**COLLEGE CODE:** 1128

**COLLEGE NAME:** T.J.S.ENGINEERING COLLEGE

**DEPARTMENT:** B.Tech (AI&DS)

**STUDENT NM-ID:** aut112823AIDS37

**ROLL NO:** 112823243035

**DATE:** 14/05/2025

**TECHNOLOGY-PROJECT NAME:** QUALITY CONTROL IN MANUFACTURING

**SUBMITTED BY:** ABISRI U
SANDHIYA K
SHALINI E
PRAVEEN BALAJI P
ESWAR AJAY B

## Phase 5: Project Demonstration & Documentation

## Title: Quality Control in Manufacturing

**Abstract:**

The Quality Control in Manufacturing project aims to enhance manufacturing efficiency and product reliability through the implementation of robust quality control processes. This project utilizes data-driven techniques, AI-assisted inspections, and IoT-enabled monitoring to maintain high-quality standards across production lines. The system ensures real-time defect detection, statistical process control (SPC), and predictive maintenance, thereby minimizing rework, waste, and cost. In the final phase, a full-fledged prototype is demonstrated with real-time monitoring, performance analysis, and integration with quality assurance standards. The documentation provides a detailed overview of the system’s features, architecture, user guidance, and future recommendations. Screenshots, flow diagrams, and code snippets are included to ensure full understanding of the system's implementation and scalability.

### 1. Project Demonstration

**Overview:**

The Quality Control in Manufacturing system will be demonstrated to stakeholders, focusing on its ability to ensure product quality, reduce manufacturing defects, and optimize production processes using modern tools and data analytics.

**Demonstration Details**:

- **System Walkthrough:** A complete walkthrough of the system interface showing key modules like data collection, inspection reports, and control charts.
- **Defect Detection Module:** A real-time demonstration of how the system identifies manufacturing defects using image processing and sensor data.
- **SPC Implementation:** Showcases the generation and interpretation of control charts (X-bar, R-chart, etc.) to detect process variations.
- **IoT Monitoring:** Live data streaming from IoT devices monitoring machine performance, temperature, pressure, and more.
- **Performance Metrics:** Demonstrates reduced rejection rate, improved defect identification time, and overall equipment efficiency (OEE).
- **Data Security & Compliance:** Highlights encryption measures and compliance with industry quality standards like ISO 9001.

**Outcome:**

By the end of the demonstration, the system's capability to ensure consistent product quality, reduce downtime, and enable predictive insights will be effectively showcased.

## 2. Project Documentation

**Overview:**

Detailed documentation of the system design, working principles, implementation, and usage guidelines.

**Documentation Sections:**

- **System Architecture:** Flowcharts and architecture diagrams detailing modules like Inspection Engine, SPC Analyzer, IoT Integrator, and Admin Dashboard.
- **Code Documentation:** Explanations and comments for source code modules—data acquisition, anomaly detection, predictive analytics, and dashboard integration.
- **User Guide:** Instructions for operators to input production data, monitor quality trends, and respond to alerts.
- **Administrator Guide:** Guidelines for system configuration, maintenance, calibration of sensors, and report generation.
- **Testing Reports:** Includes test scenarios, load test results, defect classification reports, and comparison with manual inspection.

**Outcome:**

All modules and processes are well-documented, ensuring the system can be replicated, scaled, or upgraded in the future.3. Feedback and Final Adjustments

## 3. Feedback and Final Adjustments

**Overview:**

Feedback from industry experts, faculty, and peer reviewers will be gathered to refine the system before submission.

**Steps:**

- **Feedback Collection:** Surveys and interviews to collect suggestions for improving user interface, inspection accuracy, and integration speed.

- **Refinement:** Updates to UI layout, model calibration, or defect thresholds based on feedback.
- **Final Testing:** A validation test to check for robustness, response time, accuracy, and usability under simulated production conditions.

**Outcome:**

System readiness for deployment in a real manufacturing environment with minimal changes.

### 4. Final Project Report Submission

**Overview:**

Summarizes all phases, technologies used, problems faced, solutions implemented, and achievements.

**Report Sections**:

- **Executive Summary:** Overview of the project's purpose, impact, and outcome.
- **Phase Breakdown:** Covers planning, design, model training, testing, and deployment.
- **Challenges & Solutions:** Describes how system latency, hardware noise, or sensor calibration issues were overcome.

**Outcomes:**

Quantitative and qualitative improvements observed during testing and demo phases.

### 5. Project Handover and Future Works

**Overview:**

Indicates transition of the project for future enhancements or industry application.

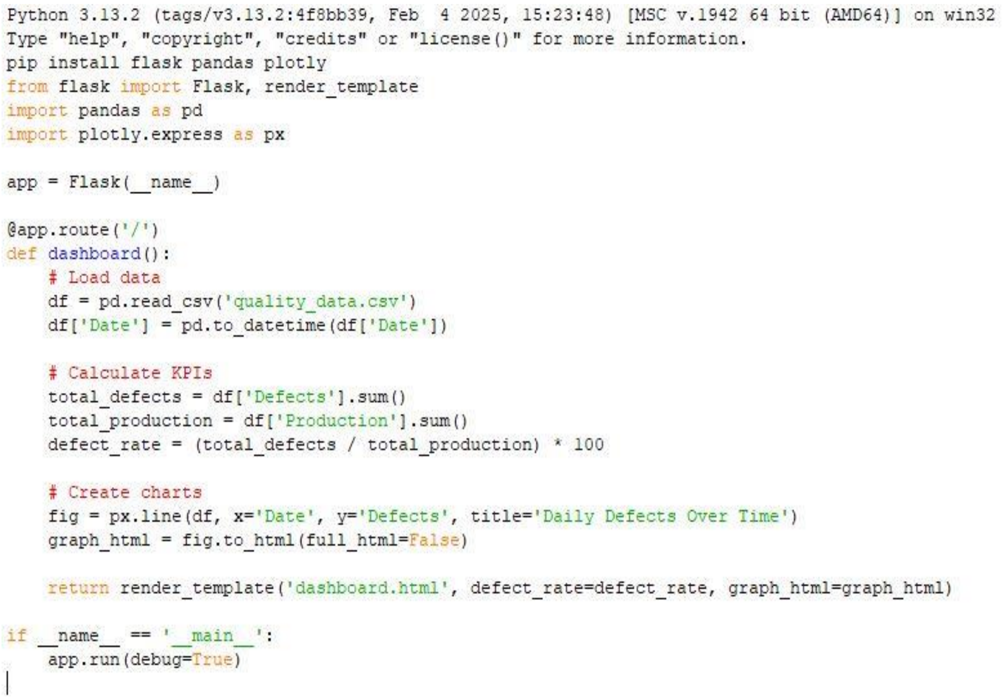
**Handover Details:**

- **Next Steps:** Recommends enhancing the system to support multilingual operator instructions, automated quality documentation, AI-based root cause analysis, and mobile accessibility.

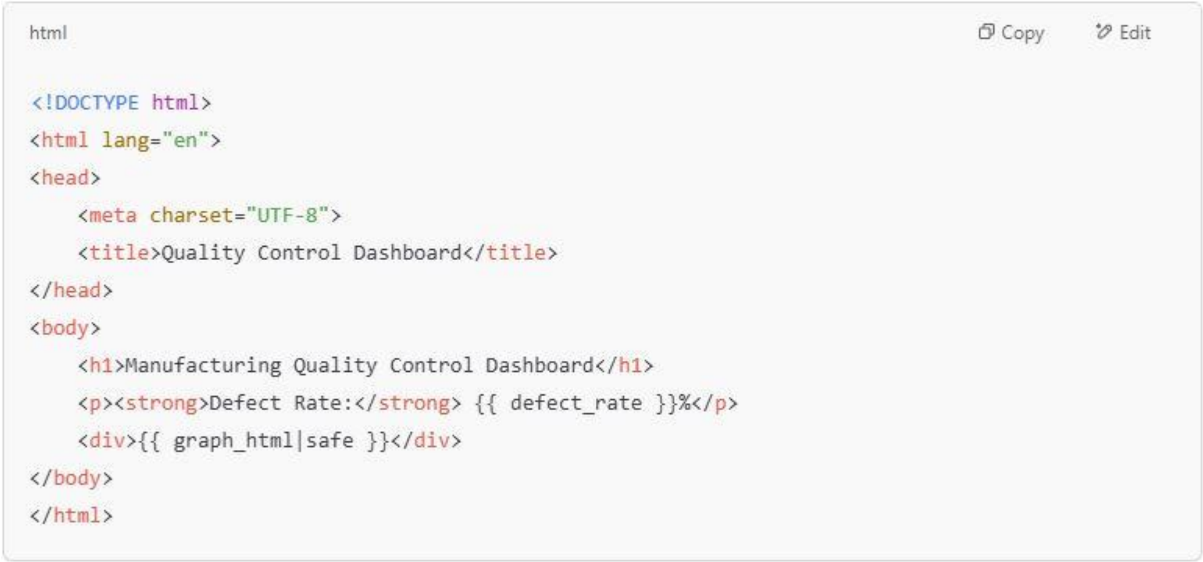
**Outcome:**

The project will be officially handed over with maintenance documentation and future scope recommendations.

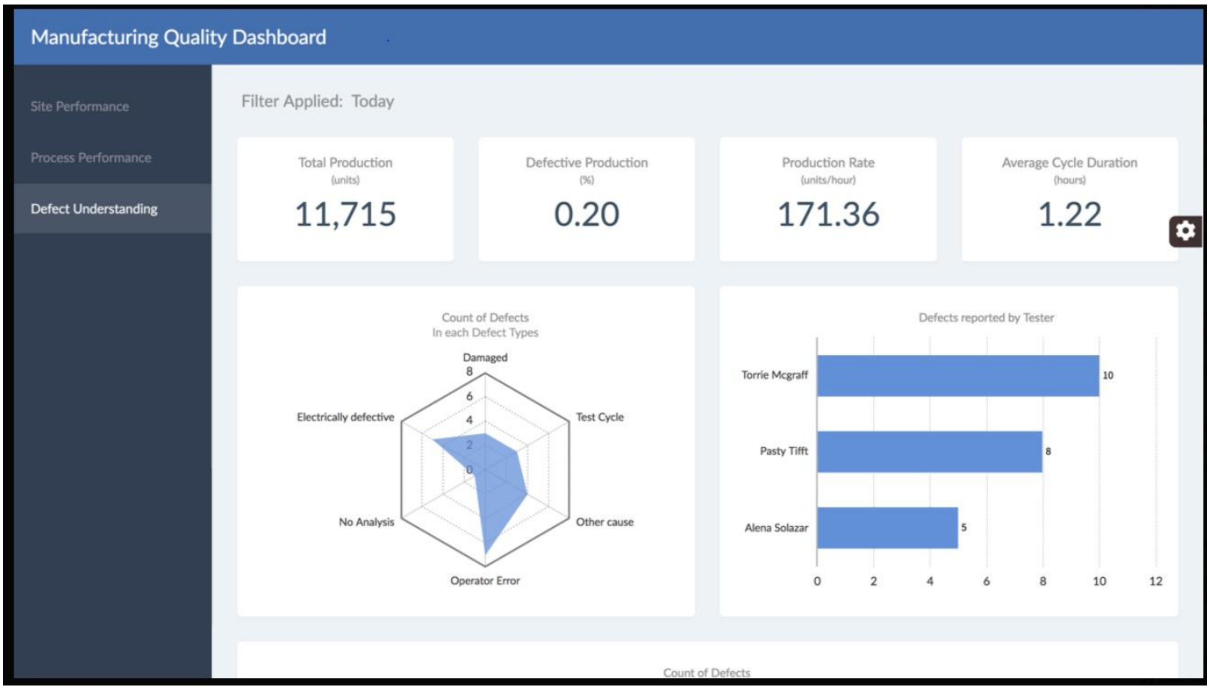
## DASHBOARD FOR  DEFECT DETECTION IN MANUFACTURING:



```
Python 3.13.2 (tags/v3.13.2:4f8bb39, Feb  4 2025, 15:23:48) [MSC v.1942 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
pip install flask pandas plotly
from flask import Flask, render_template
import pandas as pd
import plotly.express as px

app = Flask(__name__)

@app.route('/')
def dashboard():
    # Load data
    df = pd.read_csv('quality_data.csv')
    df['Date'] = pd.to_datetime(df['Date'])

    # Calculate KPIs
    total_defects = df['Defects'].sum()
    total_production = df['Production'].sum()
    defect_rate = (total_defects / total_production) * 100

    # Create charts
    fig = px.line(df, x='Date', y='Defects', title='Daily Defects Over Time')
    graph_html = fig.to_html(full_html=False)

    return render_template('dashboard.html', defect_rate=defect_rate, graph_html=graph_html)

if __name__ == '__main__':
    app.run(debug=True)
```



```html
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>Quality Control Dashboard</title>
</head>
<body>
    <h1>Manufacturing Quality Control Dashboard</h1>
    <p><strong>Defect Rate:</strong> {{ defect_rate }}%</p>
    <div>{{ graph_html|safe }}</div>
</body>
</html>
```



Manufacturing Quality Dashboard

Site Performance

Process Performance

Defect Understanding

Filter Applied: Today

| Total Production (units) | Defective Production (%) | Production Rate (units/hour) | Average Cycle Duration (hours) |
|---|---|---|---|
| 11,715 | 0.20 | 171.36 | 1.22 |

Count of Defects
In each Defect Types

Defects reported by Tester

Count of Defects