Project Title: AI-Powered Log Triage and Security Alert Aggregator for Fedora

Personal Information:
Name: Ritesh Murmu
Email: [riteshmurmu11@gmail.com]
GitHub/Portfolio:
[https://github.com/Ritesh-cloud?tab=overview&from=2025-04-01&to=2025-04-13]
University: Birla Institute of Technology, Mesra
Program: M.Sc. in Quantitative Economics and Data Science

AbstractThe aim of this project is to design and implement an AI-powered system for intelligent log triage and alert aggregation within Fedora's infrastructure. The tool will process large volumes of system and security logs, classify events using NLP/ML models, and intelligently group related alerts to reduce noise, identify patterns, and support incident response. This will streamline system monitoring and enhance the security visibility of Fedora’s systems.

Benefits to the CommunityImproved Efficiency: Automatic log triage reduces manual overhead and allows system administrators to focus on actionable events.
Enhanced Security: Early and smarter detection of critical events through ML-based aggregation.

Scalability: A modular system that can be extended across different log sources in Fedora infrastructure.

Open Contribution: This tool can be extended by other contributors and integrated into other Linux distributions and open-source systems.

DeliverablesPhase 1:
Research and identify log formats (journald, syslog, etc.).
Build pipeline to parse and clean logs.
Prototype ML model for classifying logs (normal, warning, critical).

Phase 2:
Implement clustering algorithm for grouping related alerts (unsupervised ML).
Build dashboard or CLI interface to view alert groups.
Integrate with Fedora log sources (such as Fedora Infra or Fedmsg).

Final Phase:
Implement real-time log ingestion.
Documentation and testing.

Deployment-ready release and Fedora packaging.
Detailed TimelinePeriodGoalsCommunity Bonding (May 20 – June 16)Get familiar with Fedora’s infra, interact with mentors, finalize tools/libraries (e.g., Loguru, Scikit-learn, HuggingFace, etc.)Week 1–2Set up dev environment, collect sample logs, begin parsing pipelineWeek 3–4Develop log classification model using NLP techniquesWeek 5–6Test model, evaluate accuracy, integrate with log parserWeek 7–8Mid-term evaluation; begin

clustering moduleWeek 9–10Fine-tune alert grouping, create CLI or basic UIWeek 11–12Integrate with Fedora’s infrastructure logging toolsWeek 13–FinalsFinal testing, write docs, submit final code and demosTechnical ApproachLog Preprocessing: Tokenization, filtering, and anomaly detection.

ML Classification: Using scikit-learn, XGBoost, or transformer models for semantic understanding.

Clustering: DBSCAN or K-Means for alert aggregation.
Visualization: Basic CLI or Flask dashboard to show alert categories.
Tools & Libraries: Python, Scikit-learn, Pandas, Loguru, Flask, NLTK/Spacy, systemd-journal-gatewayd.

Why Me?Background in Data Science and Machine Learning.
Experience with NLP projects and system monitoring.
Familiar with Python, Git, Linux system administration.
Passionate about open-source and already exploring Fedora systems.
Past projects include real-time detection systems using AI and dashboards using Power BI.

Future WorkPost-GSoC, I plan to help maintain and improve the tool, introduce supervised learning based on feedback loops, and explore cross-platform log integration.

Backup PlanIf real-time ingestion integration is delayed, I will focus on perfecting the log classification and aggregation module with sample log datasets, and leave real-time extension as a documented future task.