COLLEGE CODE: 3108

COLLEGE NAME: Jeppiaar Engineering College

DEPARTMENT: Information Technology

STUDENT NM-ID:4A2D13C5661655A82114037B0E9C4890

ROLL NO:310823205040

DATE:12/05/2025

TECHNOLOGY-PROJECT NAME: Structural Health Monitoring

SUBMITTED BY,

A Daniel, Harish, Kamalesh.S, Jobel Creflo

Phase 5: Project Demonstration & Documentation

Title: Structural Health Monitoring

Abstract

The Structural Health Monitoring (SHM) project integrates AI, IoT, and machine learning techniques to monitor the health of infrastructure in real time. This system employs sensors to track parameters like strain and vibration, processes this data through AI models such as CatBoost and AVOA, and detects early signs of damage. This document outlines the project's demonstration, architecture, performance, and documentation.

Index

1. Project Demonstration

Overview:

The SHM system will be demonstrated showcasing its data capture from sensors, AI-powered anomaly detection, and real-time reporting.

Demonstration Details:

- System Walkthrough: Simulation of data reading from vibration and strain sensors.
- AI Diagnosis Accuracy: Demonstrates crack detection using CatBoost and AVOA.

Phase 5: Project Demonstration & Documentation

- IoT Integration: Real-time monitoring of structural metrics via simulated sensor feeds.
- Performance Metrics: Show system's low latency and high accuracy in identifying anomalies.
- Security & Privacy: Data protection via encryption and access controls.

Outcome:

A complete working demonstration of the SHM system handling real-time data with predictive analysis, proving its readiness for real-world application.

2. Project Documentation

Overview:

This section provides detailed architecture, code, user/admin guides, and testing results.

Documentation Sections:

- System Architecture: AI model pipeline and sensor data flow diagram.
- Code Documentation: Explanation of sensor simulation code and anomaly detection logic.
- User Guide: Instructions on how stakeholders can interpret SHM alerts.
- Admin Guide: Procedures for sensor calibration and system maintenance.
- Testing Reports: Metrics on latency, accuracy, and stress tests.

Outcome:

Complete documentation facilitates future scaling and maintenance of the SHM solution.

3. Feedback and Final Adjustments

Overview:

Phase 5: Project Demonstration & Documentation

Stakeholder feedback will be gathered during the live demonstration.

Steps:

- Feedback Collection: Surveys from mentors and observers.
- Refinement: Adjust thresholds, UI/UX improvements for monitoring dashboard.
- Final Testing: Conduct another round of live simulations.

Outcome:

Refined system with improved usability and performance accuracy.

4. Final Project Report Submission

Overview:

Summarizes progress from phases 1 to 5 and the core milestones.

Report Sections:

- Executive Summary: High-level goals and achievements of SHM.
- Phase Breakdown: Details of model selection, testing, and deployment.
- Challenges & Solutions: Data noise, processing load, and sensor calibration issues.
- Outcomes: A robust, real-time SHM system ready for deployment.

Outcome:

Formal documentation of the complete project lifecycle.

5. Project Handover and Future Works

Phase 5: Project Demonstration & Documentation

Overview:

Proposal for next steps and expansion.

Handover Details:

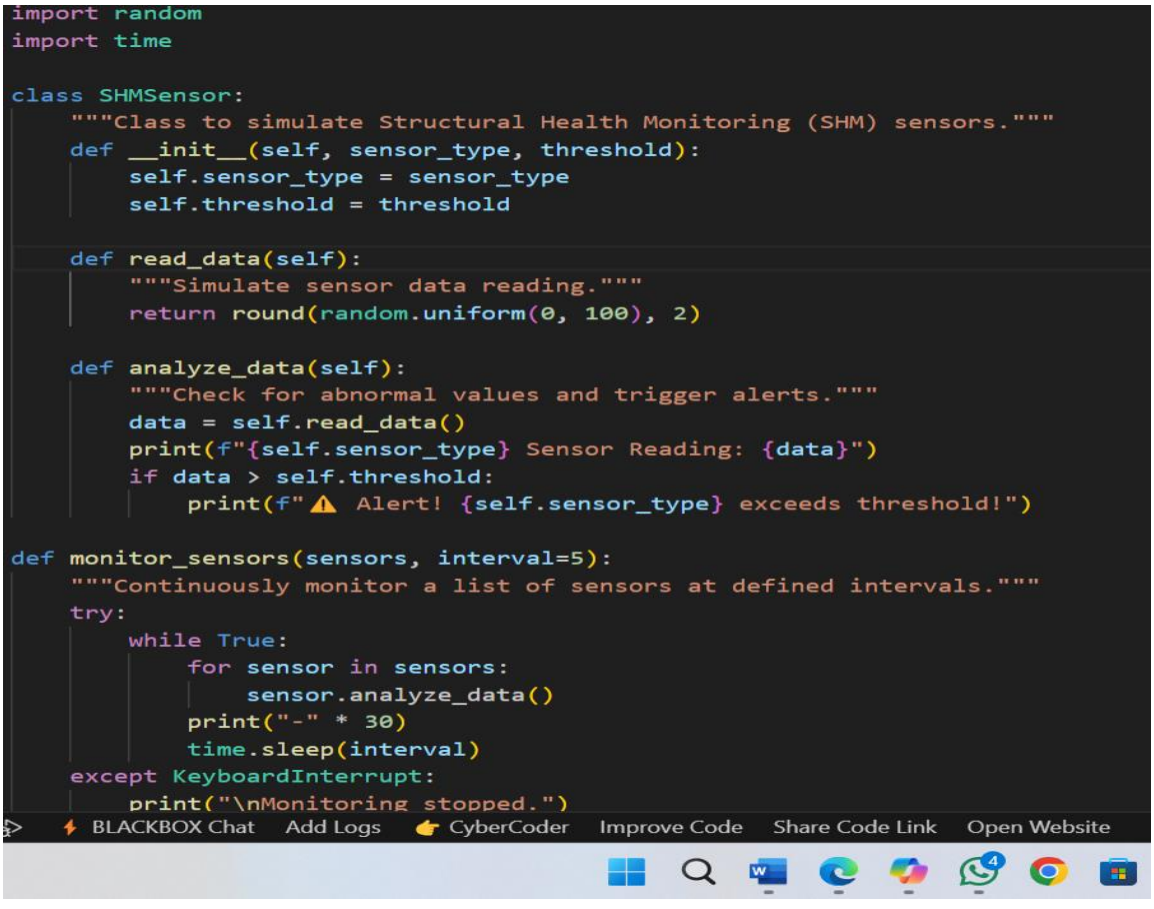
- Next Steps: Include live hardware deployment, integrate with civil infrastructure, enhance UI.
- Future Work: Expand to include corrosion detection, edge computing, and multilingual alerting.

Outcome:

SHM project prepared for handover and suitable for further development.

6. Screenshots

Python program:



Phase 5: Project Demonstration & Documentation

```python
class SHMSensor:
    def analyze_data(self):
        data = self.read_data()
        print(f"{self.sensor_type} Sensor Reading: {data}")
        if data > self.threshold:
            print(f"⚠️ Alert! {self.sensor_type} exceeds threshold!")

def monitor_sensors(sensors, interval=5):
    """Continuously monitor a list of sensors at defined intervals."""
    try:
        while True:
            for sensor in sensors:
                sensor.analyze_data()
            print("-" * 30)
            time.sleep(interval)
    except KeyboardInterrupt:
        print("\nMonitoring stopped.")

# Define sensors
sensors = [
    SHMSensor("Vibration", 70),
    SHMSensor("Strain", 50)
]

# Start monitoring
monitor_sensors(sensors)
```

Output: