Product Requirements Document (PRD) - T4Alerts

Project Name: T4Alerts

Date: 2025-12-18

Author: Antigravity AI

1. Introduction

T4Alerts is a centralized web application designed to replace the current notification system (Email, SMS, Slack). It aims to provide a secure and user-friendly interface for monitoring application logs and SSL certificate statuses for the T4App ecosystem.

2. Key Objectives

- Centralize verification and log monitoring for multiple web applications.
- Proactively monitor SSL certificate expirations with visual severity indicators.
- Provide analytics on recurrent errors to assist in troubleshooting.
- Secure access via JWT authentication.

3. Core Features & Requirements

3.1 Authentication

The system requires a secure login mechanism to restrict access to authorized personnel only.

- User Registration: Ability to create new admin users.
- Login: Secure login obtaining a JSON Web Token (JWT).
- Session Management: JWT must be used for authorization in all protected routes.

3.2 Main Dashboard Structure

Upon logging in, the user will be presented with a main dashboard divided into two primary sections:

- Log Alerts: For application error monitoring.
- SSL Alerts: For certificate expiration monitoring.

3.3 Log Alerts Section

This section allows users to inspect logs collected from various applications. The data source will include the following applications as defined in the configuration:

- GoTo Logistics (driverapp.goto-logistics.com)

- GoExperior (driverapp.goexperior.com)
- KLC T4App (klc.t4app.com)
- AccurateCargo T4App (accuratecargo.t4app.com)
- Broker GoTo Logistics (broker.goto-logistics.com)
- KLC Crossdock T4App (klccrossdock.t4app.com)

Functional Requirements:

- Selector: Dropdown or menu to choose one of the 6 monitored applications.
- Log View: Display list of logs ordered by Date and Repetition count.
- Classification: Visual distinction or tabs for 'Errors' vs 'Uncontrolled Errors'.
- Detail View: Clicking a log should show the full details (traceback, context) similar to the current email reports.

3.4 Analytics & Statistics

To help identify stability issues, the system will provide visual analytics for each application.

- Route: Dedicated analytics view per web application.
- Metric: Most recurrent 'Uncontrolled Errors' per day.
- Visualization: Bar chart displaying the frequency of specific error types over time.
- Goal: Quickly spot spikes in specific errors.

3.5 SSL Alerts Section

A dedicated view to check the SSL health of all domains at a glance.

- List View: Show all domains with their expiration status.
- Visual Severity: Use the defined color coding (Red < 8 days, Mustard < 30 days, Green > 30 days).
- Actions: Option to trigger a manual re-check.

4. Technical Stack Proposal

- Frontend: React.js or Vue.js (SPA) for a responsive dashboard.
- Backend: Python (FastAPI or Flask) or Node.js.
- Database: PostgreSQL for storing users, historical logs, and analytics data.
- Authentication: JWT (JSON Web Tokens).
- Charting: Chart.js or Recharts for the analytics bar charts.

5. Non-Functional Requirements

- Performance: Log lists should load with pagination to handle large datasets effectively.
- Security: Passwords must be hashed (e.g., bcrypt). API endpoints must be protected behind JWT middleware.

- Reliability: The system should be highly available.