🚀 k6 Performance Testing Project

Hands-On Load Testing with Real Execution & Analysis

Candidate Name: Routh Kiran Babu
Role Targeted: SDET / QA Automation / Performance Testing Engineer
Tool: k6 (Open-source Performance Testing Tool by Grafana Labs)

📑 Table of Contents

1️⃣ Introduction to k6

k6 is a **modern, open-source performance testing tool** developed by **Grafana Labs**.
It is designed for **load testing, stress testing, and performance validation** of APIs and web applications using **JavaScript**.

Recruiters prefer k6 because it:

- Uses **developer-friendly JavaScript**
- Integrates easily with **CI/CD pipelines**
- Produces **clear, meaningful performance metrics**

2️⃣ What is Performance Testing?

Performance testing evaluates:

- **Response time**
- **Throughput**
- **Stability**
- **Scalability**

It ensures that an application performs well **under expected and peak load conditions**.

3️⃣ Why k6 for Modern Testing?

Traditional Tools	k6
Heavy UI	CLI-based
Script complexity	Simple JS
Hard CI/CD	CI-ready
Costly	Open-source

4️⃣ Key Features of k6

- JavaScript-based scripting
- Built-in checks and thresholds
- Lightweight and fast execution
- Native CLI tool
- Supports HTTP, REST, GraphQL, WebSockets
- CI/CD friendly (Jenkins, GitHub Actions)
- Integrates with Grafana dashboards

5️⃣ Project Objective

To perform a **load test** on a real web application using **k6**, validate HTTP response status, and analyze performance metrics.

Target URL:
https://quickpizza.grafana.com

6️⃣ Prerequisites

- Windows / macOS / Linux
- Basic JavaScript knowledge
- Internet connection

7️⃣ k6 Installation Guide

🔹 Windows

```
winget install k6
```

🔹 macOS

```
brew install k6
```

🔹 Linux

```
sudo apt install k6
```

🔹 Verify Installation

```
k6 version
```

8️⃣ Project Test Script (With Explanation)

```
// Import HTTP module to make network requests
import http from 'k6/http';
```

```
// Import check for assertions and sleep to simulate user wait time
import { check, sleep } from 'k6';

// Define test configuration options
export const options = {
  vus: 10,        // Number of Virtual Users running in parallel
  duration: "15s" // Total test execution time
};

// Default function executed by each virtual user
export default function () {

  // Send an HTTP GET request to the application
  const res = http.get('https://quickpizza.grafana.com');

  // Validate response status using check
  check(res, {
    "Status is 200.": (r) => r.status === 200
  });

  // Pause execution for 1 second to simulate real user behavior
  sleep(1);
}
```

9️⃣ How to Execute the Test

1. Save the file as:

```
k6_load_test.js
```

2. Open terminal / command prompt
3. Navigate to the file location
4. Run the command:

```
k6 run k6_load_test.js
```

🔟 Understanding the Output

Sample Output Metrics:

Metric	Meaning
`vus`	Active virtual users
`http_req_duration`	Response time of requests
`http_req_failed`	Failed requests
`checks`	Validation success rate
`iterations`	Total executed requests

Example Interpretation:

- **100% checks passed** → Application is stable
- **Low response time** → Good performance
- **Zero failed requests** → No server errors

1️⃣1️⃣ Challenges Faced & Solutions

🔴 Challenge 1: Understanding Virtual Users

Solution:
Studied how VUs simulate concurrent users and how duration controls test runtime.

🔴 Challenge 2: Interpreting Metrics

Solution:
Mapped metrics like `http_req_duration` and `checks` to real-world performance indicators.

🔴 Challenge 3: Script Simplicity vs Realism

Solution:
Added `sleep(1)` to simulate realistic user think time.

1️⃣2️⃣ Real-World Use Cases

- API load testing before production release
- Performance regression testing
- CI/CD performance validation
- Traffic spike simulation
- Microservices performance monitoring

1️⃣3️⃣ Key Learnings

- Hands-on experience with **k6 scripting**
- Understanding of **load testing concepts**
- Interpreting real performance metrics
- Writing **clean, maintainable test scripts**
- Performance testing mindset for SDET role

1️⃣4️⃣ Conclusion

This project demonstrates:
✔ Practical performance testing skills
✔ Tool proficiency in **k6**
✔ Real execution & analysis
✔ Recruiter-ready documentation

> This project reflects my readiness to contribute as an **SDET / Performance Test Engineer** in real-world applications.

📌 Recruiter Tip (Optional to Add at End)

GitHub Repo:
(https://github.com/RouthKiranBabu/Mini_Major--Projects./tree/main/Projects/Year_Equals_2026/01_Jan/01_%5BK6%5D%20-%20HTTP%20request%20and%20Performance%20Testing)
LinkedIn: *(https://www.linkedin.com/in/routhkiranbabu/)*