SentinAI-Ops: Technical Roadmap

Team CyberSentinel
AI for National Prosperity Hackathon
February 1 - March 15, 2026

1. Project Overview

Duration: 43 Days (6 Weeks)
Team Size: 5 Members
Total Tasks: 18 Tasks across 4 Phases
Development Model: Parallel Track Architecture

Core Deliverables

- Self-hosted Docker Compose deployment
- Three ML models (Isolation Forest, NLP Logs, Behavioral Profiling)
- FastAPI backend with auto-generated documentation
- Streamlit dashboard with real-time visualizations
- LLM-powered conversational assistant
- JWT authentication system

2. Team Structure & Roles

The team will be divided into two tracks (A and B) to enable parallel development and maximize efficiency. This separation allows both teams to work independently for the majority of the project timeline, converging only at critical integration points.

Track A focuses on backend infrastructure and machine learning models, while Track B concentrates on user interface and experience. This eliminates blocking dependencies, reduces development time by approximately 50%, and ensures that security-critical backend components remain isolated within the Docker internal network.

Track A: Backend & Intelligence (3 Members)

Primary Responsibilities:

- FastAPI backend development
- Machine Learning model implementation
- Prometheus and Loki configuration
- LLM integration and context aggregation
- Multi-channel alerting system
- Security hardening and Docker network isolation

Required Skills:

- Python (FastAPI, scikit-learn, psutil, spaCy)
- Machine Learning and NLP
- DevOps (Docker, Prometheus, Loki)
- System security and Linux administration

Key Tasks:

- T1.2: Backend API Foundation
- T1.4: Monitoring Stack Configuration
- T2.1: Model 1 - Isolation Forest
- T2.2: Model 2 - NLP Log Analyzer
- T2.3: Model 3 - Behavioral Profiling
- T3.1: LLM Conversational Assistant
- T3.3: Multi-Channel Alerting
- T3.5: Security Hardening

Track B: Frontend & User Experience (2 Members)

Primary Responsibilities:

- Streamlit multi-page dashboard
- Real-time metrics visualization
- AI chat interface development
- Admin authentication UI
- Theme customization and branding
- Demo video production

Required Skills:

- Python (Streamlit, Plotly, Pandas)

- UI/UX design principles
- Data visualization
- Frontend development

Key Tasks:

- T1.3: Frontend Dashboard Foundation
- T2.4: Dashboard UI Components
- T3.2: Chat Interface & Remediation UI
- T4.3: Demo Environment & Video

Shared Responsibilities (Both Teams)

- T1.1: Project Initialization (Day 1)
- T2.5: API-Frontend Integration (Days 6-7)
- T3.4: Authentication & Authorization (Days 10-11)
- T4.1: End-to-End Testing (Days 12-14)
- T4.4: Documentation & Guide (Days 13-14)

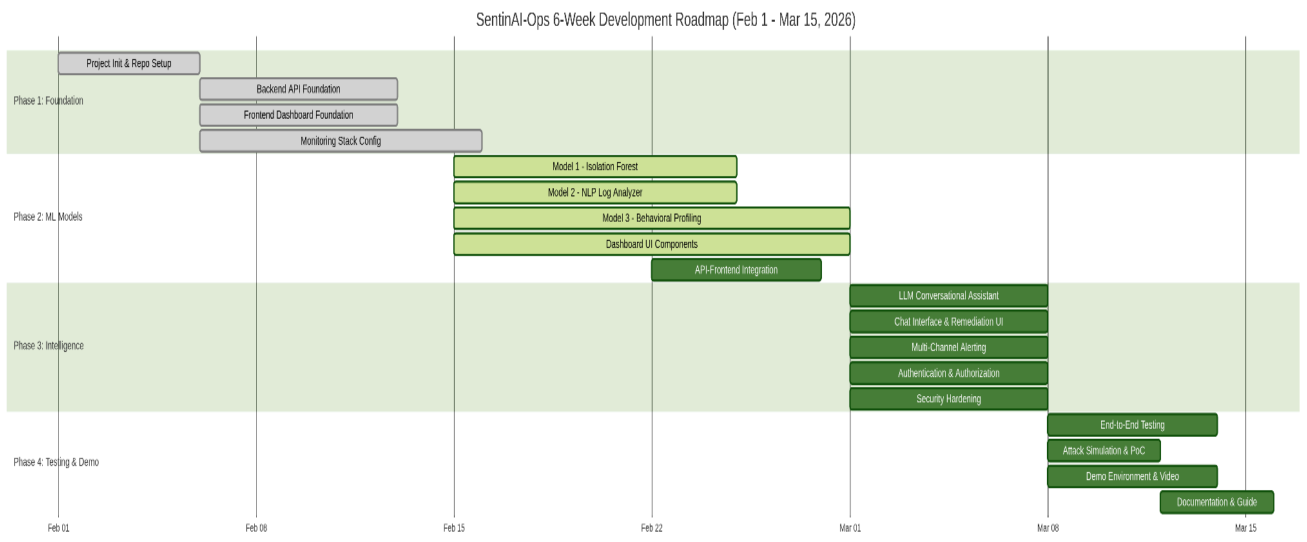
3. Technology Stack

Layer	**Technology**	**Purpose**
Frontend	Streamlit	Python-based dashboard framework with built-in charts and real-time updates
Backend API	FastAPI	Async Python framework with auto-generated OpenAPI documentation
ML Engine	scikit-learn	Isolation Forest for unsupervised anomaly detection
NLP Engine	spaCy / RapidFuzz	Lightweight NLP for log pattern recognition
Metrics Database	Prometheus	Time-series database with PromQL query language
Log Storage	Loki + Promtail	Log aggregation system designed for Prometheus integration

Orchestration	Docker Compose	Multi-container deployment with network isolation
AI Assistant	Flask + LLM API	Context aggregation and conversational threat analysis
Process Monitoring	psutil	Python library for system and process information
Alerting	Telegram, SMTP, Twilio	Multi-channel notification system

4. Development Timeline

Gantt Chart Visualization



Roadmap Diagrams: Standalone Gantt Chart & System Architecture

Phase 1: Foundation & Infrastructure (Weeks 1-2: Feb 1-14)

Objective: Establish project infrastructure and enable parallel development

Task	Name	Track	Duration	Team	Dependencies	Deliverables

T1.1	Project Init & Repo Setup	DevOps	5 days	Full Team	None	GitHub repo, docker-compose skeleton, README
T1.2	Backend API Foundation	Backend	7 days	Track A	T1.1	FastAPI structure, endpoint stubs, Swagger docs
T1.3	Frontend Dashboard Foundation	Frontend	7 days	Track B	T1.1	Streamlit multi-page setup, mock data layer
T1.4	Monitoring Stack Config	DevOps	10 days	Track A	T1.1	Prometheus, Loki, exporters configured

Milestones:

- Both teams working independently with clear interfaces
- `docker-compose up` runs all containers successfully
- Frontend uses mock data, backend connects to Prometheus/Loki

Phase 2: ML Models & Core Features (Weeks 3-4: Feb 15-28)

Objective: Implement all ML models in parallel with dashboard components

Task	Name	Track	Duration	Team	Dependencies	Deliverables
T2.1	Model 1: Isolation Forest	Backend	10 days	Track A	T1.2, T1.4	Trained model, anomaly detection API
T2.2	Model 2: NLP Log Analyzer	Backend	10 days	Track A	T1.2, T1.4	Log parsing engine, regex patterns
T2.3	Model 3: Behavioral Profiling	Backend	14 days	Track A	T1.2	psutil agent, process whitelist

T2.4	Dashboard UI Components	Frontend	14 days	Track B	T1.3	System vitals, alert timeline, metrics viz
T2.5	API-Frontend Integration	Both	6 days	Both	T2.1, T2.4	API client, real-time data binding

Milestones:

- All three ML models operational with test data
- Dashboard displays real backend data
- End-to-end anomaly detection functional

Phase 3: Intelligence & User Experience (Week 5: Mar 1-7)

Objective: Add AI assistant, authentication, alerting, and security

Task	Name	Track	Duration	Team	Dependencies	Deliverables
T3.1	LLM Conversational Assistant	Backend	7 days	Track A	T2.1, T2.2, T2.3	Flask aggregator, LLM integration
T3.2	Chat Interface & Remediation UI	Frontend	7 days	Track B	T2.4, T3.1	AI chat component, remediation display
T3.3	Multi-Channel Alerting	Backend	7 days	Track A	T2.1, T2.2, T2.3	Telegram, Email, SMS integration
T3.4	Authentication & Authorization	Security	7 days	Both	T1.2, T1.3	JWT tokens, role-based access
T3.5	Security Hardening	Security	7 days	Track A	T1.1	Network isolation, secret management

Milestones:

- AI assistant provides contextual threat explanations
- Multi-channel alerts operational
- Secure admin access implemented

- Backend isolated within Docker network

Phase 4: Integration, Testing & Demo (Week 6: Mar 8-15)

Objective: End-to-end testing, attack simulation, demo preparation

Task	Name	Track	Duration	Team	Dependencies	Deliverables
T4.1	End-to-End Testing	QA	6 days	Full Team	T3.1-T3.5	Test scenarios, bug fixes
T4.2	Attack Simulation & PoC	Security	4 days	Track A	T2.3	Reverse shell test, detection validation
T4.3	Demo Environment & Video	Demo	6 days	Track B	T4.1	Demo script, video recording
T4.4	Documentation & Guide	Docs	4 days	Full Team	T4.1	Installation guide, architecture diagrams

Milestones:

- Detection-to-alert latency under 5 seconds
- Professional demo video completed
- Complete installation documentation
- Zero critical bugs

Development Timeline & Weekly Routine

The project follows a **6-Week Agile Sprint** structure. Instead of a rigid daily timetable, the team operates on a cyclical weekly routine that balances deep work with necessary synchronization points.

Sunday: Sprint Planning & Strategy

- **Full Team Virtual Meeting (1 Hour):** The week begins with a high-level sync to review the backlog and define the specific goals for the upcoming cycle.
- **Role Assignment:** Tasks are distributed based on the current priorities. Track A (Backend) defines the necessary API contracts and data models, while Track B (Frontend) confirms that the proposed data structures meet the UI visualization requirements.

Monday & Tuesday: Parallel Execution

- **Mode of Work:** Asynchronous development.
- **Track A Focus (Backend):** Deep technical implementation, including infrastructure setup, database schema management, and ML model training.
- **Track B Focus (Frontend):** UI/UX development, creating layout scaffolding, and building visual components using mock data until the backend is ready.

Wednesday: Mid-Week Checkpoints

- **Track-Specific Virtual Meetings (30 Minutes):** Each track holds a short, focused call to address specific technical blockers.
- **Technical Review:** Track A reviews code for endpoint logic and model accuracy. Track B reviews the interface responsiveness and theme consistency.

Thursday & Friday: Deep Work & Integration

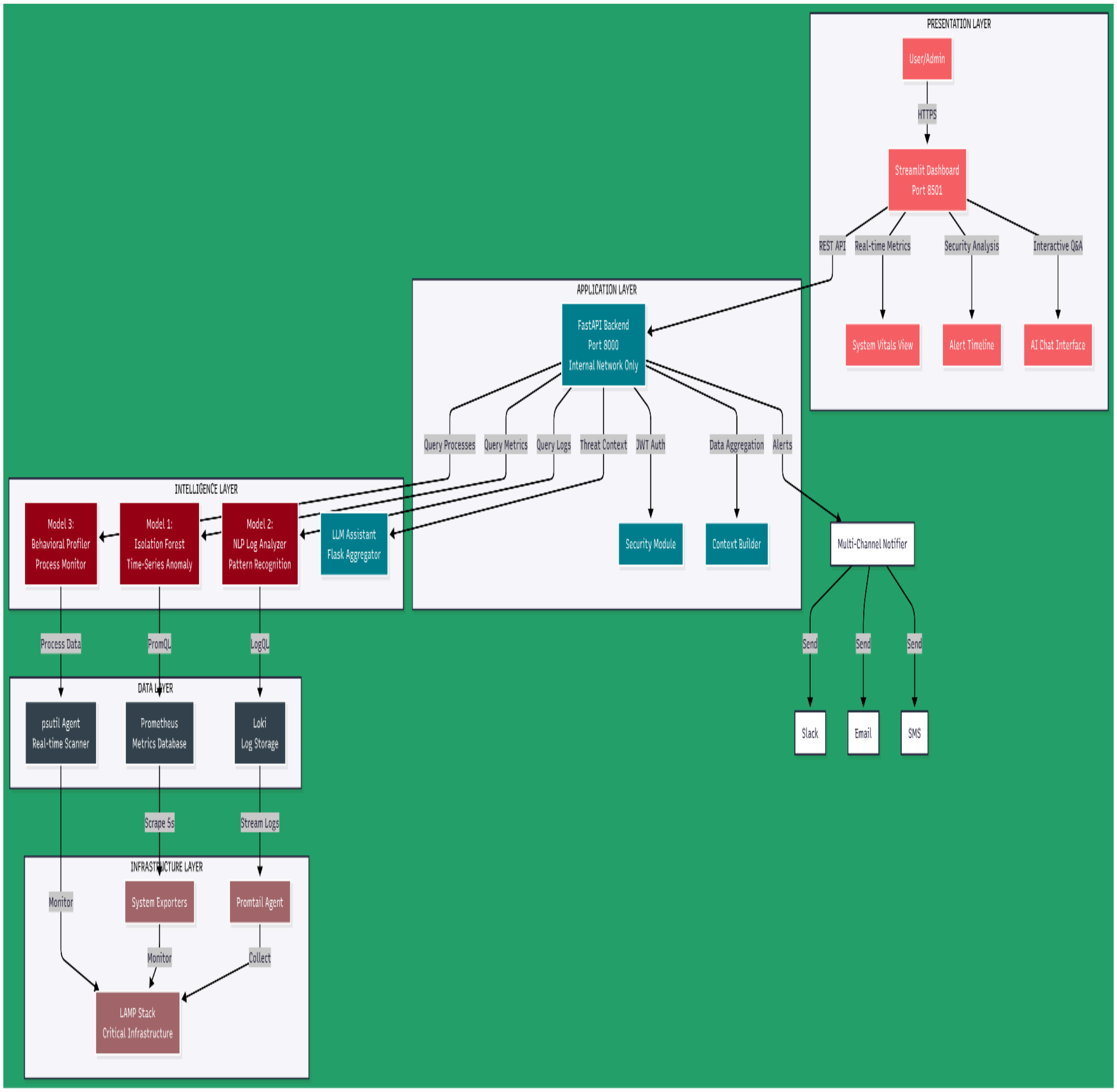
- **Heavy Coding:** Teams focus on complex logic implementation and connecting the separate components.
- **Integration (Friday):** Code branches are merged into the development environment. Track A focuses on testing Docker container stability, while Track B performs end-to-end flow testing to ensure the dashboard correctly displays real-time data from the API.

Saturday: Review & Deliverables

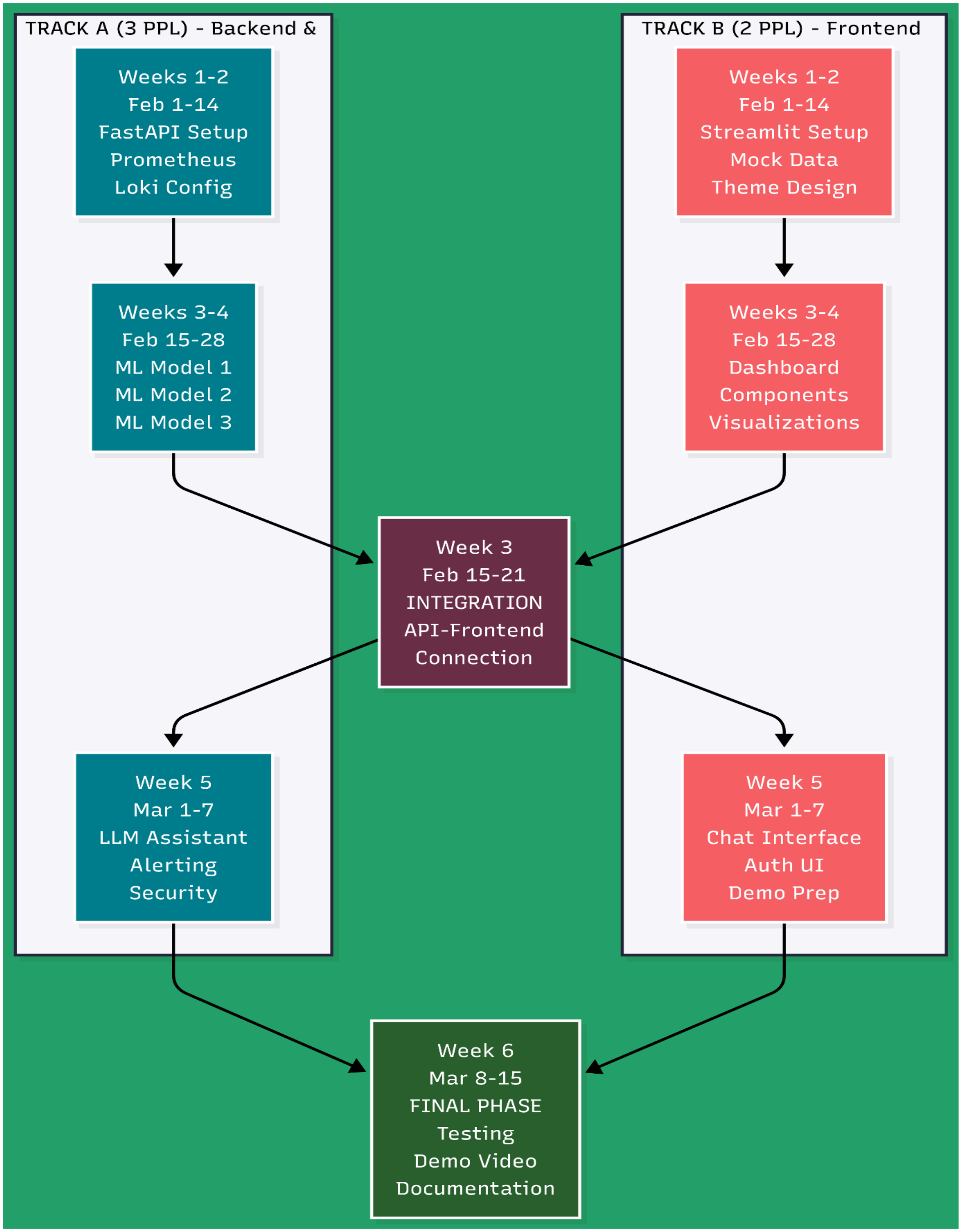
- **Full Team Physical Meeting (3 Hours):** The week concludes with a comprehensive review of the output.
- **System Demo:** The team walks through the completed features. Track A verifies system security and latency, while Track B verifies the user journey and aesthetics.
- **Next Steps:** Any incomplete items are moved to the backlog for the following Sunday's planning session.

5. Architecture Diagrams

System Architecture Flowchart



Parallel Development Workflow


TRACK A (3 PPL) - Backend &
Weeks 1-2 Feb 1-14 FastAPI Setup Prometheus Loki Config
Weeks 3-4 Feb 15-28 ML Model 1 ML Model 2 ML Model 3
Week 5 Mar 1-7 LLM Assistant Alerting Security
TRACK B (2 PPL) - Frontend
Weeks 1-2 Feb 1-14 Streamlit Setup Mock Data Theme Design
Weeks 3-4 Feb 15-28 Dashboard Components Visualizations
Week 5 Mar 1-7 Chat Interface Auth UI Demo Prep
Week 3 Feb 15-21 INTEGRATION API-Frontend Connection
Week 6 Mar 8-15 FINAL PHASE Testing Demo Video Documentation

Quick Reference

Installation Commands

# Clone repository

git clone https://github.com/CyberSentinel/sentinai-ops.git

cd sentinai-ops

# Configure environment

cp .env.example .env

# Launch stack

docker-compose up -d

# Access dashboard

open http://localhost:8501

Key Endpoints

- Dashboard: http://localhost:8501
- API Documentation: http://localhost:8000/docs
- Prometheus: http://localhost:9090
- Loki: http://localhost:3100

Team Information

Team CyberSentinel

- Shammah Sitati (Project Lead) - JKUAT
- Ben Tito - University of Nairobi
- Emmanuel Nyabicha - Kenyatta University
- Felix Robinson - Kenyatta University
- Godfrey Ayuka - Kenyatta University

Hackathon: AI for National Prosperity 2026