Performance Testing

Date	03 November 2025
Team ID	NM2025TMID01374
Project Name	To Supply Leftover Food to Poor
Maximum Marks	4 Marks

Objective

To ensure the *Food to Power* system performs efficiently under expected and peak loads by testing:

- Data collection from IoT devices (smart bins, sensors)
- Real-time data transmission to the cloud
- System response on dashboards and apps
- Energy data processing and reporting accuracy

Scope

Performance testing applies to the following components:

Component	**Description**
Smart Bin Sensors	Detect and send food waste data to the cloud
IoT Gateway	Handles multiple sensor inputs simultaneously
Cloud Server	Processes incoming data and updates dashboards
Web & Mobile Apps	Display live power, waste, and system status
Database	Stores sensor readings, power metrics, and analytics

Performance Testing Types

Type	**Purpose**	**Focus Area**
Load Testing	Evaluate system behavior under normal and peak load	Sensor data input rate, dashboard refresh rate
Stress Testing	Determine system limits beyond normal load	Cloud message queue capacity, API request limits
Endurance Testing	Check performance over a long duration	Continuous waste collection and energy generation
Spike Testing	Observe response to sudden surge in data	Simulate rapid bin fills or multiple uploads

Key Performance Metrics

Metric	**Target Value / KPI**	**Measurement Tool**
Response Time	≤ 5 seconds for dashboard updates	JMeter, Postman
Throughput	≥ 100 requests/sec (API load)	Apache JMeter
Latency	≤ 1 second from sensor to cloud	MQTT analyzer
Data Accuracy	≥ 97% of sensor data captured correctly	Custom script / database logs
Resource Utilization	CPU ≤ 75%, Memory ≤ 80% under load	Grafana, Prometheus
System Uptime	≥ 95% availability	AWS CloudWatch / Pingdom
Data Packet Loss	≤ 2% over MQTT network	MQTTLens, Wireshark
Database Query Time	≤ 2 seconds per query	SQL Profiler, MySQL Tuner

Test Environment Setup

Layer	**Environment Setup**
IoT Layer	10–50 smart bins simulated with ESP32 sending MQTT data
Network Layer	4G/LoRaWAN/WiFi communication setup
Cloud Backend	AWS / Google Cloud IoT Core for data ingestion
Application Layer	Node.js backend, React/Flutter frontend
Database Layer	MySQL + InfluxDB for hybrid data storage
Monitoring Tools	Grafana, Prometheus, JMeter, MQTTLens

Sample Test Scenarios

Test ID	**Scenario**	**Expected Result**
PT1	100 sensors send data simultaneously	System maintains ≤ 2s latency
PT2	Dashboard refreshes every 5 seconds under load	No UI lag, consistent updates
PT3	API receives 1000 requests/minute	Server handles load without errors
PT4	Sensor data spikes 5× for 1 minute	No data loss, stable system
PT5	Continuous operation for 48 hours	No crashes or data corruption
PT6	Add 500 new devices dynamically	Cloud autoscaling occurs successfully
PT7	Database grows to 10 million records	Query time remains < 3s

Recommended Tools

Tool	**Purpose**
Apache JMeter	API and load testing for backend performance
Postman	Manual performance & stress validation
Grafana + Prometheus	Real-time system and resource monitoring
MQTTLens / EMQX / Mosquitto	IoT performance testing and data flow tracking
AWS CloudWatch / Azure Monitor	Cloud performance and uptime tracking
Locust.io	Scalable user load simulation
Wireshark	Network traffic analysis and packet loss detection

Expected Outcomes

- Stable system operation under load
- Minimal latency in IoT-to-cloud data flow
- Efficient dashboard updates and analytics response
- Verified system scalability for future expansion