**Approach on Performance Testing**

The performance test will focus on load testing, stress testing, and endurance testing to evaluate:

* Response times under different user loads.
* System stability under peak traffic.
* Resource utilization (CPU, memory, database, etc.).
* Error rates during high traffic**.**

**Test Scenarios:**

**Scenario 1: Load Testing for Product Catalog Page**

* **Objective**: Measure the response time and system behavior under expected user load.
* **Parameters**:
  + Simulate **100–1,000 concurrent users** browsing the product catalog.
  + Test with different filter combinations (e.g., sorting by price, category, etc.).
  + Measure:
    - Page load time.
    - Database query performance.
    - Server resource usage (CPU, memory).
    - Error rates (e.g., timeouts, 5xx errors).

**Scenario 2: Stress Testing for Checkout Process**

* **Objective**: Determine the breaking point of the checkout process.
* **Parameters**:
  + Simulate **50–500 concurrent users** going through the checkout process.
  + Include steps like adding items to the cart, entering shipping details, and completing payment.
  + Measure:
    - Transaction success rate.
    - Payment gateway response time.
    - System recovery after peak load.
    - Error rates during payment processing.

**Scenario 3: Endurance Testing for the Entire Site**

* **Objective**: Evaluate system stability over an extended period.
* **Parameters**:
  + Simulate **200–500 concurrent users** performing mixed activities (browsing, adding to cart, checking out) for **1–2 hours**.
  + Measure:
    - Memory leaks.
    - Database connection stability.
    - Average response time over time.
    - System resource usage trends.

**Scenario 4: Spike Testing for Flash Sales**

* **Objective**: Test the site’s ability to handle sudden traffic spikes.
* **Parameters**:
  + Simulate a **10x increase in traffic** (e.g., from 100 to 1,000 users) within a short period (e.g., 1 minute).
  + Measure:
    - How quickly the system scales (if using auto-scaling).
    - Response time degradation.
    - Error rates during the spike.

**Scenario 5: Database Performance Testing**

* **Objective**: Evaluate the database’s ability to handle high query loads.
* **Parameters**:
  + Simulate heavy database operations (e.g., searching for products, updating inventory).
  + Measure:
    - Query execution time.
    - Database connection pool usage.
    - Lock contention or deadlocks.

**Scenario 6: Search Functionality Performance**

* **Objective**: Test the performance of the search feature, which is heavily used in e-commerce sites.
* **Parameters**:
  + Simulate **50–500 concurrent users** performing searches with different keywords (short, long, and misspelled).
  + Measure:
    - Search response time.
    - Database query performance.
    - Impact on server resources (CPU, memory).

**Scenario 7: User Login and Session Management**

* **Objective**: Evaluate the performance of user authentication and session handling.
* **Parameters**:
  + Simulate **100–1,000 concurrent users** logging in and performing actions (e.g., browsing, adding to cart).
  + Measure:
    - Login response time.
    - Session creation and management overhead.
    - Session timeout behavior under load.

**Scenario 8: Add to Cart and Cart Management**

* **Objective**: Test the performance of adding items to the cart and managing the cart.
* **Parameters**:
  + Simulate **200–1,000 concurrent users** adding items to their carts and updating quantities.
  + Measure:
    - Cart update response time.
    - Database performance for cart-related queries.
    - Error rates during cart updates.

**Scenario 9: Image and Media Loading Performance**

* **Objective**: Evaluate the performance of image and media loading on product pages.
* **Parameters**:
  + Simulate **100–500 concurrent users** browsing product pages with high-resolution images.
  + Measure:
    - Image load time.
    - Bandwidth usage.
    - Impact on server and CDN performance.

**Scenario 10: API Performance Testing**

* **Objective**: Test the performance of APIs used by the website (e.g., product details, pricing, inventory).
* **Parameters**:
  + Simulate **100–1,000 concurrent API requests**.
  + Measure:
    - API response time.
    - Error rates (e.g., 4xx, 5xx).
    - Throttling or rate-limiting behavior.