

## Performance and Testing

Date	03 November 2025
Team ID	NM2025TMID01909
Project Name	Medical Inventory Management
Maximum Marks	4 Marks

### Typical performance KPIs:

- **Response time:** e.g., < 2 seconds for inventory search or order submission.
- **Throughput:** Number of transactions (e.g., orders, updates) per second.
- **Concurrent users:** Number of simultaneous users (e.g., pharmacists, nurses, admin).
- **Resource utilization:** CPU, memory, I/O usage under load.
- **Scalability:** Ability to maintain performance as data or users increase.
- **Data volume:** Size of inventory database (could be millions of records).

### Identify Key Scenarios

Select **critical user flows** to test. These reflect real hospital or clinic operations.

#### Examples:

- Adding or updating stock (e.g., new drug batches).
- Searching inventory by product name, barcode, or expiry date.
- Processing supply requests from wards.
- Generating reports (stock levels, expiration alerts, usage history).
- Integrating with external systems (e.g., EHR, supplier APIs).
- Batch imports (e.g., nightly stock reconciliation).

### Test Data and Environment

- Use **realistic datasets** (inventory sizes, item categories, expiry dates).
- Test on a **production-like environment** (same database type, network config, etc.).
- Prepare **baseline metrics** for comparison (e.g., current system performance).

## Types of Performance Tests

Test Type	Goal	Example
<b>Load Test</b>	Measure performance under expected user load	500 concurrent users checking stock
<b>Stress Test</b>	Test system beyond expected load to find breaking point	Simulate 2,000 concurrent users
<b>Spike Test</b>	Sudden increases in load	Sudden batch of 1,000 supply requests
<b>Endurance (Soak) Test</b>	Long-duration stability test	Run system for 24 hours with typical activity
<b>Scalability Test</b>	Evaluate how performance changes with scaling	Double user count and observe response times

## Tools for Testing

Choose tools that suit your environment (web app, API, or cloud system):

- **Apache JMeter** – Open-source, great for API and web load testing.
- **LoadRunner** – Enterprise-grade solution for complex workflows.
- **k6** – Scriptable and developer-friendly for API load tests.
- **Gatling** – Focused on HTTP load testing and CI integration.
- **Locust** – Python-based load testing framework.
- **New Relic / Datadog / Grafana** – For monitoring application performance in real-time.

## Test Data and Environment

- Use **realistic datasets** (inventory sizes, item categories, expiry dates).
- Test on a **production-like environment** (same database type, network config, etc.).
- Prepare **baseline metrics** for comparison (e.g., current system performance).

## Metrics to Capture

- Response times (avg, 90th percentile, max)
- Throughput (requests/sec)
- Error rates
- CPU/memory usage
- Database query times
- Network latency
- Disk I/O

## Analyze and Optimize

- Identify **bottlenecks** (e.g., slow queries, unoptimized APIs, memory leaks).
- Tune system: caching, indexing, connection pooling, hardware scaling.
- Re-test after each optimization to validate improvements.

## Compliance & Safety

Because this involves **medical data**, ensure:

- **Data anonymization** during tests (no real patient info).
- **Compliance** with HIPAA / GDPR / FDA 21 CFR Part 11.
- Secure environments and encrypted connections (TLS).

## Example Test Case (Search Inventory)

Parameter	Value
Scenario	Pharmacist searches inventory for “Insulin”
Users	200 concurrent
Expected Response Time	≤ 2 seconds
Data Volume	1 million inventory records
Test Duration	30 minutes
Pass Criteria	95% of requests within SLA, <1% error rate