

# Public Users vs Registered Users

## Overview

This document summarizes the differences between Public Users (unregistered/anonymous visitors) and Registered Users (authenticated users) in the Dog Rescue application, reflecting the current implementation.

## Key Differences

Capability	Public User (Unregistered)	Registered User (Authenticated)
Browse dogs list	Yes (/dogs)	Yes
View dog details	Yes (/dogs/:id)	Yes
Submit rescue request	Yes (/rescue)	Yes (with optional user info attached)
Submit volunteer application	Yes (public form)	Yes (same public form)
Register account	N/A	Yes (/register)
Login	N/A	Yes (/login)
Apply for adoption	No	Yes (/adopt/:dogId)
View personal dashboard	No	Yes (/user-dashboard)
View my adoption applications	No	Yes
Withdraw my adoption application	No	Yes (if status Pending/Under Review)
Email notifications for adoption status	N/A	Yes (Approved/Rejected)
Email confirmation for volunteer submission	Yes	Yes
Access admin dashboard	No	No (Admins only)

## Access Control and Enforcement

- Frontend routing
  - Rescue submissions: public route (/rescue).
  - Adoption application and user dashboard: protected by user auth guard.
  - Admin dashboard: protected by admin auth guard.
- Backend APIs
  - Rescue submissions: public POST (/api/rescue-submissions).

- Volunteer applications: public CRUD endpoints (intended to be admin-managed but currently not auth-protected).
- Adoption applications:
  - Submit, list mine, withdraw: require user auth (`/api/applications` with user middleware).
  - Admin review/update: intended admin-only; currently not protected and should be hardened.

## Identity, State, and Tokens

- Public User
  - No account, no token.
  - Can submit rescue/volunteer data with contact info only.
- Registered User
  - Account with role `user` and status `active/inactive`.
  - JWT (`userToken`) stored client-side; backend enforces `active` status.
  - Optional user linkage on rescue submissions (user id/name/email included when logged in).

## Notes and Recommendations

- Security hardening (recommended): protect admin-intended endpoints with admin middleware; consider CAPTCHA/rate limiting for public submissions.
- UX: when a logged-in user submits a rescue, prefill contact info from their profile.