

Quality Control in Manufacturing - README

Quality Control in Manufacturing

Project Overview

This project implements a scalable, AI-powered quality control system for manufacturing environments. It leverages **AI** for defect detection, **IoT** for real-time monitoring, and **secure data handling** to enhance inspection accuracy and speed while reducing human error.

Features

- **AI-based Inspection**: Detects scratches, dents, and size anomalies.
- **IoT Integration**: Real-time sensor data improves decision-making.
- **Live Dashboard**: Displays alerts, metrics, and insights.
- **Secure Access**: Encrypted channels and role-based control.
- **Feedback Loop**: Enhances system based on real-world usage.

Project Demonstration

- **Walkthrough**: Image capture → AI inspection → Alert generation.
- **Accuracy Testing**: Precision in detecting common manufacturing defects.
- **Stress-Tested**: Handles operational load and continuous shift cycles.
- **Security Demo**: Demonstrates privacy compliance features.

Documentation Highlights

Quality Control in Manufacturing - README

- **Architecture**: AI + IoT + Database layers.
- **User Guide**: Visual steps for operators and admins.
- **Admin Guide**: Sensor thresholds, logging, deployment.
- **Testing Reports**: Case logs and outcomes for QA reference.

Feedback & Adjustments

- Conducted structured interviews for usability improvement.
- Enhanced dashboard visuals and confidence indicators.
- Regression and stress-tested all updates.

Final Report Summary

- **Executive Summary**: From idea to deployment.
- **Phased Development**: Tools, challenges, and fixes.
- **Outcomes**: Improved quality, reduced waste, increased efficiency.

Future Scope

- Mobile notifications for defects.
- Predictive analytics using time-series data.
- Multilingual dashboard for global operations.

Source Code

Quality Control in Manufacturing - README

The system includes a Python simulation to validate product size within set thresholds:

```
```python

import random

MIN_SIZE = 9.5

MAX_SIZE = 10.5

def simulate_quality_control():

 passed = 0

 failed = 0

 print("Quality Control Report:\n" + "-" * 30)

 for i in range(1, 21):

 size = round(random.uniform(9.0, 11.0), 2)

 status = "PASS" if MIN_SIZE <= size <= MAX_SIZE else "FAIL"

 print(f"Item {i:02}: Size = {size} mm -> {status}")

 passed += status == "PASS"

 failed += status == "FAIL"

 print("-" * 30)

 print(f"Total Passed: {passed}")

 print(f"Total Failed: {failed}")

simulate_quality_control()

```
```