MICHIGAN STATE UNIVERSITY

Project Plan Presentation

ML/AI Pipeline for Condition-Based Maintenance 2.0

The Capstone Experience

Team Magna AI4CBM

Mahin Bhat
Daniel Mar
Minh Nguyen
Himanshi Pakale
Raj Patel
Sanjay Senthil Marison

Department of Computer Science and Engineering
Michigan State University

Spring 2026



From Students…
…to Professionals

Project Sponsor Overview

- Magna is one of the world's largest automotive suppliers.
- It has 158,000 employees spanning across 342 manufacturing operations and 91 centers in 27 countries
- They develop technologies that make vehicles safer and cleaner such as ADAS, RIM bumpers, blind-spot detection and lane departure warning systems.
- Magna has ranked consistently in the Fortune Global 500 list for 20 years in a row since 2001.



Project Functional Specifications

- Unplanned machinery downtime is a major source of cost and disruption in manufacturing environments
- This project develops Condition-Based Maintenance (CBM) system that operates on live production equipment
- The system ingests real-time vibration, temperature, and audio sensor data from active machinery
- Machine learning models analyze the data to assess machine health and detect early signs of failure
- Results are presented through clear health indicators, confidence scores, and automated alerts
- The goal is to enable proactive maintenance, reduce unplanned downtime, and improve operational reliability

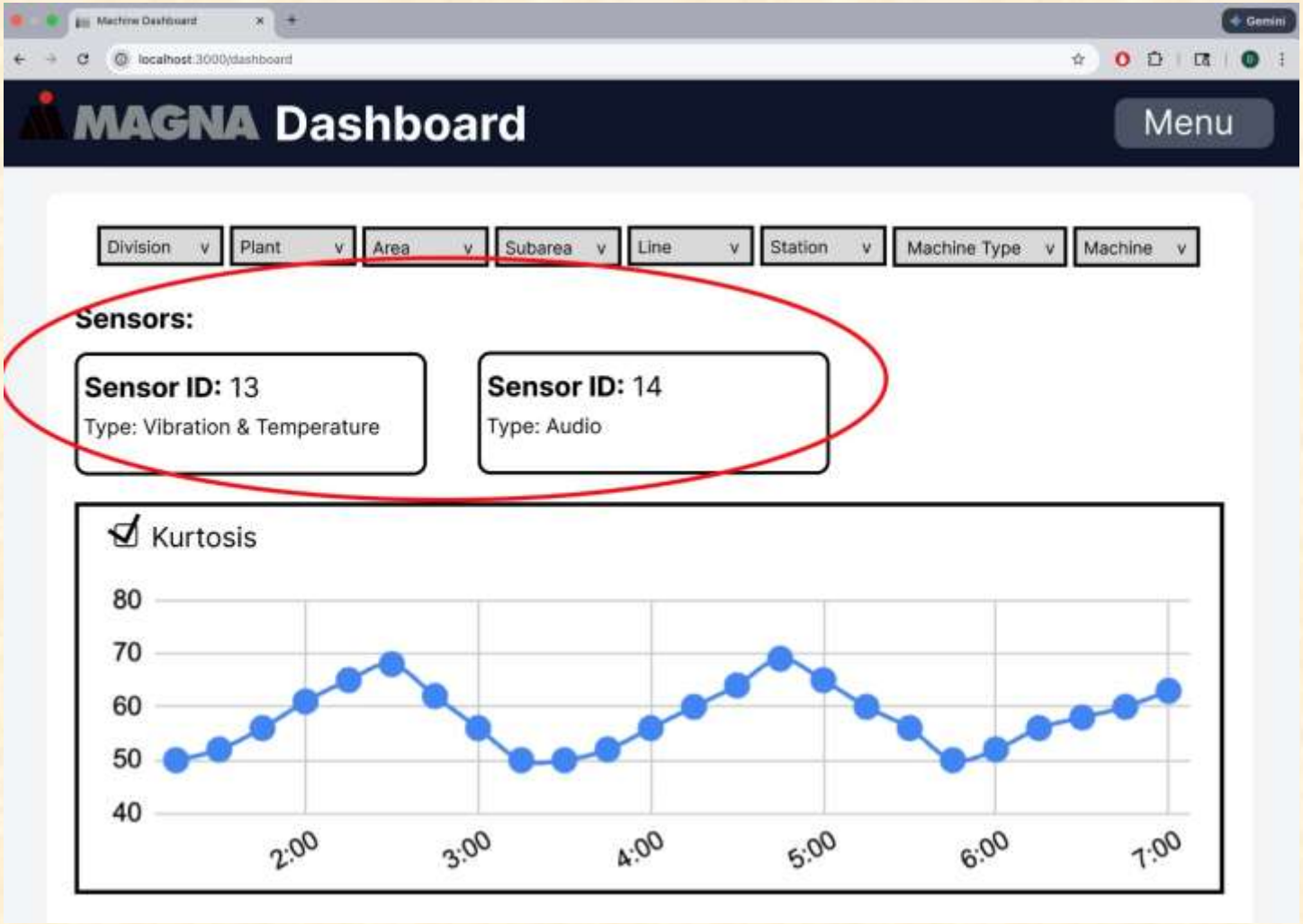


Project Design Specifications

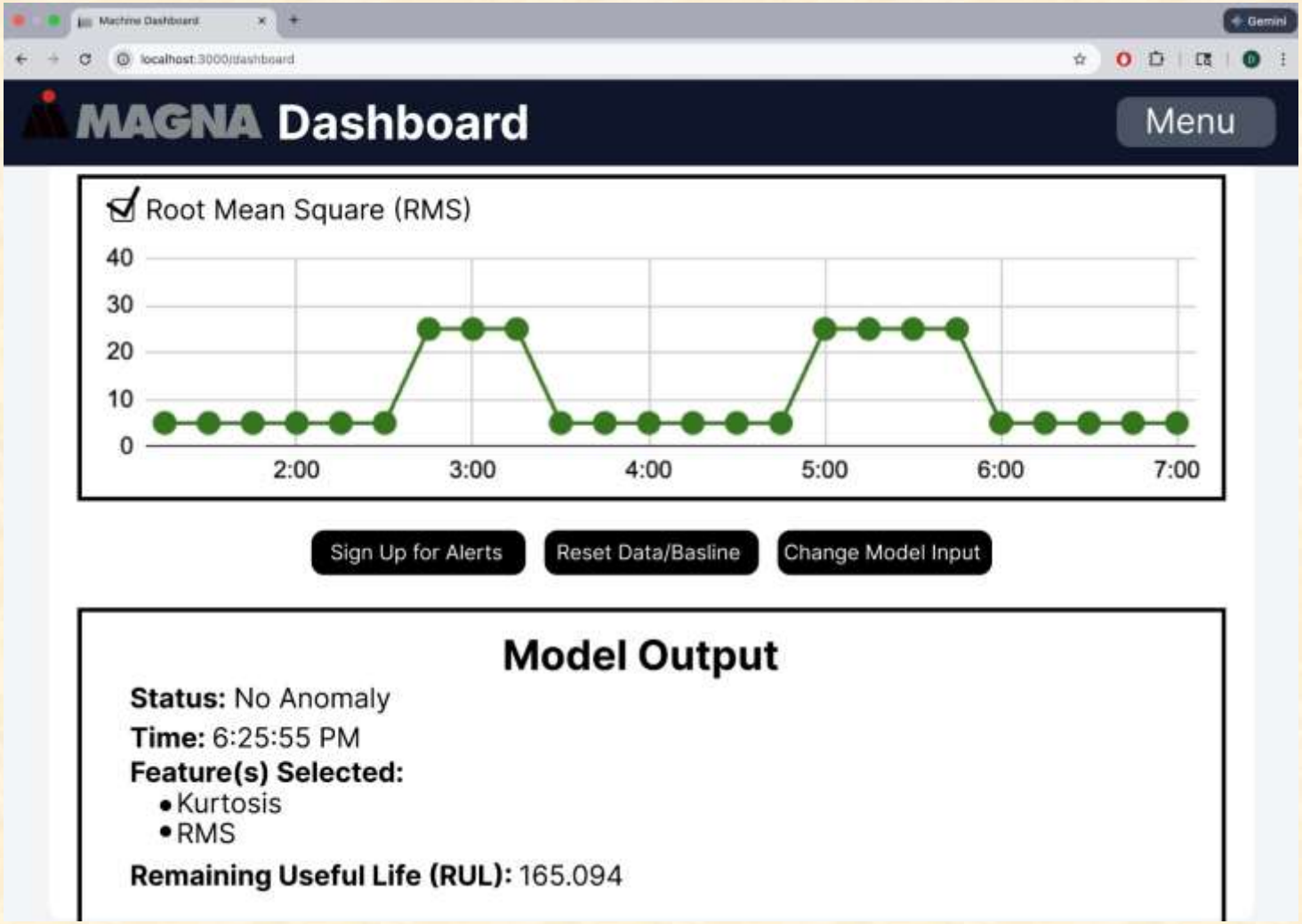
- Estimate the Remaining Estimated Life (REL) for every machine using a Condition-Based Model
- Display machine data in a comprehensible and intelligible way by creating graphs for both real-time and historical data
- Notify engineers when anomalous activity or machine failure has occurred by sending out emails



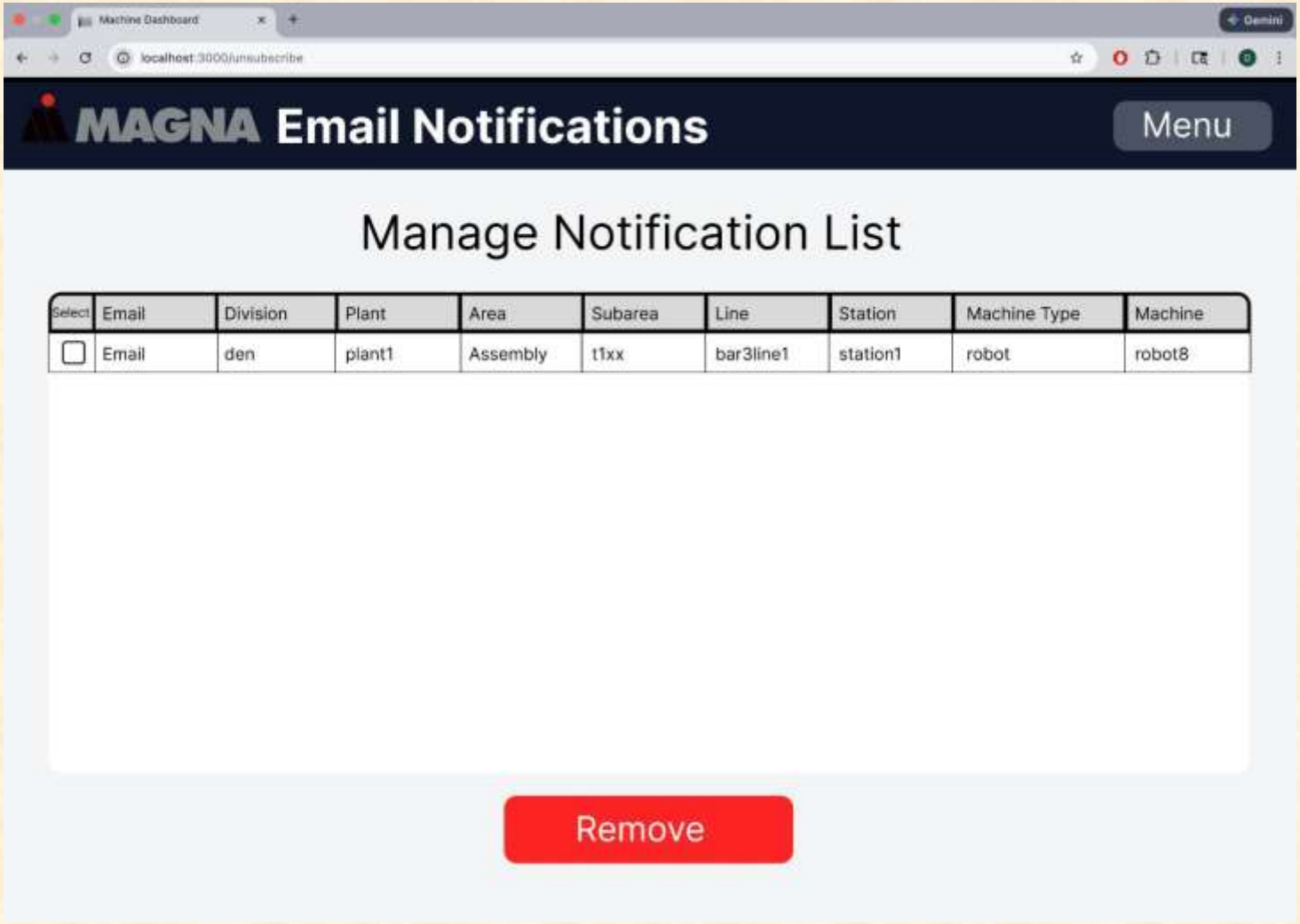
Screen Mockup: Dashboard



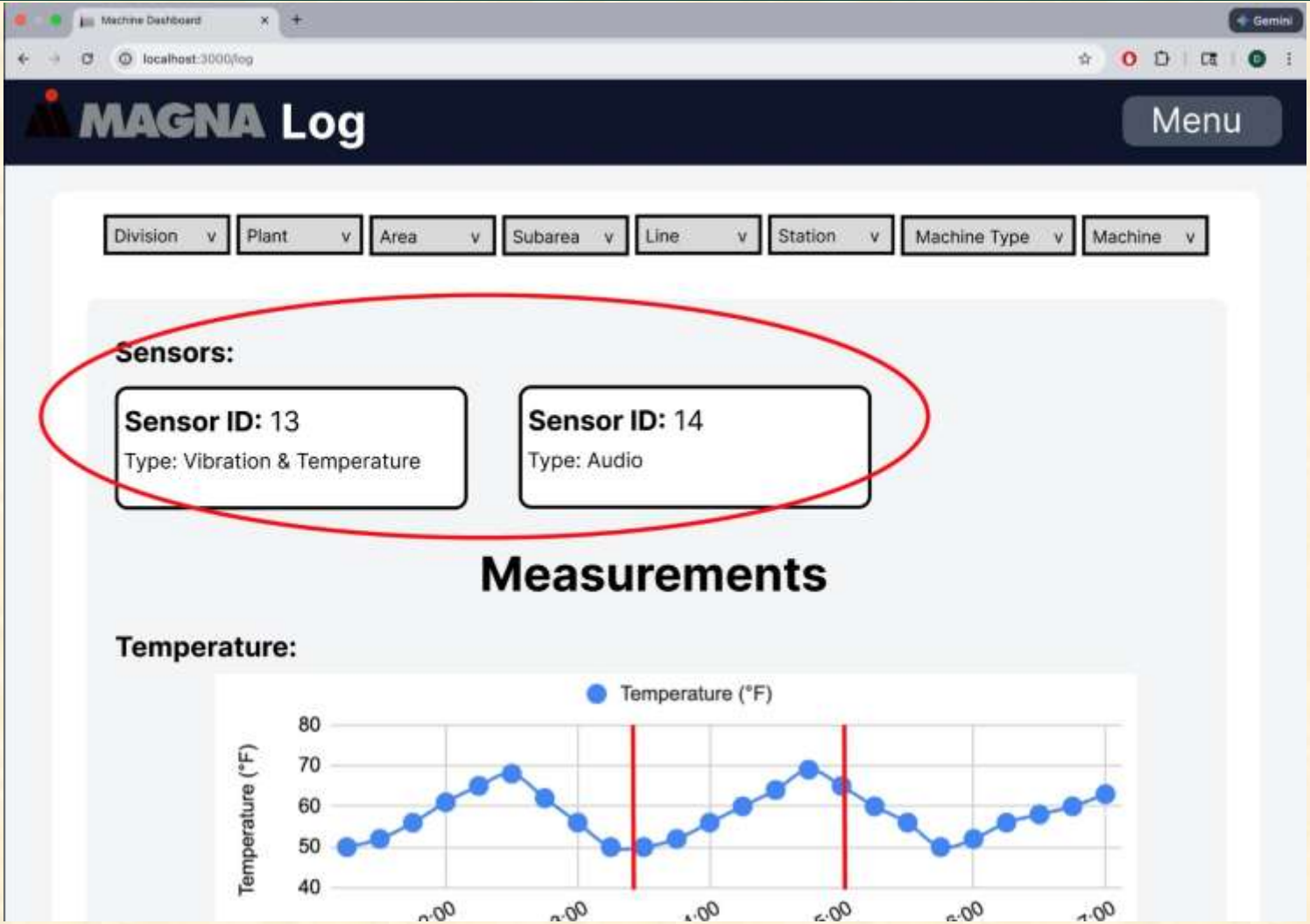
Screen Mockup: Dashboard



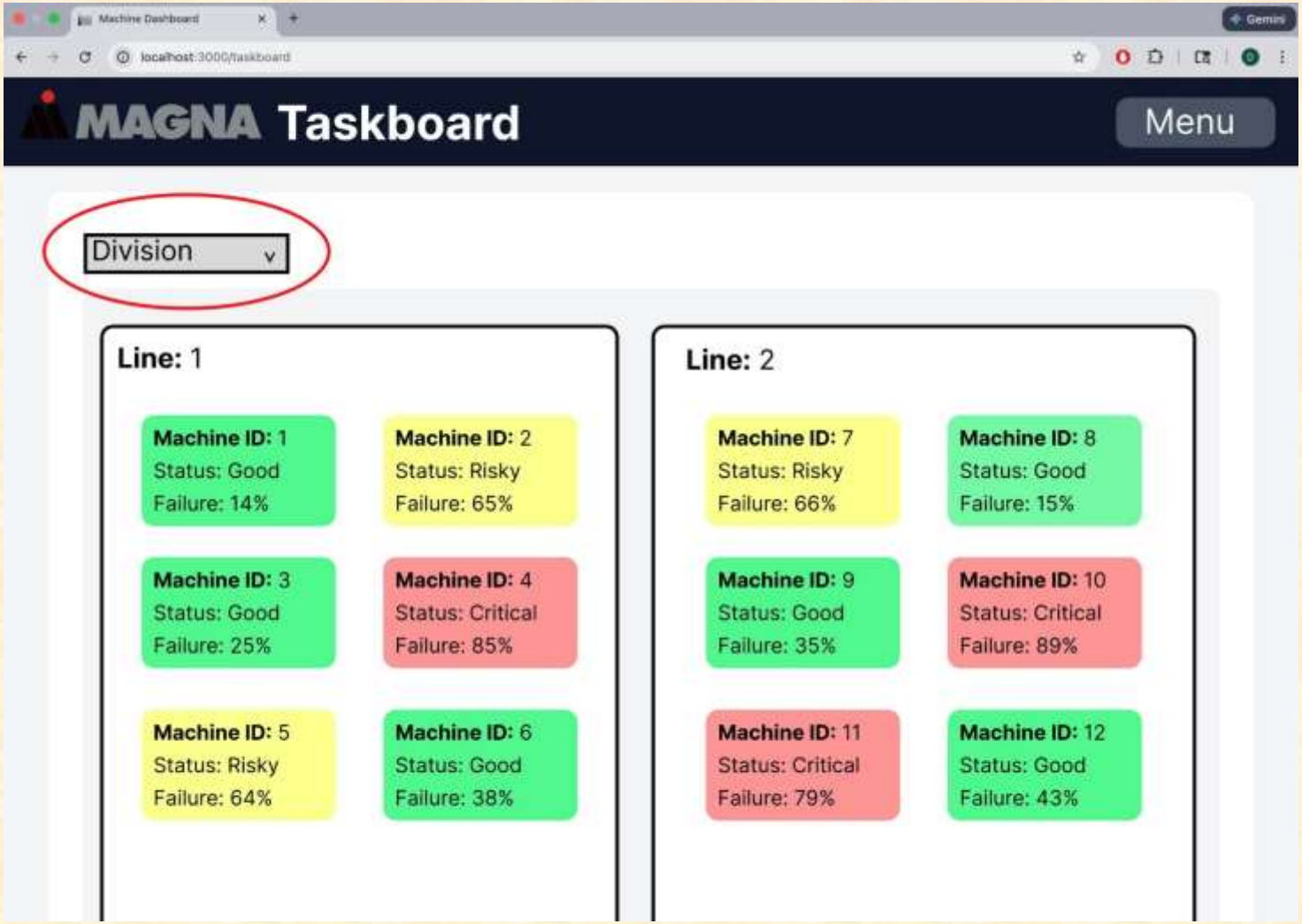
Screen Mockup: Email notifications



Screen Mockup: Log



Screen Mockup: Task board



Project Technical Specifications

- **Data Collection and Messaging**
  - Physical machine sensors (vibration, temperature)
  - Audio sensor (microphone)
  - MQTT broker (HiveMQ)
- **ROS2 Processing Pipeline**
  - ROS2 Humble framework
  - MQTT and ROS2 bridge node
  - Preprocessing node (windowing, scaling, feature selection)
  - Audio preprocessing node (feature extraction, windowing)
  - Model deployment node
  - Control node (runtime configuration)
- **Machine Learning**
  - Siamese LSTM for anomaly detection and RUL estimation
  - Chronos/Chronos 2 models for audio-based anomaly detection and time series forecasting
  - Multimodal feature fusion (audio and scalar data from sensors)
  - Pretrained pytorch models

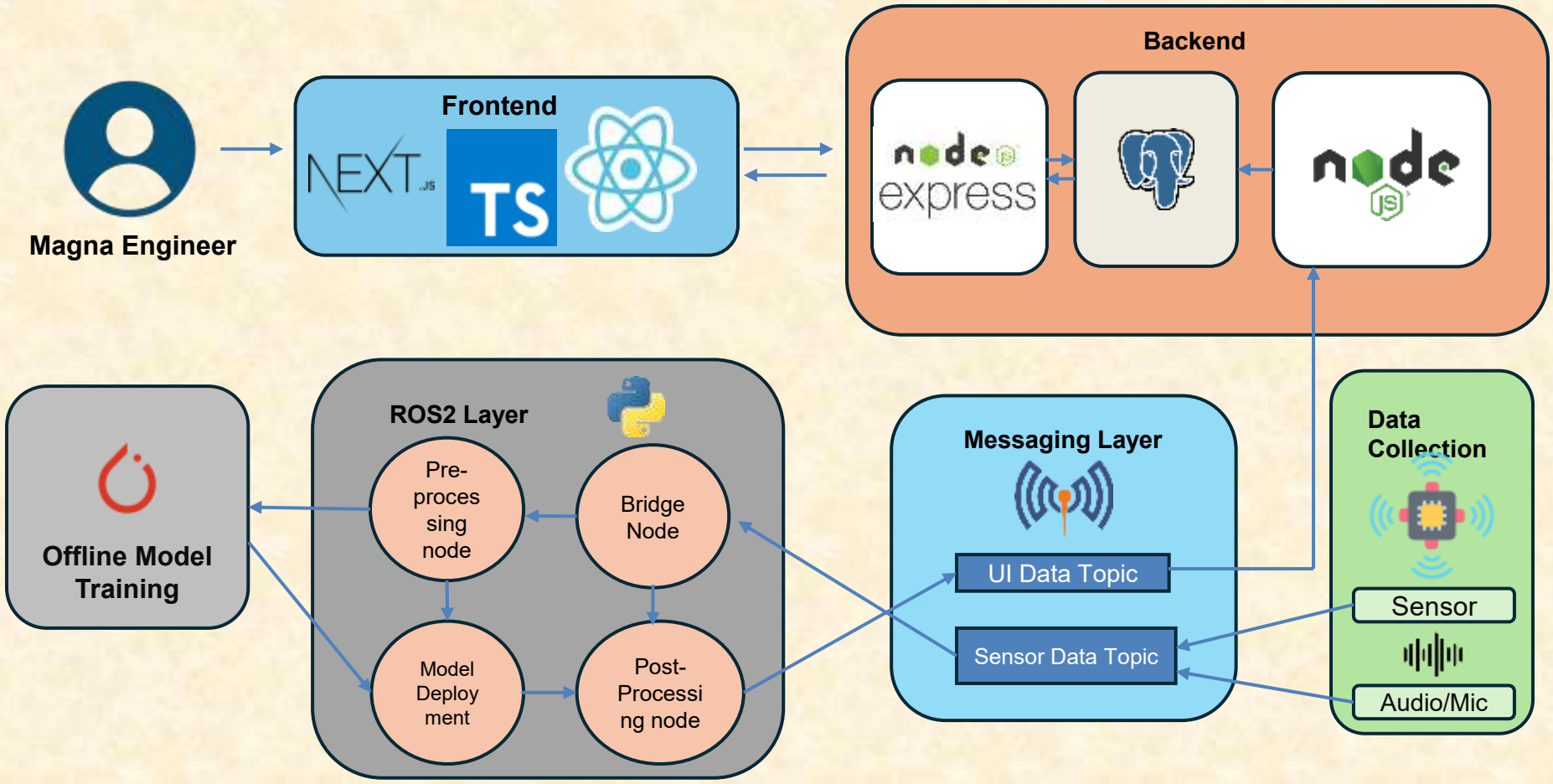


Project Technical Specifications

- **Backend Infrastructure**
  - AWS EC2 instance
  - Node.js services
    - MQTT subscriber
    - REST API
  - PostgreSQL
  - Schemas for raw sensor data, audio features and model outputs
- **Frontend and notifications**
  - Next.js web application (React, Typescript)
  - Dashboard, task board, and log views
  - Audio anomaly indicators powered by Chronos/Chronos 2
  - Vibration and fused anomaly health indicators
  - Email notifications
- **Deployment and Tooling**
  - Ubuntu 22.04, ROS2 Humble
  - Python 3.10, Node.js, npm, PM2
  - Colcon build system for ROS2
  - Audio processing libraries (e.g. Torchaudio)



Project System Architecture



Project System Components

- **Hardware Platforms**
  - Sensors: Vibration, Temperature, Audio
  - Compute: Ubuntu 22.04 server, NVIDIA Jetson Nano
  - Cloud: AWS EC2 instance
  - Networking: Network interface for MQTT communication (Wifi)
- **Software Platforms / Technologies**
  - OS: Ubuntu 22.04, MacOS
  - Robotics: ROS2 Humble, ROS2 launch system, colcon build system
  - Messaging and communication: MQTT protocol, HiveMQ broker
  - ML: PyTorch, Siamese LSTM model, Chronos/Chronos 2 model
  - Backend: Node.js, Express.js, MQTT, PM2
  - Database: PostgreSQL
  - Frontend: Next.js, React, Typescript, NextAuth, Resend
  - Simulation and Testing: IMS Vibration
  - Development and Deployment: Python 3.10, npm, Node, Git



Project Risks

- **Risk 1: Implementing Suitable Audio Sensor (High)**
  - Description: Connecting a suitable audio sensor for the scope of this project will come with challenges such as mounting, usability, and data streaming
  - Mitigation: Prioritize using existing machine audio datasets before final selection. This will build a simulation for a smoother and effortless transition to real time sensor data flow.
- **Risk 2: NVIDIA Jetson Nano Configuration (Medium)**
  - Description: The Jetson Nano is a powerful embedded computer we will use to connect to the sensors. However, the team has no experience with this hardware nor experience with Jetson connectivity.
  - Mitigation: Implement a variety of steps such as identifying OS setup, available drivers, and ports. Then, validate broker configuration to servers and endpoints. Finally, use data simulation before processing with real time sensors.
- **Risk 3: Underperforming Condition-Based Monitoring Models (Medium)**
  - Description: Current models can end pre-maturely, leading to false alarms or missed faults.
  - Mitigation: Run benchmarks for models such as ZOO & Chronos. Evaluate them and select the best based on accuracy.
- **Risk 4: Network and Connectivity Security (Low)**
  - Description: Unstable connectivity can lead to data drops or delayed messages which can cause false alarms or unsuitable predictions.
  - Mitigation: Connection status indicators will be added to UI to display complications and problems. This will help with clear communication regarding network issues and display clear warnings.



Questions?

