PERFORMANCE TESTING PHASE

Performance Testing

Date	**28 June 2025**
Team ID	**LTVIP2025TMID60545**
Project Name	**Cafeteria Menu Display Using Service now**
Maximum Marks	**4 Marks**

✅ Assumptions

- You're using ServiceNow to develop or host a cafeteria menu (e.g., a custom app, portal widget, or catalog item).
- Performance testing is needed to ensure it loads quickly and scales well under user load.

🔍 Step-by-Step: Performance Testing Plan

1. Define Objectives

- What are you testing?
    - Menu load time
    - Data fetch from ServiceNow tables
    - Widget or catalog item response
- Performance metrics:
    - Response time
    - Load time
    - Throughput
    - Concurrent users

2. Identify Test Scenarios

Examples:

- User opens the cafeteria menu page/widget
- User selects a date or filters by meal type
- Admin updates the menu item list

3. Collect Baseline Data

Use **ServiceNow Performance Analytics** or **instance logs** to:

- Identify current response times

- Find peak usage hours
- Track database/API calls for the menu app

4. Choose a Testing Tool

Since ServiceNow is web-based, consider:

- **JMeter** (via HTTP Requests)
- **LoadRunner**
- **Locust**
- **ServiceNow ATF** (for functional, not load testing)
- **Browser Dev Tools** (for basic performance insight)

Note: Be careful with load testing ServiceNow production environments — always test on a **sub-production instance** and notify your ServiceNow admin/support.

5. Build Performance Test Scripts

If using **JMeter**, simulate:

- Logging in (if required)
- Navigating to the Cafeteria Menu page
- Triggering menu loading (GET or POST requests)

6. Run Tests

- Start with small user loads (5, 10, 50)
- Gradually increase to expected peak (e.g., 100–500 concurrent users)
- Monitor response times and error rates

7. Analyze Results

Key things to look at:

- Does the menu load in <3 seconds?
- Any 500/400 HTTP errors?
- Is the system stable at high concurrency?
- Database/API latency?

8. Optimize

Based on bottlenecks:

- Cache frequent menu data (use ServiceNow GlideCache or Edge Caching)
- Reduce server calls

- Use asynchronous loading or pagination

📊 Sample Metrics Table

Scenario	Expected Load	Avg Response Time	Max Response Time	Status
Load Menu Page	100 users	1.2s	2.5s	✅ Pass
Filter Menu Items	100 users	1.0s	2.1s	✅ Pass
Update Menu (Admin)	10 users	1.8s	3.0s	⚠️ Warn

🔐 Tips for ServiceNow-specific Testing

- Use GlideRecord wisely — avoid heavy queries in client scripts.
- Minimize synchronous server calls in widgets or forms.
- Use **Instance Scan** to detect performance anti-patterns.