TITLE PAGE

Project Title: TrackFleet - IoT Enabled Vehicle Tracking and Predictive Maintenance System

Author: [Student Name]

Department: [University / Course Name]

Institution: [University Name]

Date: [Month, Year]

1. INTRODUCTION

Transportation fleets require continuous monitoring to ensure operational efficiency and safety. Traditional maintenance practices are reactive and based on fixed schedules, often resulting in unexpected failures. TrackFleet is an IoT based system that enables real time vehicle tracking and predictive maintenance using onboard sensors and analytical processing. The project demonstrates an academic implementation of sensing, communication, data processing, and visualization.

2. OBJECTIVES

The primary objectives of TrackFleet are: 1. To design an IoT based system for real time vehicle location tracking. 2. To collect and analyze engine and motion parameters. 3. To implement predictive maintenance alerts using threshold based rules. 4. To develop dashboards for monitoring fleet activity and health status.

3. SYSTEM OVERVIEW

TrackFleet consists of onboard sensing modules, a wireless communication layer, a backend processing service, and a visualization dashboard. The system supports real time data acquisition and historical trend analysis.

4. HARDWARE COMPONENTS

4.1 Microcontroller Unit

ESP32 development board with WiFi and Bluetooth support.

4.2 Sensors

GPS module for geolocation. Accelerometer (MPU6050) for motion and vibration detection. Temperature sensor for engine heat monitoring.

4.3 Power Supply

Powered through vehicle battery with voltage regulation.

5. CIRCUIT AND HARDWARE SETUP

Steps: 1. Interface GPS TX/RX pins to ESP32 UART. 2. Connect MPU6050 using I2C pins. 3. Mount components securely within the vehicle. 4. Perform continuity and signal tests before deployment.

6. SOFTWARE DEVELOPMENT

6.1 Firmware Workflow

Tasks performed by ESP32 include: WiFi setup and reconnection attempts. GPS coordinate extraction. Motion and vibration sampling. MQTT publishing at fixed intervals.

6.2 Data Format

Example payload: { gps_lat: 28.6139, gps_long: 77.2090, vibration: 0.34, temp: 68.2 }

6.3 Topic Structure

MQTT topics follow hierarchy: trackfleet/vehicle1/location trackfleet/vehicle1/engine

7. BACKEND AND STORAGE

7.1 MQTT Broker

Mosquitto broker running on local server or cloud instance.

7.2 Database

InfluxDB used for time series data. Retention policies applied for historical storage.

8. DASHBOARD AND VISUALIZATION

Grafana dashboards display: Live vehicle map location. Engine temperature timeline. Vibration levels to identify abnormal patterns.

9. IMPLEMENTATION STEPS

9.1 Hardware Deployment

Install modules in vehicle. Label units for fleet identification.

9.2 Firmware Deployment

Flash ESP32. Configure network parameters. Test MQTT data transmission.

9.3 Backend Setup

Start MQTT broker. Configure database buckets. Verify end to end data flow.

10. TESTING

10.1 Functional Testing

Check GPS accuracy and signal stability. Verify data refresh interval and dashboard updates.

10.2 Performance Testing

Measure latency between transmission and display. Ensure stable operation during continuous vehicle motion.

11. LIMITATIONS

System depends on cellular or WiFi coverage. GPS accuracy reduces in tunnels or blocked areas.

12. FUTURE ENHANCEMENTS

Integration with machine learning models. Battery powered long■range LoRaWAN support. Automated maintenance scheduling systems.

13. CONCLUSION

TrackFleet demonstrates an academic prototype of an IoT based fleet monitoring system and provides a foundation for future predictive maintenance research.

14. REFERENCES

[1] GPS Module Datasheet. [2] MQTT Protocol Specification. [3] InfluxDB and Grafana Documentation.