High Level Design Document

Introduction

This High Level Design (HLD) document outlines the architecture and core components for **SafeServe - AI Food Safety Inspector**. SafeServe is a web and IoT solution leveraging generative AI to analyze sensor data from food storage and preparation environments, providing safety compliance reports, actionable alerts, and exportable documentation for hospitality businesses and regulators.

---

1. System Architecture Overview

Architecture Description:
SafeServe consists of IoT sensor devices, a secure backend for data ingestion and processing, an AI analysis engine, a web-based user interface, and a reporting/alerting subsystem. All components interact via secure APIs and data channels.

Component	Role
IoT Sensor Network	Collects real-time environmental data (temperature, humidity, etc.)
Data Ingestion API	Receives and validates sensor data
Data Store (MongoDB)	Stores sensor data, user info, reports, and audit logs
AI Analysis Engine	Processes data, generates compliance reports, and detects anomalies
Alerting Module	Sends real-time alerts to users on safety issues
Web Application	User interface for monitoring, management, and report export
User Management	Handles authentication, authorization, and role-based access

---

2. Component Interactions

Sequence Step	Interaction Description
1	IoT sensors send data to Data Ingestion API over secure channel
2	Data Ingestion API validates and stores data in MongoDB
3	AI Analysis Engine retrieves new data, analyzes for compliance and anomalies
4	Alerting Module notifies users if issues are detected
5	Web Application fetches data, reports, and alerts for user display and export
6	User Management secures all user actions and data access

---

3. Data Flow Overview

Source	Destination	Data Type	Purpose
IoT Sensors	Data Ingestion API	Environmental metrics	Real-time monitoring
Ingestion API	MongoDB	Validated sensor data	Persistent storage
MongoDB	AI Engine	Sensor data	Compliance analysis
AI Engine	MongoDB	Reports, alerts	Store results and notifications
MongoDB	Web App	Reports, alerts, data	User access and export
Web App	User Management	Auth credentials	Secure access and role enforcement

---

4. Technology Stack

Layer/Component	Technology/Frameworks
IoT Devices	Python, MQTT/HTTPS
Backend/API	Python (FastAPI/Flask)
AI Analysis Engine	Python (Generative AI, ML libraries)
Database	MongoDB
Web Application	React.js (or similar), REST API
Authentication	JWT, OAuth2
Deployment	Docker, Kubernetes (optional)
Monitoring/Logging	Prometheus, ELK Stack (optional)

---

5. Scalability & Reliability

- **Scalability:**
  - Modular microservices enable independent scaling of ingestion, AI, and web layers.
  - MongoDB supports horizontal scaling for large data volumes.
  - Stateless APIs and containerization (Docker/Kubernetes) facilitate load balancing.
- **Reliability & Security:**
  - Secure communication (TLS) between IoT devices and backend.
  - Role-based access control and encrypted user data.
  - Automated alerting and audit logging for compliance.
  - Redundant deployment and health monitoring for high availability.

End of Document