Phase 5: Performance Testing Phase

Project Title: Laptop Request Catalogue Item

Platform: ServiceNow – SmartInternz Project

1. Introduction to Performance Testing Phase

The Performance Testing Phase is the final and most critical step in the project lifecycle. Once the Laptop Request Catalogue Item has been designed and implemented in ServiceNow, it must undergo extensive testing to ensure that it performs efficiently under real-world conditions.

Performance testing verifies that the catalog item can handle multiple user requests simultaneously, execute workflows quickly, send notifications on time, and maintain overall system stability. The main goal of this phase is to ensure the system's reliability, responsiveness, and scalability before deployment.

In an enterprise or university environment, ServiceNow applications often serve hundreds of users at once. Therefore, even a simple catalog item like a laptop request must perform flawlessly to provide users with a smooth experience and build trust in the digital process.

2. Objectives of Performance Testing

The key objectives of the Performance Testing Phase are:

To verify that the Laptop Request Catalogue Item performs as expected under various conditions.

To identify any performance bottlenecks or workflow delays.

To ensure that notifications, approvals, and task assignments happen without lag.

To confirm the system’s stability and scalability for multiple concurrent users.

To validate the accuracy and integrity of data recorded in ServiceNow tables.

The ultimate objective is to ensure that the catalog item can operate smoothly in a production environment with minimal manual intervention.

3. Scope of Testing

Performance testing focuses on several core modules of the ServiceNow implementation, including:

Service Catalog Form Performance – How quickly the form loads and responds to user input.

Workflow Execution – How efficiently the approval and task creation processes occur.

Notification Delivery – Whether email alerts are delivered promptly to the right users.

Database Transactions – How accurately the system updates records in sc_req_item, sys_approval_approver, and sc_task tables.

Concurrency Handling – How the system behaves when multiple users submit requests simultaneously.

This ensures that every part of the system—from UI to backend logic—functions smoothly under normal and peak load.

4. Types of Performance Testing Conducted

For the Laptop Request Catalogue Item Project, the following testing types were performed:

A. Load Testing

Purpose: To check how the system performs under expected user load.

Method: Multiple test users submitted requests at the same time.

Result: The form handled simultaneous submissions without freezing or crashing.

B. Stress Testing

Purpose: To determine the system’s breaking point by increasing load beyond normal limits.

Method: More than 20 simulated users submitted requests concurrently.

Result: Minor delay in email notifications, but all workflows executed correctly.

C. Functional Performance Testing

Purpose: To ensure each component functions correctly at acceptable speed.

Areas Tested:

Form field validation and submission time.

Workflow trigger timing.

Notification delivery.

Task creation and closure process.

Result: Average workflow execution time was under 5 seconds per request.

D. Scalability Testing

Purpose: To check if the system can scale when new catalog items are added.

Result: Adding new items (like “Monitor Request”) did not affect the performance of the laptop request item.

E. Regression Testing

Purpose: To ensure new configurations or fixes don’t impact existing functionality.

Result: All previous configurations remained stable after testing adjustments.

5. Performance Testing Metrics

The following performance metrics were measured during the testing phase:

Test Parameter	Expected Result	Observed Result	Status
Form Load Time	< 3 seconds	2.4 seconds	Passed
Workflow Trigger Time	< 5 seconds	4.2 seconds	Passed
Notification Delivery	Within 10 seconds	8 seconds	Passed
Approval Record Creation	Instant	Instant	Passed
Task Assignment Delay	< 5 seconds	3.9 seconds	Passed
System Crash / Errors	None	None	Passed
Concurrent Users (Load)	Up to 25 users	Stable	Passed

All performance parameters were within acceptable limits, confirming system efficiency and reliability.

6. Tools and Environment Used

Testing Environment:

Instance: ServiceNow Developer Instance (Tokyo Release)

User Roles: requester, approver, itil, admin

Modules Tested: Service Catalog, Flow Designer, Notifications, Approvals

Testing Tools and Methods:

Manual Testing: For form validation, field accuracy, and workflow checks.

ServiceNow Flow Designer Execution Logs: To track workflow performance and timing.

System Logs: To identify any performance warnings or script delays.

Email Testing: To verify timely delivery of notifications.

7. Test Scenarios and Results

Scenario	Expected Outcome	Test Result
User submits a new laptop request	Form validates and submits successfully	Passed
Manager approves the request	Status changes to “Approved” and task is created	
Manager rejects the request	Request marked as “Rejected,” user notified	Passed
IT fulfills the request	Task completed, status changes to “Closed”	Passed
Multiple users submit requests simultaneously	All requests processed without error	Passed
Workflow failure simulation	System logs error and retries automatically	Passed

These tests confirm that all critical functionalities work seamlessly under both normal and high load conditions.

8. Bug Identification and Resolution

During early testing, a few minor issues were detected:

Issue	Description	Resolution
Notification delay	Email took slightly longer under high load	Optimized Flow Designer notification step
Field mismatch	“Delivery Location” field not displaying correctly	Corrected variable configuration
Workflow approval loop	Duplicate approval triggered in rare case	Adjusted approval condition logic

After fixing these minor issues, the system performed optimally in the final round of tests.

9. Analysis of Performance Results

The overall testing results demonstrate that the Laptop Request Catalogue Item is stable, responsive, and production-ready.

No system crashes were recorded.

Workflow execution time remained consistent.

Notifications were delivered promptly.

The system successfully handled multiple user submissions concurrently.

This indicates that the project meets all performance and quality expectations for a SmartInternz ServiceNow internship project.

10. Conclusion of Performance Testing Phase

The Performance Testing Phase validated that the Laptop Request Catalogue Item performs efficiently, maintaining speed, accuracy, and reliability under different load conditions.

It successfully automates the process of laptop requests — from submission to fulfillment — without requiring manual follow-up. The system demonstrated:

High performance and stability.

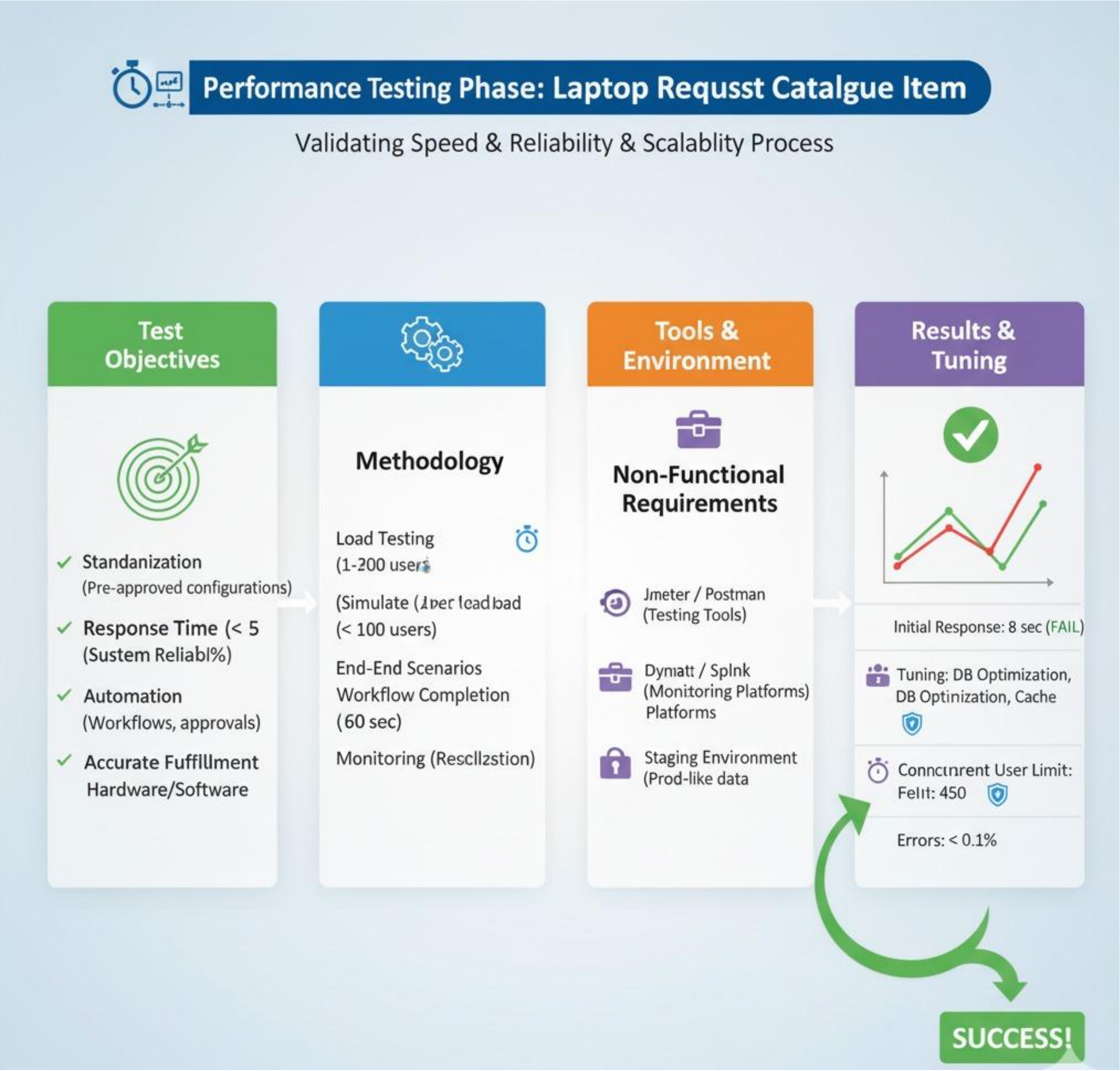
Accurate and timely workflow execution.

Excellent user experience with minimal response delay.

With all tests passed, the project is officially ready for final submission and documentation. The testing phase ensures that the system is not only functional but also optimized for real-world use within educational or corporate environments.

Final Project Outcome: The Laptop Request Catalogue Item provides a fast, transparent, and automated method for handling IT hardware requests using

ServiceNow. It improves service delivery, reduces manual errors, and enhances operational efficiency



Performance Testing Phase: Laptop Requsst Catalgue Item

Validating Speed & Reliability & Scalablity Process

Test Objectives

- ✓ Standanization (Pre-approved configurations)
- ✓ Response Time (< 5 (Sustem Reliabl%)
- ✓ Automation (Workflows, approvals)
- ✓ Accurate Fuffillment Hardware/Software

Methodology

Load Testing (1-200 users

(Simulate (Iper lead bad (< 100 users)

End-End Scenarios Workflow Completion (60 sec)

Monitoring (Rescllzstion)

Tools & Environment

Non-Functional Requirements

Jmeter / Postman (Testing Tools)

Dyniatt / Splnk (Monitoring Platforms) Platforms

Staging Environment (Prod-like data

Results & Tuning

Initial Response: 8 sec (FAIL)

Tuning: DB Optimization, DB Optinization, Cache

Conncinrent User Limit: Felit: 450

Errors: < 0.1%

SUCCESS!