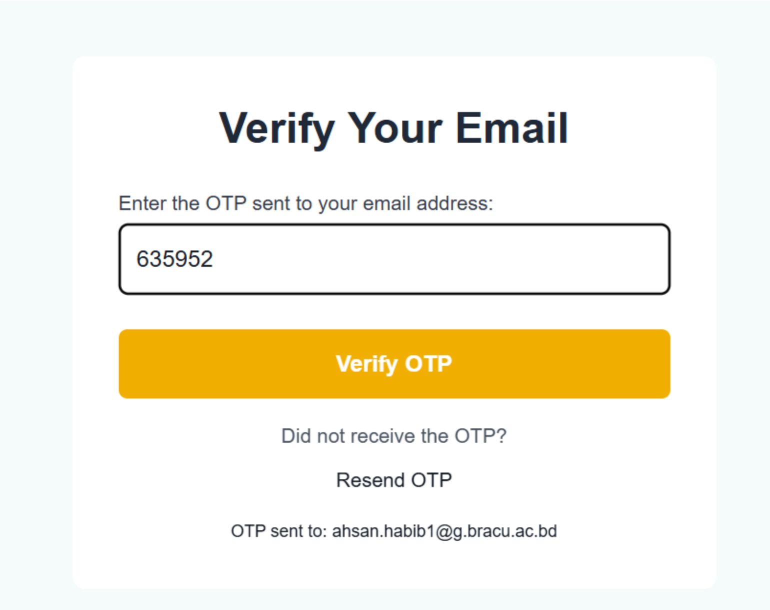
1.3 Values Stored In MongoDB user Collection:



```
_id: ObjectId('6891b4bac5bd1f223fc5e314')
name : "Ahsan Habib"
email : "ahsan.habib1@g.bracu.ac.bd"
password : "$2b$10$e8vIFonQ4dEXTWpNA1nDuOknr.8xvnEglePCNsLFc8W7xsN23atgS"
role : "vet"
phone : "+8801601580044"
address : "Abul Kashem Road, Satkhira"
bio : "A Proud pet doctor"
isActive : true
isBlocked : false
isEmailVerified : true
createdAt : 2025-08-05T07:37:30.925+00:00
updatedAt : 2025-08-05T07:40:43.822+00:00
__v : 0
```

```
_id: ObjectId('6891b844c5bd1f223fc5e330')
name : "Satoru Gojo"
email : "ahsanauddry.ndc@gmail.com"
password : "$2b$10$mcKbAz/18pRGBE098iSPmuvE/IGIdXTbrKpophaPZPSCIzzly18Ke"
role : "user"
phone : "+8801300502013"
address : "Brac University, Dhaka"
bio : "Proud pet owner"
isActive : true
isBlocked : false
isEmailVerified : true
createdAt : 2025-08-05T07:52:36.086+00:00
```

Feature 2: OTP verification - With 10 minutes Expiry time



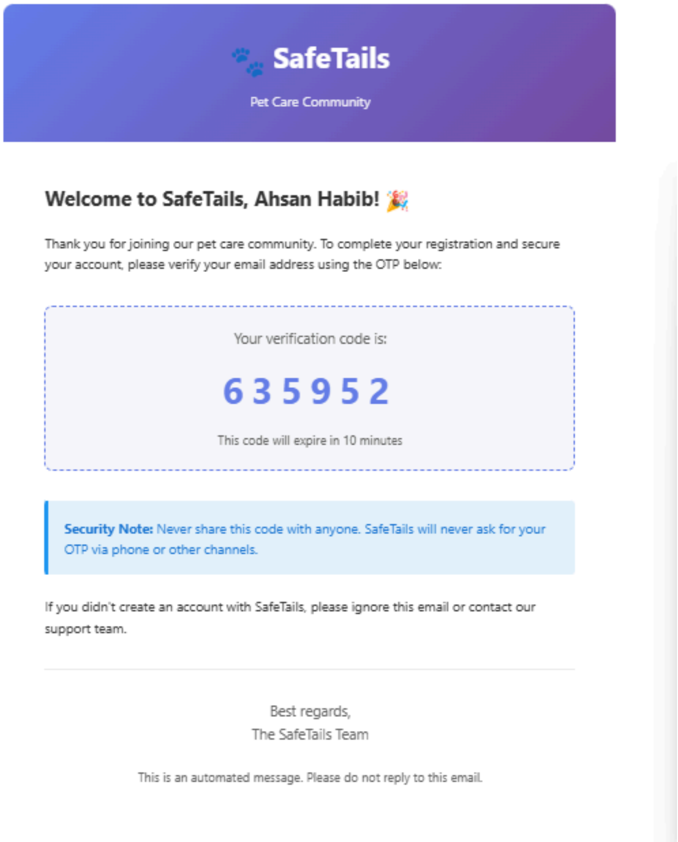
Verify Your Email

Enter the OTP sent to your email address:

635952

Did not receive the OTP?

OTP sent to: ahsan.habib1@g.bracu.ac.bd



🐾 SafeTails

Pet Care Community

Welcome to SafeTails, Ahsan Habib! 🎉

Thank you for joining our pet care community. To complete your registration and secure your account, please verify your email address using the OTP below:

Your verification code is:

635952

This code will expire in 10 minutes

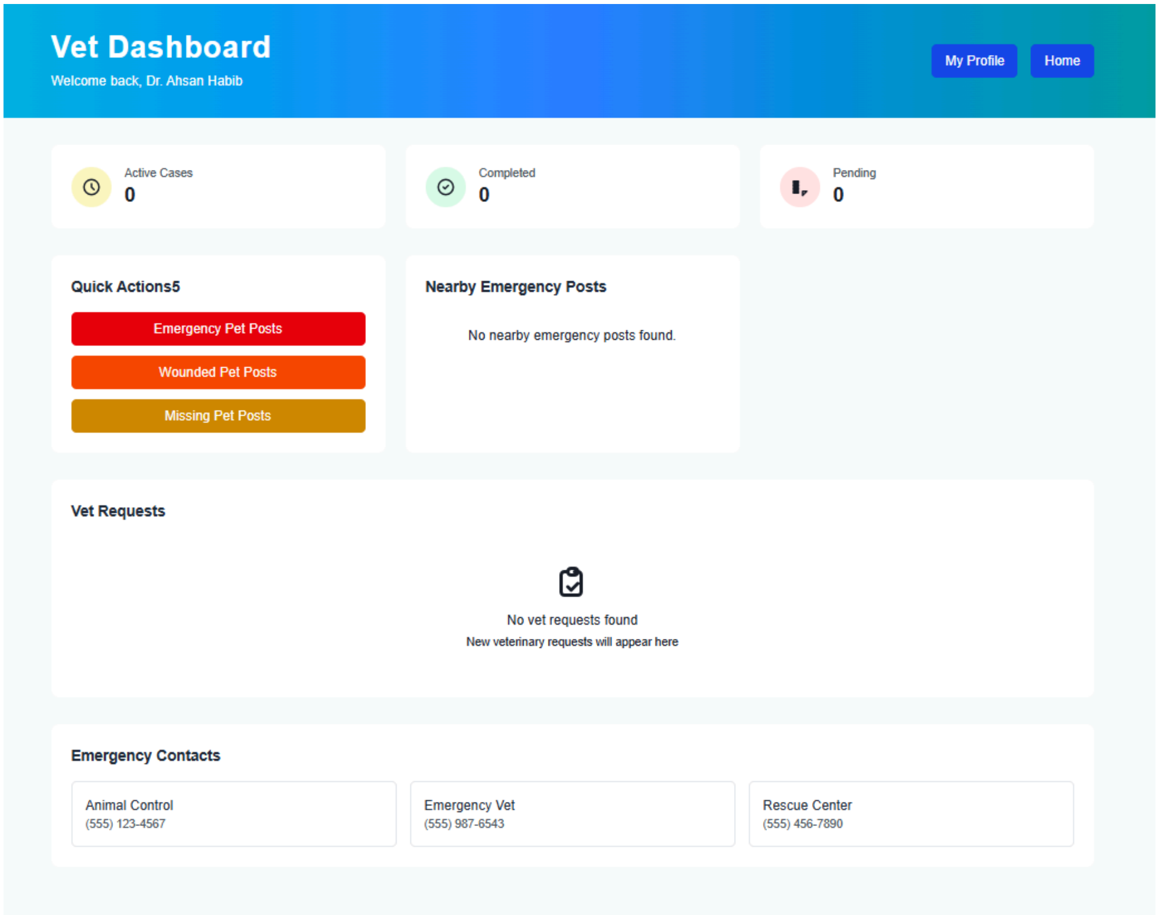
Security Note: Never share this code with anyone. SafeTails will never ask for your OTP via phone or other channels.

If you didn't create an account with SafeTails, please ignore this email or contact our support team.

Best regards,
The SafeTails Team

This is an automated message. Please do not reply to this email.

Feature 03: Dashboard creation For Users (Vet Dashboard, Pet dashboard)



Nearby Emergency Posts

No nearby emergency posts found.

Vet Requests

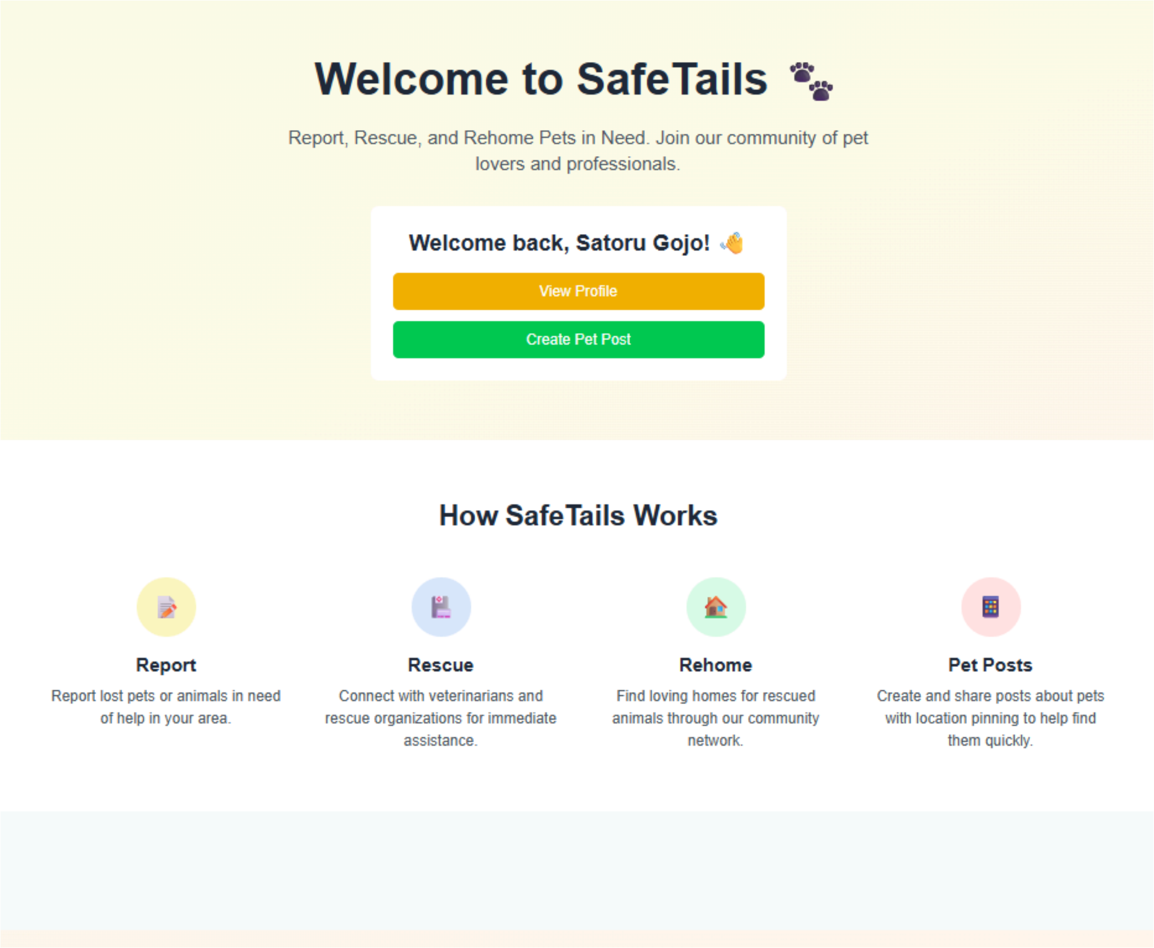
No vet requests found

New veterinary requests will appear here

Emergency Contacts

Animal Control (555) 123-4567	Emergency Vet (555) 987-6543	Rescue Center (555) 456-7890

Pet Owner Dashboard



Welcome to SafeTails 🐾

Report, Rescue, and Rehome Pets in Need. Join our community of pet lovers and professionals.

Welcome back, Satoru Gojo! 👋

View Profile

Create Pet Post

How SafeTails Works

Report

Report lost pets or animals in need of help in your area.

Rescue

Connect with veterinarians and rescue organizations for immediate assistance.

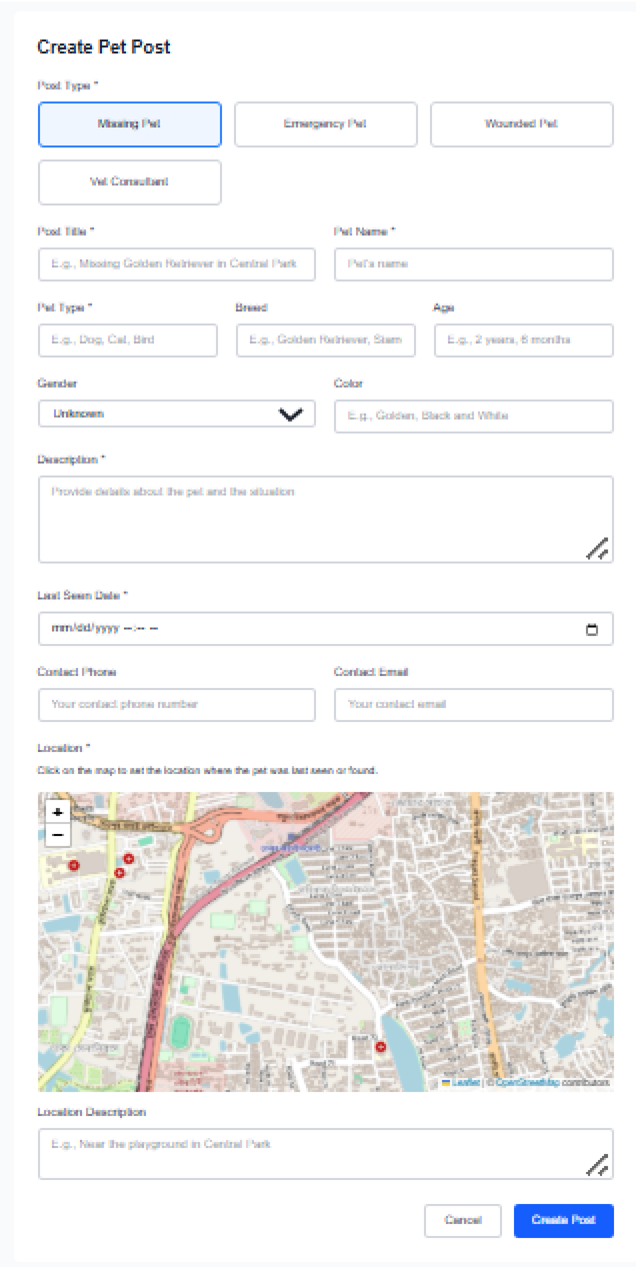
Rehome

Find loving homes for rescued animals through our community network.

Pet Posts

Create and share posts about pets with location pinning to help find them quickly.

Feature 04: New Pet Post Creation of 4 types:



Create Pet Post

Post Type *

Missing Pet | Emergency Pet | Wounded Pet | Vet Consultant

Post Title *: E.g., Missing Golden Retriever in Central Park

Pet Name *: Pet's name

Pet Type *: E.g., Dog, Cat, Bird

Breed: E.g., Golden Retriever, Siam

Age: E.g., 2 years, 6 months

Gender: Unknown

Color: E.g., Golden, Black and White

Description *: Provide details about the pet and the situation

Last Seen Date *: mm/dd/yyyy --:-- --

Contact Phone: Your contact phone number

Contact Email: Your contact email

Location *

Click on the map to set the location where the pet was last seen or found.

Location Description: E.g., Near the playground in Central Park