BATCH-7 DESIGN THINKING ACTIVITY

THIRUMURUGAN V CB.EN.U4EEE22151

SHREHAARAAN A CB.EN.U4ELC22050

SUREN KARTHIK B J CB.EN.U4ELC22051

SUBHIKSHA P CB.EN.U4ELC22054

SAYA SAI KOWSHALYA CB.EN.U4EEE22041

Problem Description:
Conveyors in food packaging experience mechanical wear. IR and MPU6500 sensors detect motion and vibration data, which is sent to the cloud. ML models learn normal behavior and predict faults. Data analytics reveals usage patterns, wear trends, and failure precursors. Maintenance dashboards and KPIs help schedule servicing proactively, minimizing downtime and enhancing reliability.

STAGE 1: Brain Storming GD

Challenges:

(sensors issues):

Noise detection from external machines

Opportunities:

Using sensor fusion and filtering to improve data accuracy and fault detection.

Ideas:

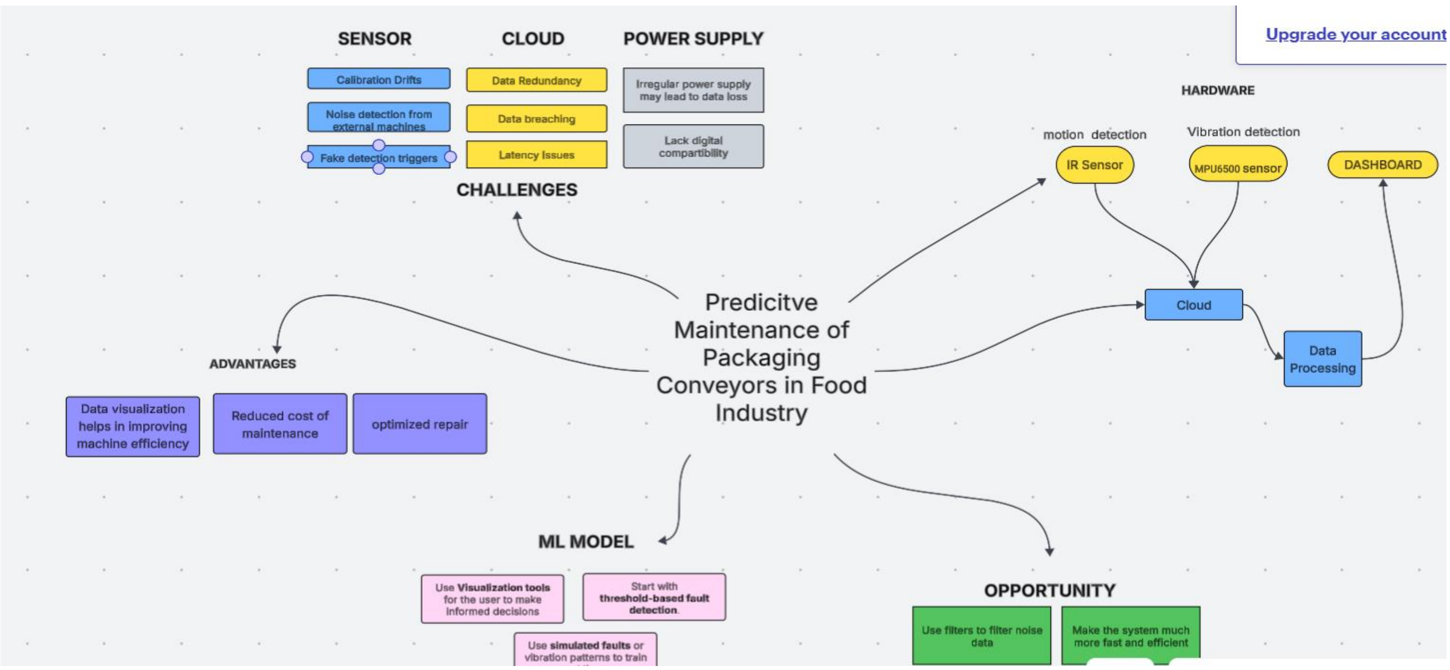
May be extended to implement edge computing for real-time sensor fusion and noise reduction, allowing faster and more accurate fault detection directly at the machine level.

Stage 2: Mind-map & Abstract

In the food packaging industry, conveyor belts are essential for maintaining smooth and continuous operations. However, unexpected breakdowns result in costly downtime, resource wastage, and production delays. Traditional maintenance method whether scheduled or reactive fail to prevent such issues effectively.

This project proposes a predictive maintenance system for packaging conveyors in the food industry using IR and MPU6500 sensors. These sensors capture motion and vibration data, which is transmitted to the cloud and analyzed using machine learning models to detect anomalies and predict mechanical failures in advance. A real-time dashboard provides visibility into machine health, usage trends, and KPIs.

Unlike traditional maintenance methods or existing systems that rely on fixed schedules or expensive industrial-grade sensors, this solution is low-cost, scalable, and data-driven. It enables early fault detection, minimizes unplanned downtime, and supports remote monitoring, aligning with Industry 4.0 practices. This makes it more accessible and efficient than many existing market alternatives.



Stage 3: Questionnaire

FOR OPERATORS

What are the most common issues you face with the conveyor belts?

How do unexpected breakdowns affect the productivity of the factory?

How do you currently detect or report faults?

FOR MAINTENANCE PERSONNEL

What types of faults or breakdowns occur most frequently?

How often do u repair conveyors?

How do track the health of conveyor?

FOR PRODUCTION MANAGERS

How does conveyor downtime impact production targets?

What information would help you make better decisions?

FOR TECHNICAL STAFF

What challenges do you face with current monitoring or data systems?

Are there any integration or connectivity issues with existing equipment?

STAGE 4: IDEA LAYOUT

Converyors in food packing industires face mechanical wear problem which causes huge financial loss in industries.

Motion and Vibration data collected from sensors are sent to cloud.

ML Algorithmns such as One class SVM,Random forest,Convloutional Neural networks are used to detect faults and predict future faults.

A Dashboard is designed to present the scheduling ,servicing proactively,minimizing downtime and enhancing reliablity.

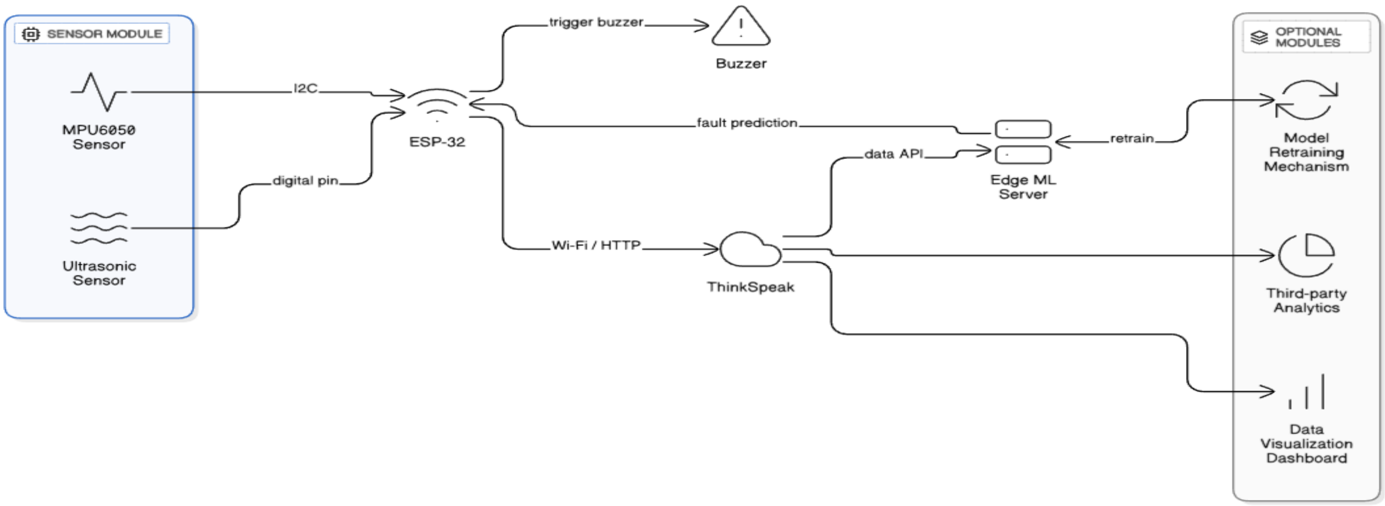
This project lays the foundation for smart, self-monitoring factories. Future advancements in AI, IoT, and robotics could transform it into a fully autonomous maintenance ecosystem.

This project can be enhanced by adding the features such as Multi-Sensor fusion,Edge Aland Energy efficiency optimization.

Stage 5: Ideate

- Identify critical conveyor belt components for monitoring
- Ensure Sensor placement for optimal data collection.
- Check compatibility with existing conveyor and IT infrastructure
- Establish Baseline performance data for conveyors
- Define KPI such as downtime and maintenance cost etc
- Integrate PMS with existing MMS
- Schedule regular calibration and inspection of sensors
- Stock critical spare parts to minimize repair delays
- Ensure relevance to workplace safety standards and temperature

Stage 6: Modular Architecture

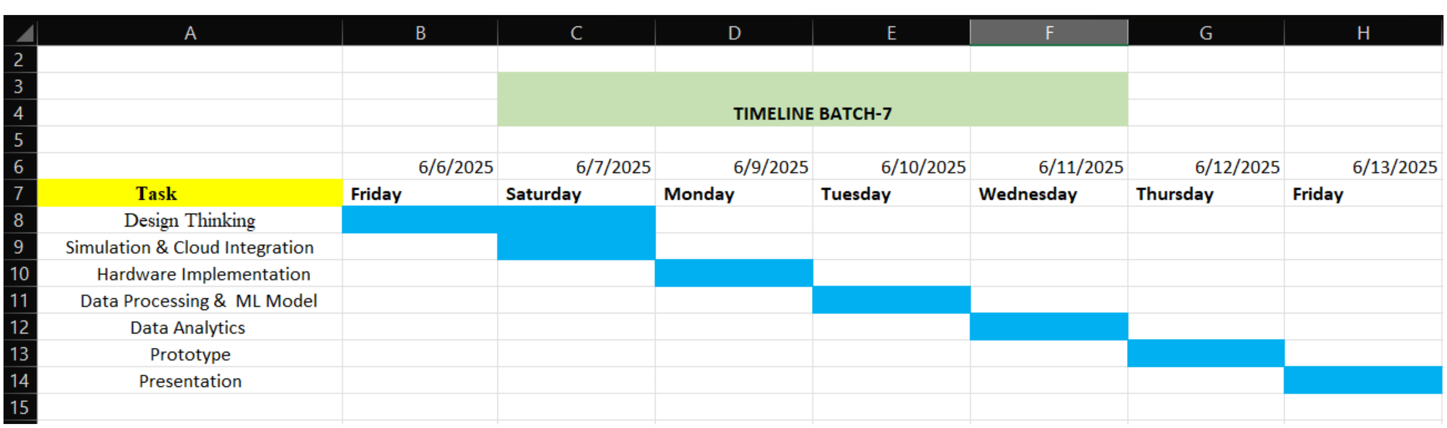


Stage 7: Requirement Specification

S.no	Component/Tool	Qty	Type	Cost (Rs.)
1	Ultrasonic Sensor	1	hardware	58
2	MPU6500	1	hardware	113
3	ESP32	1	hardware	500
4	FireBase	-	software	NA
5	Power BI	-	software	NA

Stage 8: Planning

Timeline Gantt chart :



Risk analysis:

FMEA REPORT

Process/Component	Potential Failure Mode	Potential Effect(s) of Failure	Severity (S)	Potential Cause(s)	Occurrence (O)	Current Controls	Detection (D)	RPN	Recommended Actions
Ultrasonic Sensor	No distance readings	Object detection failure, collision risk	8	Sensor damage, loose wiring	5	Checksum validation, periodic calibration	4	160	Add redundancy or regular self-tests
MPU6050	Incorrect vibration data	Missed early warning signs of failure	9	EMI, loose connection	4	Smoothing/filtering, deviation checks	5	180	Use sensor fusion or filters
ESP32	Hangs or resets unexpectedly	No data sent to ThingSpeak, blind	8	Power fluctuation, firmware bugs	4	Watchdog timer, brown-out detection	3	96	Improve firmware reliability, use reset logic
Wi-Fi Module (ESP32)	Connectivity loss	Gaps in monitoring data	6	Network outage, signal interference	6	Reconnect logic, offline buffer	5	180	Add data caching and retries
ThingSpeak	Server not responding	No visualization or alerts	7	API/server error, rate limits	3	Retry mechanisms, API limit handling	3	63	Add fallback or alert on server failure
Power Supply	Power dropout	Total system failure	10	Loose cable, battery drain	3	Battery backup or capacitor buffering	2	60	Add UPS or supercapacitor
Code (Firmware)	Logic error in threshold checking	Wrong alert or missed failure	9	Coding mistake, wrong calibration	5	Simulated test cases, hardware-in-loop validation	4	180	Extensive testing and modular code reviews
Sensor Mounting	Vibration loosens sensor	Inaccurate readings	7	Weak adhesives, mechanical stress	5	Physical inspection, sensor position logging	6	210	Improve mount design, use vibration-proof clamps

Stage 9 : Redesign

Conclusion from the stage 9 - risk analysis and Stage 5 -check list

- **Focus on the right components**
  We monitor motors, rollers, and belt tension areas — the parts most likely to fail.
- **Place sensors where it matters**
  We install sensors in high-vibration and high-stress zones to collect useful data.
- **Ensure system compatibility**
  ESP32 and sensors work well with the existing conveyor setup and Wi-Fi network.
- **Collect baseline data**
  We record how the system performs under normal conditions to detect future changes easily.
- **Define key performance indicators (KPIs)**
  We track downtime, maintenance costs, alert frequency, and early warning times.
- **Integrate with the maintenance system**
  Alerts from the predictive system connect directly to our existing maintenance workflow.
- **Schedule regular checks**
  We calibrate and inspect sensors regularly to keep the system accurate.
- **Stock critical spare parts**
  We keep essential components ready to avoid delays in case of failure.
- **Follow safety and compliance standards**
  The setup meets safety norms and handles real factory conditions like heat and dust.

Stage 10: Execution Framework

Module	Purpose	Tools & Technologies	Key Metrics
Edge Sensing & Data Push	Read distance (ultrasonic) & vibration (MPU6050); send data to cloud	ESP32 (Arduino), Ultrasonic sensor, MPU6050, Wi-FiMQTT/HTTPS → ThingSpeak	Data latency (ms), Sample accuracy
Cloud Ingestion & Logging	Receive, store, and visualize sensor data	ThingSpeak channelsThingSpeak JSON API	Data uptime (%), Data loss (%)
Anomaly Detection (ML)	Detect deviation from “normal” vibration & distance patterns	Python (scikit-learn, pandas)CSV data → Cleaned → Trained ML model	Detection accuracy, FP/FN rate
Live Monitoring Dashboard	View vibration trends, distance logs, and flags in real-time	ThingSpeak Charts, MATLAB Visualizations	Refresh rate, Alerts per hour
Notification System	Notify operator if vibration/distance exceeds threshold	IFTTT / Email Alerts / Telegram Bot	Alert delay (ms), Operator response time
System Health Check	Check if sensors & board are working and communicating	Heartbeat packet system, ESP32 self-check routine	Uptime %, Last data received time

Stage 11: Micromodules

Micromodule Name	Description
1. Sensor Data Acquisition Module	Gathers readings from ultrasonic sensor and MPU6050 using ESP32. Handles basic filtering (e.g., Kalman filter).
2. Wi-Fi Sync & ThingSpeak Uplink	Connects to Wi-Fi, formats sensor data, and uploads via HTTP or MQTT to ThingSpeak. Includes failover retry.
3. Data Storage & Cleaning	Exported ThingSpeak CSV is cleaned for noise, gaps, or outliers. Prepares data for ML training.
4. Machine Learning Classifier	Learns “normal” patterns from historical data. Detects abnormal readings using trained model (e.g., isolation forest).
5. Alert Generation Module	Triggers alert if sensor data deviates from norm. Connects to IFTTT or custom script for SMS/Email alerts.
6. Dashboard & Trend Analysis	Visualizes live and historical data on ThingSpeak. Shows system status, trends, and anomalies.
7. System Self-Test & Recovery	Periodically checks if sensors are online. If stuck, restarts ESP32 or logs fault.

8. Maintenance Prediction Module	Predicts when a conveyor issue might happen based on trend slope and anomaly frequency.