System Architecture – Overview

- The project follows a classic web-based client-server architecture.

- Main components:
- • Web Frontend (Client)
- • REST Backend API
- • Relational Database

- Architecture is designed for scalability, maintainability, and clear separation of concerns.

Frontend Architecture

- Technology stack:
- • Vite + React
- • TypeScript

- Responsibilities:
- • User interface and user experience
- • Authentication flow handling
- • Wizard-based support case creation
- • Chat UI and case status visualization

- Frontend communicates with backend via REST APIs.

Backend Architecture

- Technology stack:
- • NestJS (Node.js)
- • RESTful API

- Responsibilities:
- • Authentication & authorization
- • Business logic for support cases
- • Case assignment and status management
- • Chat message handling

- Layered architecture: Controllers, Services, Modules.

Database Architecture

- Database:
- • PostgreSQL hosted on Neon

- Core entities:
- • Users (Kitten / Guardian)
- • Support Cases
- • Case Statuses
- • Chat Messages

- Relational model ensures data integrity and consistency.

Deployment & Infrastructure

- Hosting:
- • Frontend: Vercel
- • Backend: Vercel
- • Database: Neon (PostgreSQL)

- Communication:
- • HTTPS between frontend and backend
- • Secure database connections

- System is deployed in a public environment and ready for real usage.
  https://paw-path-one.vercel.app/