Burger House — k6 Performance Test Report

Environment:
NestJS (TypeORM + SQLite), Node.js 20.x
Executed via k6 on Windows PowerShell
Scenarios: 20–40 VUs × 30s

Smoke Test (Public Endpoints)

Endpoints: GET /categories, GET /items
Scenario: 20 VUs for 30s

Metric	Result
Average latency	≈ 10.1 ms
95th percentile (p95)	≈ 27.5 ms
Error rate	0 % (0/1200 requests)
Checks passed	100 % (1200/1200)

■ All public catalogue endpoints returned 200 OK under 20 concurrent users.
■ Stable average latency below 15 ms and no failed checks.

Load Test (Public + Authenticated Read)

Endpoints: GET /categories, GET /items, GET /auth/me
Scenario: 40 VUs (20 public + 20 authed) for 30s

Metric	Result
Average latency	≈ 18.5 ms
95th percentile (p95)	≈ 44.3 ms
Error rate	0 % (0/1803 requests)
Checks passed	100 % (1800/1800)

■ Full success rate at 40 VUs with average request latency under 20 ms.
■ Authenticated endpoints handled concurrency without token errors.

Load Test (Write Operations — WIP)

Endpoints: POST /cart, POST /orders (authenticated)
Scenario: 40 VUs (20 public + 20 authed) for 30s

Metric	Result
Average latency	≈ 31.8 ms
95th percentile (p95)	≈ 80.8 ms
Success (add-to-cart)	≈ 99 % OK
Success (create-order)	≈ 9 % OK (expected – still in development)

■ Partial success — Cart flow mostly stable; order API still under development.
No server errors or crashes observed; only expected 4xx validations.

Summary

Conducted end-to-end API performance benchmarking using k6 on the Burger House NestJS backend.
• Designed smoke, load, and write-scenario tests simulating 20–40 concurrent users.
• Achieved p95 latency ≈ 27–44 ms and 0 % errors on public/auth endpoints.
• Validated API reliability under load; transactional routes under active development.