**Load test**

Load tests and scenarios are needed to investigate the behavior of the search

module of https://www.n11.com/ header and listing the results after the search

➔ Write test scenarios for the relevant situation

➔ Code scenarios using JMeter or Locust (1 user is enough)

During the load testing on N11 website by searching telephone, it gives 403 Forbidden. Based on my search, the Cloudflare can be caused this issue. As a solution can be use VPN as showing to access the page from Turkiye.

You can find some screenshots that show the result of the action.

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

**Note:** Accessing the N11 from abroad and not getting proper result. Another scenario which is the similar the given scenario is executed on Amozon.co.uk website.  
**Steps:**  
1) Went to Amazon.co.uk



2) Type ‘laptop’ in search box

3) Click enter

4)Select the “Dell” laptop

5)Click the “Buy Now” button

6)Page navigates to “Sign” page

A screen shot of a computer

AI-generated content may be incorrect.

A computer on a website

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

**The following are the results:**

**1st Scenario**

Number of Threads =1

Ramp-up period(sec)=1

Loop =1

**Results: 3 steps successfully passed**

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

**2nd Scenario**

Number of Threads =10

Ramp-up period(sec)=50

Loop =1

**Results: 3 steps for each Threads successfully passed**

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

**3rd Scenario:**

Number of Threads =50

Ramp-up period(sec)=100

Loop =1

**Results: Some of the steps are failed**

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

**Comments on the 3 results:**

The first two scenarios show that the system can easily handle 1 and 10 users, while the third scenario with 50 users starts to hit capacity limits and produces errors.

**Basic compatison of scenarios:**

**First (1 thread):**

* Total 3 requests, average response time , error rate 0%.
* Throughput about 0.66 request/s**,**, so the system is very stable under low load.

**Second (10 threads):**

* Total 30 requests, average response time 723ms, error rate 0%.
* ​Throughput about 0.63 request/s, and even with 10 users there are no errors, so the system handles this level comfortably

**Third (50 threads):**

* Total 150 requests, average response time 643ms (the average looks better, but the distribution is much wider).
* ​Error rate 5.33% (especially 12% errors on search requests) and max response time up to 1588ms, which shows stability problems at 50 users.

**Stability & Service Level Agreement(SLA):**

If the SLA is for example “p95 < 1.5 s, errors < 1%”, at 50 users the test approaches or exceeds such thresholds, especially on checkout, which might be unacceptable for production.

**Next steps for the further examination**

Run the same 3 scenarios for longer durations (at least 5–10 minutes) with loop>1 and track p90/p95 response times and error trends over time.

​

Include intermediate load levels like 20, 30, 40 threads to find the exact point where errors and response times start to spike and build a clear capacity curve.

​

Inspect detailed error information (Response Data, HTTP codes) for search and checkout samplers to identify whether the bottleneck is network, application, or an external dependency.