Performance Testing Plan for Magento Website

Target Area for Performance Testing:

I will prefer to test performance for **APIs** first for all main components to make sure the apis are working efficiently and point out the key area to improve. Once apis load test will be done then i will perform the performance testing on website which will cover all aspects.

The main target areas will be:

- Checkout Flow
  - Cart
  - Shipping
  - Payment
  - Order Placement
- Search product
- Categories
- Filter/sorting

Why This Area?

The checkout process is the most critical user journey on any e-commerce platform because:

- It directly impacts conversion rates and revenue.
- It involves multiple integrated services (cart, authentication, shipping, payment, inventory).
- Performance issues here lead to abandoned carts and poor user satisfaction.

Thus, testing this area gives maximum return on performance validation efforts.

Other important area used on ecommerce website are search product results, category data and filtering data.

Testing Approach

We would design a combination of load, stress, and endurance tests targeting the checkout process, supported by real-user simulations.

1. **Load Testing**

Objective: Measure system behavior under expected traffic (normal peak load).

This is important to make sure the website is working fine under anticipated load.

2. **Stress Testing**

Objective: Identify the breaking point by increasing load beyond expected usage.

For any product its important to know the maximum limit of users. Its critical to know what is breaking point of the application which help to improve the

3. **Endurance Testing**

Objective: Check how the system performs over long durations.

While having sale days like new year or black fridays, this really helps to understand the application behavior when customers are using or selecting the products under a big load.

Key Scenarios to Cover

- **Guest Checkout:** Add product to cart → Checkout as guest → Place order
- **Logged-in User Checkout:** Login → Add multiple items → Use saved address → Place order
- **Simultaneous Add-to-Cart Actions:** Multiple users adding different products concurrently
- **Shipping Method Selection:** Simulate delays or selection issues from shipping APIs
- **Payment Submission:** Load test simulated payment gateway responses
- **Search product:** Load test to check the performance of the test results

Performance Test Parameters

Parameter	Value / Target
Concurrent Users	100, 500, 1000
Ramp-up Time	10
Test Duration	30 mins to 1 hours
Response Time Target	< 2 sec
Error Rate Threshold	< 1%
Throughput	Orders per second
System Resource Metrics	CPU, Memory, DB connections, I/O

Metrics to Monitor

- Page Load Times
- API Response Times (especially for cart, shipping, payment, place order, search products, categories)
- Error Rates
- Throughput
- System Resource Usage (CPU, RAM, Disk, DB)

Reporting

I will create a comprehensive report for each type of testing, clearly defining the metrics, parameters, and detailed results to ensure complete transparency. The purpose of these reports is to help all stakeholders — especially developers — easily understand the identified issues, performance bottlenecks, or weak areas. Each report will include clear insights into the root cause, supported by evidence, to enable swift and effective resolution. This approach ensures continuous quality improvement and faster feedback loops throughout the development lifecycle.

Conclusion

The checkout process is the ideal candidate for performance testing because it combines critical functionality, real-time integration, and high business value. A well-designed performance test for this flow helps ensure that the platform performs reliably under real-world conditions and user loads, especially during peak times like promotions or sales.