COLLEGE CODE:3126

COLLEGE NAME:Thangavelu engineering College

DEPARTMENT:B.TECH-IT

STUDENT NM-ID: 330894bbac6dbaebe3d4c5bb5955b9c2

ROLL NO:312623205010

DATE:17/05/2025

TECHNOLOGY-PROJECT NAME:AI

SUBMITTED BY,

HARISH.P

S.ASWIN

S.SAMVEL

B.GUNAL

S.GOPALAKRISHNAN

AI-Enabled Structural Health Monitoring System

Title: AI-Enabled Structural Health Monitoring System

Abstract:

The AI-Enabled Structural Health Monitoring (SHM) project aims to modernize infrastructure safety by integrating artificial intelligence and IoT sensors. This system monitors the health of civil structures like bridges, buildings, and dams in real time, providing early warnings of damage or stress. The platform collects vibration, strain, and displacement data using embedded sensors and applies machine learning algorithms to detect anomalies. It ensures data security and scalability and is compatible with existing enterprise systems. This document covers the full project lifecycle including demonstration, technical documentation, performance reports, and future development suggestions.

Index:

1. Project Demonstration

Overview:

The SHM system will be demonstrated to stakeholders, showcasing its real-time structural monitoring, AI analysis, and predictive maintenance alerts.

Demonstration Details:

- System Walkthrough: Live demonstration showing sensor data input and AI-based structural condition outputs.
- AI Analysis: Detection of stress points and potential failures using trained machine learning models.
- IoT Integration: Real-time data from sensors measuring strain, displacement, and vibrations.
- Performance Metrics: System response time, accuracy in anomaly detection, and scalability.
- Security: Data encryption and secure transmission protocols to ensure safe monitoring.

Outcome:

Demonstrates the system's real-world application, timely alerts, and data integrity under operational conditions.

2. Project Documentation

Overview:

Comprehensive documentation detailing SHM system design, architecture, algorithms, and usage.

Sections:

- System Architecture: Diagrams of sensor networks, AI processing, and alert mechanisms.
- Codebase: Source code for AI models, sensor integration, and data visualization.
- User Guide: Instructions for engineers or administrators to monitor and interpret data.
- Admin Guide: Maintenance, calibration, and testing protocols for long-term deployment.
- Testing Reports: Accuracy, latency, fault tolerance, and reliability metrics.

Outcome:

Complete reference material for further development or deployment.

3. Feedback and Final Adjustments

Steps:

- Collect feedback from domain experts and stakeholders.

- Refine algorithms based on detected false positives/negatives.

- Re-test under controlled and real-world conditions.

Outcome:

Optimized SHM system ready for pilot implementation.

4. Final Project Report Submission

Sections:

- Executive Summary: Project goals, architecture, and achievements.

- Phase Breakdown: From initial design to final testing.

- Challenges: Environmental noise, sensor calibration, data handling.

- Solutions: Advanced filtering, sensor fusion, AI tuning.

- Outcomes: Current performance and limitations.

5. Project Handover and Future Works

Handover Details:

- System and code repository

- Future recommendations (e.g., integration with GIS, UAV inspections)

- Potential for large-scale deployment and cross-domain applications.

Attachments:

- Source code screenshots

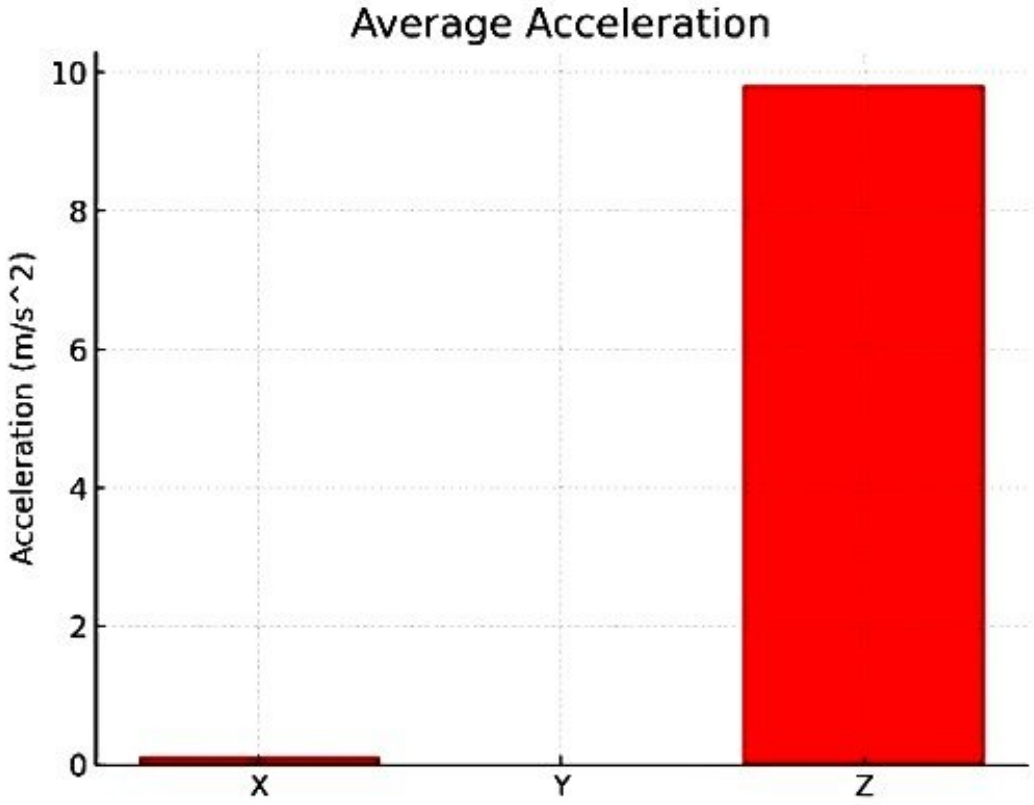
- Real-time monitoring dashboard snapshots

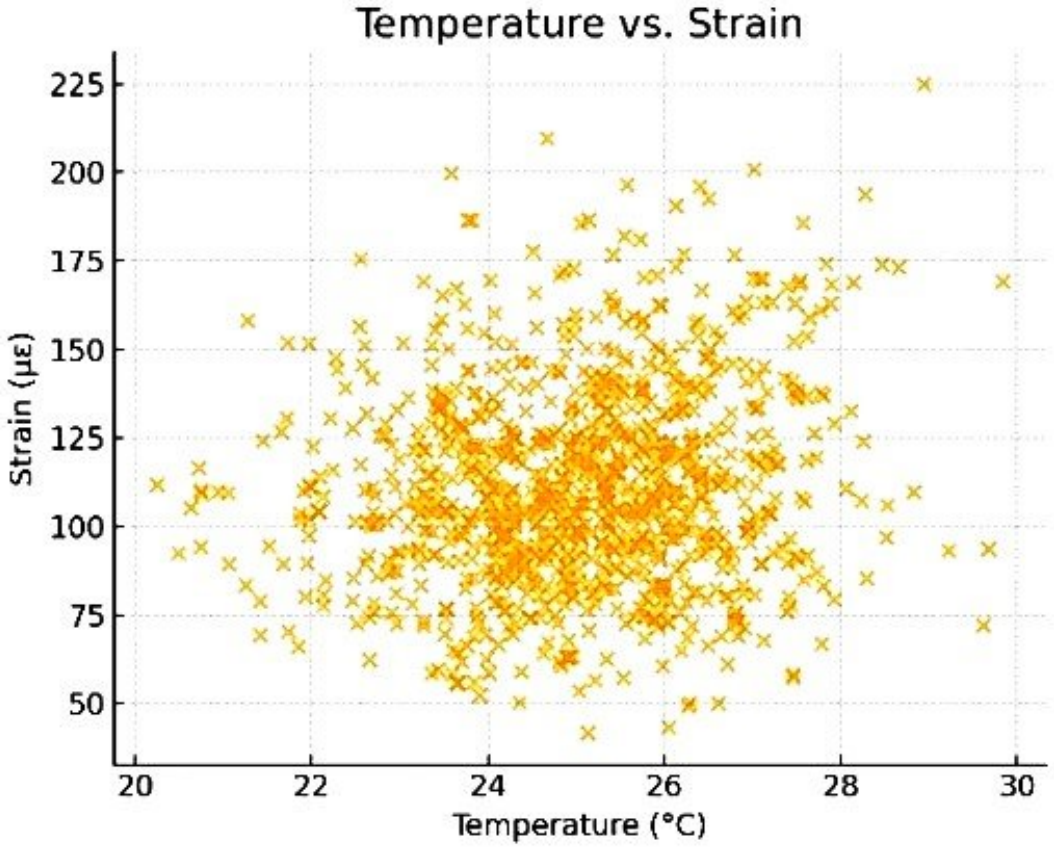
- Diagrams of structural models and sensor layouts

Source Code for Bar Chart and Scatter Plot

```
import pandas as pd
import matplotlib.pyplot as plt

# Load CSV
df = pd.read_csv('building_health_monitoring_dataset.csv')

# Plot
fig, axes = plt.subplots(1, 2, figsize=(12, 5))

# Bar chart: average acceleration
avg_accels = [
    df['Accel_X (m/s^2)'].mean(),
    df['Accel_Y (m/s^2)'].mean(),
    df['Accel_Z (m/s^2)'].mean()
]
axes[0].bar(['X', 'Y', 'Z'], avg_accels, color='red')
axes[0].set_title('Average Acceleration')
axes[0].set_ylabel('Acceleration (m/s^2)')

# Scatter plot: Temperature vs. Strain
axes[1].scatter(df['Temp (°C)'], df['Strain (με)'], alpha=0.6)
axes[1].set_title('Temperature vs. Strain')
axes[1].set_xlabel('Temperature (°C)')
axes[1].set_ylabel('Strain (με)')

plt.tight_layout()
plt.show()
```

Output of the Code (Bar and Scatter Plots)





Software Requirements

- Python 3.8 or later

- Pandas library (for data handling): pip install pandas

- Matplotlib library (for plotting): pip install matplotlib

- python-docx library (for Word document generation): pip install python-docx

- A CSV file named 'building_health_monitoring_dataset.csv' with appropriate sensor data